Matching Recall and Storage in Sequence Learning with Spiking Neural Networks

Johanni Brea, Walter Senn, Jean-Pascal Pfister
Physiology Department
University of Bern
Switzerland

{brea, senn, pfister}@physio.unibe.ch

Senior editor	
Section	Behavioral/Systems/Cognitive Neuroscience
Abbreviated title	Sequence Learning
Corresponding author	Johanni Brea Physiology Department University of Bern 5 Bühlplatz, CH-3012 Bern

Number of figures	5
Number of tables	0
Number of pages	31
Keywords	sequence learning, supervised learning, recurrent neural network, synaptic plasticity, spiking neurons
Acknowledgments	This research is supported by the Swiss National Science Foundation (J.B. & W.S.: 31-133094, J.-P.P.:PZ00P3_137200) and a grant from the Swiss SystemsX.ch initiative (Neurochoice). We gratefully acknowledge discussions with Robert Urbanczik and Danilo Jimenez-Rezende.

Abstract

Storing and recalling spiking sequences is a general problem the brain needs to solve. It is however unclear what type of biologically plausible learning rule is suited to learn a wide class of spatio-temporal activity patterns in a robust way. Here we consider a recurrent network of stochastic spiking neurons composed of both visible and hidden neurons. We derive a generic learning rule that is matched to the neural dynamics by minimizing an upper-bound on the Kullback-Leibler divergence from the target distribution to the model distribution. The derived learning rule is consistent with Spike-Timing Dependent Plasticity in that a presynaptic spike preceding a postsynaptic spike elicits potentiation while otherwise depression emerges. Furthermore, the learning rule for synapses that target visible neurons can be matched to the recently proposed voltage-triplet rule. The learning rule for synapses that target hidden neurons is modulated by a global factor which shares properties with astrocytes and gives rise to testable predictions.

1 Introduction

Increasing experimental evidence in different brain areas shows that precise spike timing can be learned and reliably reproduced over trials. For example, in adult songbirds who learned to repeat the same song, HVC neurons - which are targeting the pre-motor area RA - reproduce precise spiking patterns during the song production (Hahnloser et al., 2002). In the rat, repeated presentation of a moving spot imprints a stereo-typical spiking activity in the visual cortex that can be retrieved after learning (Xu et al., 2012). However, it remains unclear how those spiking patterns can be efficiently learned through synaptic plasticity.

Learning to autonomously reproduce a given spatio-temporal activity pattern is a fundamental challenge approached by the earliest models of recurrent neural networks (Amari, 1972; Hopfield, 1982; Herz et al., 1988; Gerstner et al., 1993; Horn et al., 2000). However, the proposed simple temporal Hebbian rule could be problematic because of its lack of robustness during recall (Hopfield, 1982). Since then, heuristic models for supervised sequence learning in recurrent networks have also been developed for spiking neurons (Ponulak and Kasiński, 2010). All these studies suffer from the same fundamental problem: the synaptic learning rule (storage) is not matched to the neural dynamics (recall) in the sense that the plasticity rule is not derived from first principles that are formulated in terms of neural dynamics.

Another limitation of existing models for sequence learning with spiking neurons (Barber and Agakov, 2002) is the restricted class of spiking patterns that can be produced with only visible neurons, i.e. neurons that receive the supervising signal. One possible solution is to include a reservoir of hidden neurons, which do not receive a supervised teaching signal, but the synaptic weights between those neurons are usually fixed (Maass and Markram, 2002). Other approaches, such as the Boltzmann Machine (Ackley et al., 1985), Helmholtz Machine (Dayan et al., 1995), or their extensions to the temporal domain (Hinton et al., 1995; Sutskever et al., 2009) consider learning rules for the weights towards hidden neurons, but the biological plausibility of those rules is open to discussion.

Here we study a general framework with both visible and hidden neurons. In a local neural circuit, neurons that may receive strong input (teaching signal) from outside the circuit are considered to be visible, and neurons that receive only input from neurons inside the circuit are considered to be hidden. We propose a generic synaptic learning rule that is matched to the neural dynamics and that can be adapted to a wide range of neuron models. The learning rule minimizes an upper-bound on the Kullback-Leibler divergence from the target spiking distribution to the distribution produced by the network. This learning rule is consistent with Spike-Timing Dependent Plasticity (STDP) (Bi and Poo, 1998). The match of recall and storage appears as an explicit link between the time constants in the learning window and the neural time constants. Furthermore, the plasticity rule for synapses that target visible neurons is consistent with the voltage-triplet rule (Clopath et al., 2010). Finally, beside the pre- and postsynaptic components, the learning rule for synapses that target hidden neurons is modulated by a global factor. Interestingly, this global factor shares common properties with astrocytes.

2 Materials and Methods

Neuron model. We consider a recurrent network of N spiking neurons over a duration of T time bins. Spiking of neuron i is characterized by the spike train x_i, with $x_i(t) = 1$ if a spike is emitted at time step t, and $x_i(t) = 0$ otherwise. The membrane potential of neuron i is described as in the Spike-Response Model (Gerstner and Kistler, 2002):

$$u_i(t) = u_0 + \sum_{j=1}^{N} w_{ij} x_j^{\epsilon}(t) + x_i^{\kappa}(t)\,, \tag{1}$$

where w_{ij} is the synaptic strength from neuron j to neuron i, $x_k^{\alpha}(t) = \sum_{s=1}^{\infty} \alpha(s) x_k(t-s)$ represents the convolution of spike train x_k with kernel α and u_0 is the resting potential. The postsynaptic kernel is characterized by $\epsilon(s) = \delta_{s,1}$ for the one time step response kernel scenarios of Fig. 2 and Fig. 3, whereas in Fig. 4 it is given by $\epsilon(s) = (\exp(-s/\tau_1) - \exp(-s/\tau_2))/(\tau_1 - \tau_2)$ for $s \geq 0$ and the adaptation kernel is characterized by $\kappa(s) = c \exp(-s/\tau_r)$ for $s \geq 0$, with both kernels vanishing for $s < 0$. For the clarity of the exposition, we chose such a simple neural model description. Note however that almost any neural model could be considered (e.g. conductance-based models). The only constraint is that the dynamical model should be linear in the weights, i.e. any dynamical model of the form $\dot{u}_i = f_i(u_i) + \sum_j w_{ij} g_{ij}(u_i, x_j)$ is suitable.

Consistently with the stochastic Spike-Response Model or equivalently the Generalized Linear Model (Pillow and Latham, 2008), noise is modeled by stochastic spiking given the (noise-free) membrane potential u Eq. 1, i.e. the probability that neuron i emits a spike in time bin t is a function of its membrane potential: $P(x_i(t) = 1 | u_i(t)) = \rho(u_i(t))$. We stress the fact that given its own membrane potential, the spiking process is conditionally independent of the spiking of all the other neurons at this time. Due to this conditional independence, the probability that the network with (recurrent) weight matrix w is generating the spike trains $x = (x_1, \ldots, x_N)$ can be calculated explicitly as the product of the probabilities for each individual spike train, hence a product of factors $\rho(u_i(t))$ and $(1 - \rho(u_i(t))$, depending on whether neuron i did or did not fire at time t, respectively. Abbreviating $\rho_i(t) = \rho(u_i(t))$, this amounts for the log- likelihood (Pfister et al. (2006))

$$\log P_w(x) = \sum_{t=1}^{T} \sum_{i=1}^{N} x_i(t) \log \rho_i(t) + (1 - x_i(t)) \log(1 - \rho_i(t)) \,. \tag{2}$$

Unless mentioned otherwise, the firing probability will be assumed to be a sigmoidal $\rho(u) = 1/(1 + \exp(-\beta u))$, with parameter β controlling the level of stochasticity. We introduced this parameter for convenience: for given weights w, the stochasticity of the network can be varied by changing the parameter β, which multiplies the weights. In the limit $\beta \to \infty$ each neuron acts like a threshold unit and therefore makes the network deterministic.

Learning task. We separate our N neurons into disjoint classes of N_v visible and N_h hidden neurons. Correspondingly, the spike trains x are separated into those generated by the visible and hidden neurons, $x = (v, h)$. We assume that learning in the recurrent network consists of adapting all the synaptic weights w_{ij} between and among the two types of neurons such that the Kullback-Leibler (KL) divergence

$$\mathcal{D}(P^*(v) \| P_w(v)) = \left\langle \log \frac{P^*(v)}{P_w(v)} \right\rangle_{P^*(v)} \tag{3}$$

from the target distribution $P^*(v)$ to the model distribution $P_w(v)$ of the spike trains of visible neurons becomes as small as possible (see also Fig. 1). Gradient descent would amount to change the synaptic strength proportionally to the gradient of the negative KL-divergence in Eq. 3,

$$\Delta w_{ij} \propto \left\langle \frac{\partial}{\partial w_{ij}} \log P_w(v,h) \right\rangle_{P_w(h|v)P^*(v)} , \tag{4}$$

using the fact that $\partial/\partial w_{ij} \log P_w(v) = \sum_h P_w(h|v) \partial/\partial w_{ij} \log P_w(v,h)$. Unfortunately, sampling the sequences of hidden states given a sequence of visible states as suggested by Eq. 4, $h \sim P_w(h|v)$, is tricky, since a certain hidden state at time t should be consistent with all visible states, in particular those at later points in time. How to select such hidden states without violating causality is unclear. To circumvent this problem, we suggest to minimize instead an upper bound of the KL-divergence. In an alternative approach, using a Monte Carlo Markov Chain to sample from $P(h|v)$, Mishchenko and Paninski (2011) propose sampling by forward-backward algorithm from the conditional distribution of one hidden neuron spike train given all the other spike trains. It is however unclear how this can be implemented in biologically plausible neural networks.

Learning by minimizing an upper bound on the KL divergence. To define a biologically plausible sampling procedure we make use of the fact that the firing of each neuron is independent of the activity in the other neurons given the past (c.f. Eq. 2). This allows to factorize the probability for visible and hidden spike sequences into $P_w(v,h) = R_w(v|h)\, Q_w(h|v)$, where

$$\begin{aligned} R_w(v|h) &= \prod_{i=1}^{N_\mathrm{v}} \prod_{t=1}^{T} P(v_i(t)|x(t-1), x(t-2), \ldots) \quad \text{and} \\ Q_w(h|v) &= \prod_{i=N_\mathrm{v}+1}^{N} \prod_{t=1}^{T} P(h_i(t)|x(t-1), x(t-2), \ldots)\,. \end{aligned} \tag{5}$$

The factor $Q_w(h|v)$ describes the probability of a hidden activity pattern, when only considering the past hidden and visible activity in each time step. Note that in general $Q_w(h|v) \neq P_w(h|v) \equiv P_w(v,h)/\sum_h P_w(v,h)$, since in $P_w(h|v)$ the whole visible activity pattern (past and future) is considered in each time step. Similarly, $R_w(v|h)$ describes the probability of a visible activity pattern when considering only the past. To obtain samples $h \sim Q_w(h|v)$, one runs the natural dynamics for the hidden neurons (Eq. 1 including stochastic spiking) while the visible neurons are clamped to $v \sim P^*(v)$. Based on this sampling procedure we introduce the upper bound $\mathcal{F}$ of the KL-divergence,

$$\mathcal{F}(P^*(v)\|P_w(v)) = \left\langle \log \frac{P^*(v)}{R_w(v|h)} \right\rangle_{Q_w(h|v)P^*(v)} \geq \mathcal{D}(P^*(v)\|P_w(v))\,. \tag{6}$$

To prove the inequality we note that $P_w(h|v) = R_w(v|h)Q_w(h|v)/P_w(v)$. Using the definition and positiveness of the KL-divergence we find that $0 \leq \mathcal{D}(Q_w(h|v)\|P_w(h|v)) = \log P_w(v) - \langle \log R_w(v|h) \rangle_{Q_w(h|v)}$ and conclude that $-\langle \log R_w(v|h) \rangle_{Q_w(h|v)} \geq -\log P_w(v)\,.$ Averaging this last inequality across $P^*(v)$ and subtracting the target entropy $H^* = -\langle \log P^*(v) \rangle_{P^*(v)}$, we obtain Eq. 6.

Note that due to the KL-properties this bound is tight if and only if $Q_w(h|v) = P_w(h|v)$. This is the case if either the activity in the hidden neurons has no effect on the visible activity, i.e. $R_w(v|h) = P_w(v)$ and hence $P_w(h|v) = P_w(v,h)/P_w(v) = R_w(v|h)Q_w(h|v)/P_w(v) = Q_w(h|v)$, or if the dynamics in the hidden neurons is deterministic, i.e. $P_w(h|v) = Q_w(h|v) = \delta_{h,\mathrm{h}(v)}$ for some function $\mathrm{h}(v)$. Note that with $N_\mathrm{h} = (k-1)N_\mathrm{v}$, any factorizable Markov chain $\prod_i P_w(v_i(t)|v(t-1),\ldots,v(t-k))$ of order k, which can be parametrized by $P_w(v_i(t)|v(t-1),\ldots,v(t-k))$ (c.f. Eq. 1 and Eq. 2) is implementable with deterministic dynamics in the hidden neurons: the first group of N_v hidden neurons are driven by the visible neurons such that their activity is the same as the visible activity one time step in the past, the second group of N_v hidden neurons is driven by the first hidden group such that their activity is the same as the visible activity two time steps in the past and so forth, i. e. $h_{(k-1)N_\mathrm{v}+i}(t) = v_i(t-k)$ (see e.g. Fig. 3B).

Deriving the batch learning rule. The negative gradient of the upper bound $\mathcal{F}$ in Eq. 6 is evaluated to

$$-\frac{\partial \mathcal{F}}{\partial w_{ij}} = \left\langle \frac{\partial}{\partial w_{ij}} \log R_w(v|h) \right\rangle_{Q_w(h|v)P^*(v)} + \left\langle (\log R_w(v|h) - \bar{r}) \frac{\partial}{\partial w_{ij}} \log Q_w(h|v) \right\rangle_{Q_w(h|v)P^*(v)} \tag{7}$$

where $\bar{r}$ can be any arbitrary constant since $\left\langle \frac{\partial}{\partial w_{ij}} \log Q_w(h|v) \right\rangle_{Q_w(h|v)} = \sum_h \frac{Q_w(h|v)}{Q_w(h|v)} \frac{\partial}{\partial w_{ij}} Q_w(h|v) = \frac{\partial}{\partial w_{ij}} \sum_h Q_w(h|v) = \frac{\partial}{\partial w_{ij}} 1 = 0$. Taking advantage of this arbitrariness, the constant $\bar{r}$ can be chosen to minimize the variance of the second term in Eq. 7. Practically, this will be approximated by $\bar{r} \approx \langle \log R_w(v|h) \rangle_{Q_w(h|v)P^*(v)}$. To justify this choice we note that the optimal value for $\bar{r}$ can be found as follows: for $\langle e \rangle = 0$, $\mathrm{Var}((r-\bar{r})e) = \langle r^2e^2 - 2r\bar{r}e^2 + \bar{r}^2e^2 \rangle - (\langle re \rangle + \langle \bar{r}e \rangle)^2 \Rightarrow \frac{\partial}{\partial \bar{r}} \mathrm{Var}((r-\bar{r})e) = 2\langle re^2 \rangle - 2\bar{r}\langle e^2 \rangle \overset{!}{=} 0 \Rightarrow \bar{r} = \frac{\langle re^2 \rangle}{\langle e^2 \rangle}$. If $\langle re^2 \rangle \approx \langle r \rangle \langle e^2 \rangle$ one can approximate $\bar{r} \approx \langle r \rangle$. Note that in contrast to Eq. 4, the hidden spike sequences in Eq. 7 can now be naturally sampled.

To deduce from Eq. 7 a learning rule of the form $\Delta w_{ij} \propto -\partial \mathcal{F}/\partial w_{ij}$ we first have to distinguish between weights w_{ij} projecting onto visible neurons ($i = 1, \ldots, N_\mathrm{v}$) which we call *visible weights* and weights projecting onto hidden neurons ($i = N_\mathrm{v} + 1, \ldots, N$) – *the hidden weights*. Due to the conditioning, $R_w(v|h)$ does not depend on the hidden weights and $Q_w(h|v)$ does not depend on the visible weights (see Eq. 5). Hence, if the postsynaptic neuron i is visible, the second term in Eq. 7 vanishes, and if it is hidden, the first term vanishes. Given the explicit form of the log-likelihoods $\log R_w(v|h)$ and $\log Q_w(h|v)$ as in Eq. 2 we can directly take the derivatives in Eq. 7 and obtain the batch learning rule

$$\Delta w_{ij}^{\mathrm{batch}} = \eta \sum_{t=1}^{T} g_i(t) \left(x_i(t) - \rho_i(t)\right) x_j^{\epsilon}(t) \cdot \begin{cases} 1 & \text{if } i \text{ visible} \\ \\ (\log R_w(v|h) - \overline{r}) & \text{if } i \text{ hidden} \end{cases} \tag{8}$$

where η is the learning rate and $g_i(t) = \frac{\rho_i'(t)}{\rho_i(t)(1-\rho_i(t))}$, with $\rho_i'(t) = d\rho(u)/du|_{u=u_i(t)}$. For the sigmoid transfer function $\rho(u) = 1/(1+\exp(-\beta u))$, which is a reasonable and common choice when dealing with binary neurons, the prefactor $g_i(t)$ equals β, since $d\rho(u)/du = \beta\rho(u)(1-\rho(u))$. In batch learning (Eq. 8) the weights are adapted only after the presentation of several spike patterns v. The visible neurons are clamped with spike trains sampled from the distribution $v \sim P^*(v)$, and the hidden neurons follow the neural dynamics, $h \sim Q_w(h|v)$. As can be seen in Eq. 8, the learning rule is different for visible synapses and for hidden synapses. This stands in contrast to our previous work (Brea et al., 2011), where the learning rule is identical for both types of synapses.

In the absence of hidden neurons the learning rule is identical to the one proposed in Pfister et al. (2006) for feedforward networks. The generalization to recurrent networks with hidden neurons, which we consider here, is feasible because of the conditional independence of firing (Eq. 2, Eq. 5). It should be noted that, even though the learning problem was formulated as "minimize the KL divergence from target to model distribution", this minimization can be implemented with synapses that have only access to local information (presynaptic: $x_j^{\epsilon}(t)$, postsynaptic: $g_i(t)(x_i(t) - \rho_i(t))$) and in the case of hidden synapses one global signal ($\log R_w(v|h) - \overline{r}$). We will assume that each synapse has direct access to the postsynaptic spiking information $x_i(t)$ as well as the probability of spiking $\rho_i(t)$.

Online learning rule. To obtain an online learning rule which updates the synaptic weights at each time step, we need to replace the temporal summations in the batch rule (Eq. 8) by leaky integrators. For the synapse-specific part in Eq. 8 this leads to the synaptic eligibility trace

$$e_{ij}(t) = (1-\gamma_1)\, e_{ij}(t-1) + \gamma_1 g_i(t) \left(x_i(t) - \rho_i(t)\right) x_j^{\epsilon}(t) \tag{9}$$

with $\gamma_1 = 1/T$. Similarly, $\log R_w(v|h)$ is replaced by

$$r(t) = (1-\gamma_1)\, r(t-1) + \gamma_1 \sum_{i=1}^{N_\mathrm{v}} v_i(t) \log \rho_i(t) + (1-v_i(t)) \log(1-\rho_i(t))\,. \tag{10}$$

Note that $\log R_w(v|h)$ is given by the expression in Eq. 2 with the summation over neurons from 1 to N replaced by a summation over visible neurons 1 to N_v. The identical time constant $\gamma_1 = 1/T$ of the leaky integrators in Eq. 9 and in Eq. 10 reflects the fact that in Eq. 8 the terms $g_i(t)(x_i(t)-\rho_i(t))x_j^\epsilon(t)$ are summed over $t = 1,\ldots,T$ and so are the terms in $\log R_w(v|h)$ (c.f. Eq. 2). Finally, the term $\langle \log R_w(v|h)\rangle_{Q_w(h|v),P^*(v)}$ can be estimated as

$$\overline{r}(t) = (1-\gamma_2)\overline{r}(t-1) + \gamma_2 r(t-1)\,. \tag{11}$$

where the time constant γ_2 of this leaky integrator is much larger than γ_1 in order to estimate the average $\langle \log R_w(v|h)\rangle_{Q_w(h|v),P^*(v)}$. With those dynamic quantities, the synaptic learning rule in Eq. 8 now becomes an online rule of the form

$$\delta w_{ij}(t) = \eta\, e_{ij}(t) \cdot \begin{cases} 1 & \text{if } i \text{ visible} \\ (r(t)-\overline{r}(t)) & \text{if } i \text{ hidden} \end{cases} \tag{12}$$

In Eq. 9 the term $g_i(t)\,(x_i(t)-\rho_i(t))$ corresponds to the postsynaptic and $x_j^\epsilon(t)$ to the presynaptic contribution to the weight change. The product of these two terms is low-pass filtered (Eq. 9) to form the eligibility trace, which in the case of hidden neurons is multiplied by the global factor $(r(t)-\overline{r}(t))$ (see Eq. 12). It is interesting to note that the global factor $(r(t)-\bar{r}(t)) \simeq (\log R_w(v|h) - \langle \log R_w(v|h)\rangle)$ can be seen as an “internal reward signal”, which depends on how much more than the average a given hidden activity h helps to produce the visible activity v. We call the signal internal, since it is provided by the visible neurons and not by an external teacher.

Initial condition of the dynamics. In the derivation of the learning rule we assumed that $\rho_i(t)$ (see Eq. 2 and surrounding paragraph) is given at every moment in time t. For times t larger than the neural time constant, $\rho_i(t)$ is fully determined by the spiking activity of the network. However, at the beginning of each pattern presentation or recall, $\rho_i(t)$ would also be influenced by spikes that occured earlier, i.e. before time $t=1$. In practice, for patterns with only visible neurons we took the spikes of the last pattern presentation or recall to initialize the dynamics. In systems with hidden units the whole system was effectively clamped to a predefined spatio-temporal pattern before each pattern presentation or recall, except in Fig. 2E, where the system also converged without reset to the inital state.

Recall. Recall means sampling from the model distribution $P_w(x)$ and in particular that the visible activity patterns are distributed as $P_w(v)$. Therefore, the learning rule minimizes the upper bound $\mathcal{F}(P_w(v)\|P_w(v)) \geq \mathcal{D}(P_w(v)\|P_w(v)) = 0$ during recall. If the bound is tight, i.e. $\mathcal{F}(P_w(v)\|P_w(v)) = \mathcal{D}(P_w(v)\|P_w(v)) = 0$, the gradient $\nabla\mathcal{F}(P_w(v)\|P_w(v))$ equals zero, because $\mathcal{F}(P_w(v)\|P_w(v))$ is in a local minimum. Therefore the weight change is zero on average, i.e. $\langle\Delta w\rangle = 0$. However, unless $P_w(x)$ is deterministic, the variance $\mathrm{Var}(\Delta w)$ is not zero and therefore diffusion takes place. If $\mathcal{F}(P_w(v)\|P_w(v)) > \mathcal{D}(P_w(v)\|P_w(v))$ an additional drift is expected to occur. For the simulations in Fig. 3 we compared the time it takes to "forget", i.e. drift away from the final $\mathcal{D}$ value by 0.1 bits, with the time it takes to "relearn" the target. For Markovian targets ($k = 1$) learning was approximately 100 times faster than forgetting. For non-Markovian targets ($k > 1$), where performance heavily depends on the hidden weigths, learning was approximately 6 times faster than forgetting.

Linear separability and Markovianity of sequences. In the one time step response kernel scenario of Fig. 2 with deterministic target sequences the notion of linear separability and Markovianity proves helpful for classification. Suppose the target sequence requires a neuron i at time t to be active $x_i(t) = 1$ (silent $x_i(t) = 0$). This means that during recall, neuron i should get a positive (negative) input at time t, i.e. $\sum_j w_{ij}x_j(t-1) > 0$ (< 0). This puts a constraint on the weights w_{ij}. If synaptic strengths w_{ij} exist that respect these constraints for all times t and neurons i, the sequence is called linearly separable. There exist many methods to test linear separability (Elizondo, 2006).

Is the proposed learning rule capable to find the weights for a linearly separable sequence? The answer is yes. Firstly, since we know that w^* exists, the smallest possible divergence from target to model distribution is zero $\mathcal{D}(P^*(x)\|P_{w^*}(x)) = 0$. Secondly, it can be shown that this divergence is convex in the synaptic weights (Barber, 2011). Therefore a suitable stochastic gradient descent will lead to a perfect solution with $\mathcal{D}(P^*(x)\|P_w(x)) = 0$.

We call a deterministic sequence Markovian if any state in the sequence depends only on the previous state, i.e. $x(t) = f(x(t-1))$, for some function f and hence $x(t) = x(t') \Rightarrow x(t+1) = x(t'+1), \forall t, t'$. Note that all linearly separable sequences are Markovian: a non-Markovian sequence contains transitions with the same initial state but different final states, i.e. for some t and t' it contains the subsequences

$\ldots x(t)x(t+1)\ldots$ and $\ldots x(t')x(t'+1)\ldots$ with $x(t) = x(t')$ but $x(t+1) \neq x(t'+1)$, for which there do not exist any synaptic weights that satisfy the linear separability constraints, since for a neuron i for which e.g. $1 = x_i(t+1) \neq x_i(t'+1) = 0$ there are no w_{ij} such that $0 < \sum_j w_{ij}x_j(t) = \sum_j w_{ij}x_j(t') < 0$; hence there is no non-Markovian sequence that is linearly separable and therefore any linearly separable sequence is Markovian.

Markovian sequences that are not linearly separable require appropriate activity states in a hidden layer (Rigotti et al., 2010) such that the whole sequence of visible and hidden states becomes linearly separable. The minimal number of required hidden neurons depends on the problem. But $N_\mathrm{h} = T$ hidden neurons are always sufficient, since it is always possible to find N_h linearly separable states in N_h dimensions. Required are synaptic connections from visible to hidden and hidden to visible. In non-Markovian sequences the visible state at time t does not only depend on the visible activity in the previous time step $t-1$ but depends potentially on earlier activity states, i.e. in time steps $t-2, \ldots$. In this case the activity in the hidden layer can be seen as representing the context in that it carries some information about past visible activity states. Consequently non-Markovian sequences require connections between hidden neurons.

Link to the voltage-triplet rule. In the limit of continuous time, the learning rule for visible synapses can be written as a triplet potentiation term (2 post, 1 pre) and a depression term (1 post, 1 pre):

$$\dot{w}_{ij} = \eta g_i(t) x_i(t) x_j^\epsilon(t) - \eta \rho_i'(t) x_j^\epsilon(t) \tag{13}$$

where $x_i(t)$ denotes the Dirac spike train of neuron i, $\rho_i'(t) = d\rho(u)/du|_{u=u_i(t)}$ denotes the derivative of the firing intensity function and the prefactor is defined by $g_i(t) = \frac{\rho_i'(t)}{\rho_i(t)}$. Note that in continuous time the prefactor has a slightly different form than in discrete time: to arrive at a continuous time description we explicitly introduce the time bin size δt, set the probability of spiking in one time bin to $\rho_i(t)\delta t$, thereby reinterpreting $\rho_i(t)$ as a spike density function, and get in the limit of vanishing time bin size $\lim_{\delta t \to 0} \frac{\rho_i'(t)\delta t}{\rho_i(t)\delta t(1-\rho_i(t)\delta t)} = \frac{\rho_i'(t)}{\rho_i(t)}$ (see Pfister et al. (2006); Brea et al. (2011) for details). Interestingly, formulated in this way, the learning rule closely resembles the voltage-triplet rule proposed by Clopath et al. (2010) - which is an extension of the pure spike-based triplet rule (Pfister and Gerstner, 2006). The weight change prescribed by the voltage-triplet rule of Clopath et al. (2010), which we will compare to our rule (Eq. 13) can be also written as a post-post-pre potentiation term and a post-pre depression term:

$$\dot{w}_{ij} = A_3 \lfloor u_i^\alpha(t) - \theta_1 \rfloor \lfloor u_i(t) - \theta_2 \rfloor x_j^\beta(t) - A_2 \lfloor u_i^\gamma(t) - \theta_1 \rfloor x_j(t) \tag{14}$$

where the notation $\lfloor\cdot\rfloor$ denotes rectification, i.e. $\lfloor x\rfloor = x$, if $x \geq 0$, otherwise $\lfloor x\rfloor = 0$. The convolution kernels α, β and γ are exponential decay kernels with time constants τ_α (resp. τ_β, τ_γ), e.g. $\alpha(s) = \tau_\alpha^{-1}\exp(-s/\tau_\alpha)\Theta(s)$ where $\Theta(s)$ denotes the Heaviside function, i.e. $\Theta(s) = 1$ for $s \geq 0$ and $\Theta(s) = 0$ otherwise.

The correspondence between our learning rule for visible synapses of Eq. 13 and the voltage-triplet rule (Clopath et al., 2010) in Eq. 14 becomes tighter under the following observations and assumptions. Firstly, it should be noted that in the voltage-triplet model, the threshold is high such that $\lfloor u_i(t) - \theta_2\rfloor$ is non zero only at the timing of the spike. Therefore this term can be replaced by our Dirac spike train $x_i(t)$. Secondly, we can easily assume that the response kernel $\epsilon(s)$ matches the $\gamma(s)$ convolution kernel. Indeed in Clopath et al. (2010), the time constant τ_γ (between 15 an 30 ms) was fitted to recordings. Thirdly, in the limit of fast time constants τ_α, $\tau_\gamma \to 0$ (which is reasonable since those time constants in Clopath et al. (2010) are smaller than 10 ms), the low-pass filtered versions of the membrane potential can be replaced by their instantaneous value: $u_i^\alpha \ \to u_i$, (resp. $u_i^\gamma \to u_i$). Finally, if we choose a gain function given by $\rho(u) = g_0\lfloor u-\theta\rfloor^2 + \nu_0$, the factor $\rho'(u) = 2g_0\lfloor u-\theta\rfloor$ is a rectified linear function consistent with the voltage triplet model and the factor $\rho'(u)/\rho(u) = 2g_0(u-\theta)/(g_0(u-\theta)^2+\nu_0)\Theta(u-\theta)$ is close to a rectified linear function for $u \ll \theta + \sqrt{\nu_0/g_0}$ since it is zero for $u < \theta$ and starts linearly (with a slope $2g_0\nu_0^{-1}$) for $u > \theta$. Note that any rectified polynomial gain function would work as well (with ρ'/ρ differentiable at $u = \theta$ for power of the polynomial $p \geq 2$).

Simulation details

In all learning curve plots the measures were normalized by the number of visible neurons and time steps. If for all time steps and neurons the probability to be active is $P(x_i(t) = 1) = 0.5$, which is the case for $u_i(t) = 0$ and the sigmoidal spike probability function, the normalized Kullback-Leibler divergence from target to model distribution equals one bit minus the normalized entropy of the target distribution. In all simulations the initial weights were chosen to be zero, i.e. $w_0 = 0$.

Fig. 2A: Number of neurons $N = N_\mathrm{v} = 10$, initial weights $w_{ij} = 0$, learning rate $\eta = 50$, parameter $\beta = 0.2$, resting potential $u_0 = 0$, learning phase of 1000 target sequence presentations.

Fig. 2B: The fraction of learnable patterns μ was estimated as the mean of the posterior $P(\mu|D(x_1^*),\ldots,D(x_l^*)) \propto \prod_{i=1}^{l} P(D(x_i^*)|\mu)$ with flat prior $P(\mu) = \mathrm{Uniform}([0,1])$, where $D(x_i^*)$ tells whether the sequence x_i^* is learnable or not ($l = 100$). The target distribution are delta distributions $P^*(x) = \delta_{x,x^*}$, where each x_i^*

was sampled from a uniform distribution, i.e. $x_i^*(t) \sim P(x_i(t) = 1) = 0.5$. Linear separability, which is the criterion for learnability, was tested with the linear programming method provided by Mathematica (see (Elizondo, 2006) for details). For the asymmetric Hebb rule, $w_{ij} = 1/T \sum_t (2x_i(t+1)-1)(2x_i(t)-1)$, we tested, whether these weights lead to perfect recall of the sequence. Note that this measure is more stringent than for example the average Hamming distance between target and recalled pattern.

Fig. 2D: The same procedure as in Fig. 2B was applied to temporally correlated target patterns. We used patterns with $N_\mathrm{v} = 100$ neurons and $T = 5$ time steps. To generate patterns with correlation length α, we choose an initial state $x(0)$ with $P(x_i(0) = 1) = 0.5$ and subsequent states $x(t)$ with probabilities $P(x_i(t) = x_i(t-1)|x_i(t-1)) = 2^{-\frac{1}{\alpha+1}}$. This is equivalent to the more intuitive interpretation of the correlation length as the number of time steps needed until the patterns are uncorrelated, i.e. $P(x_i(\alpha+1) = x_i(0)|x_i(0)) = 0.5$.

Fig. 2E: Only visible: number of neurons $N = N_\mathrm{v} = 10$, parameter $\beta = 0.2$, resting potential $u_0 = 0$, learning rate $\eta = 0.1$. Batch algorithm: number of neurons $N_\mathrm{v} = 10, N_\mathrm{h} = 10$, parameter $\beta = 0.1$, learning rate $\eta = 0.1$, estimate of $\bar{r}$ and update of weights every 25 target pattern presentation. Online algorithm: number of neurons $N_\mathrm{v} = 10, N_\mathrm{h} = 10$, learning rate $\eta = 0.5$, first low-pass filter time constant $\gamma_1 = 1/12$, second low-pass filter time constant $\gamma_2 = 1/120$, the low-pass filter variables e_{ij}, r, $\bar{r}$ were initialized with zeros; during the first 100 pattern presentations no update of the hidden weights took place to omit transient effects of the low-pass filter dynamics. In the batch simulation the state of the hidden neurons was reset to a given state h_1 (indicated by the green line) after each target pattern presentation and before each recall. In the online simulation the hidden states were never reset. In all simulations the initial weights were set to zero, i.e. $w_{ij} = 0$, the learning phase consisted of 2.5×10^4 target pattern presentations.

Fig. 3A: Target distribution $P^*(v) = P_{w^*}(v)$ with w_{ij}^* drawn from a normal distribution with mean zero and standard deviation 5, number of visible neurons $N_\mathrm{v} = 5$, number of hidden neurons $N_\mathrm{h} = 0$ for target and model with only visible, $N_\mathrm{h} = 15$ for model with hidden neurons, parameter $\beta = 2/\sqrt{N}$, learning rate for visible weights $\eta_v = 4 \times 10^{-4}$, learning rate for hidden weights $\eta_h = 10^{-4}$, resting potential $u_0 = 0$, initial model weights $w_0 = 0$, $P(x_0) = \delta_{x_0,x_0*}$. Training data consisted of 10^7 patterns of length 20 time steps, which were generated by the target distribution by running the neural dynamics with target weights w^*. The transition frequency data was obtained empirically by generating 10^4 samples of 20 timesteps.

Fig. 3B: Same as in Fig. 3A but for a different target. The construction of the $k = 3$ Markov target is explained in the second paragraph after Eq. 6 and in the caption of Fig. 3B. We excluded target weights w^* that parametrize a distribution with highly correlated subsequent visible states, i.e. $\langle(2v_i(t+1)-1)(2v_i(t)-1)\rangle_{t,i,P_{w^*}(x)} > 0.8$, since such distributions can be accurately approximated with a Markovian model.

Fig. 3C: For each k, results were obtained for 16 different target weight matrices w^* and initial conditions x^*. For simulations with static hidden weights the visible to hidden and hidden to hidden weights were the randomly reordered final weights from the corresponding simulation with plastic hidden weigths, the initial visible weights were set to zero and trained with the same target.

In Fig. 4 and Fig. 5 we identify one simulation timestep with 1 ms.

Fig. 4A: Number of neurons $N = N_{\mathrm{v}} = 100$, response kernel time constants $\tau_1 = 10$ ms, $\tau_2 = 2$ ms, adaptation time constant $\tau_r = 15$ ms, adaptation kernel amplitude $c = -10$, parameter $\beta = 1/3$, learning rate $\eta = 0.5$, resting potential $u_0 = -5$, initial model weights zero, target weights $w^*_{ij} = 20$, if $i \in \{10(l+1)+1, \ldots 10(l+2)\}$ and $j \in \{10l+1, \ldots 10(l+1)\}$, $w^*_{ij} = -5$, otherwise. Training data consisted of 10^6 states generated by the target distribution by running the neural dynamics with target weights w^*.

Fig. 4B: Number of neurons $N = N_{\mathrm{v}} = 20$, response kernel time constants $\tau_1 = 10$ ms, $\tau_2 = 2$ ms, adaptation time constant $\tau_r = 3$ ms, adaptation kernel amplitude $c_{\mathrm{adapt}} = -200$, parameter $\beta = 0.1$, learning rate $\eta = 100$, resting potential $u_0 = -5$, initial model weights zero, delta target distribution $P^*(x) = \delta_{x,x^*}$, with x^* drawn from a distribution with $P(x^*_i(t) = 1) = 0.1$, the learning phase consisted of 2×10^4 target pattern presentations. For recall, the system is always initialized in the same way.

Fig. 4C: Number of neurons $N = N_{\mathrm{v}} = 10$, fast response kernel kernel time constants $\tau_1 = 15$ ms, $\tau_2 = 7$ ms, slow response kernel time constants $\tau_1 = 50$ ms, $\tau_2 = 30$ ms, fast adaptation kernel time constant and amplitude $\tau_r = 7$ ms, $c = -50$, slow adaptation kernel time constant and amplitude $\tau = 30$ ms, $c = -5$, parameter $\beta = 0.2$, learning rate $\eta = 10$, $u_0 = -5$, initial model weights zero. Target spike trains were generated by injecting strong external positive currents in high firing rate phases and negative currents in low firing rate phases, which we modeled using neural dynamics given by $u_i(t) = u_0 + x^\kappa_i(t) + u_{\mathrm{ext}}$, where $u_{\mathrm{ext}} = +50(-50)$ in high (low) firing rate phases of the neuron and the sigmoidal spiking firing probability. Each high (low) firing rate phase had a duration of 50 ms. The training phase consisted of 3×10^3 samples drawn from the target distribution.

Fig. 5A: The change in synaptic strength Δw after 50 pairings of one presynaptic with one postsynaptic

spike on an interval of 200 ms was computed for different adaption kernels $\kappa(t) = ce^{-t/\tau}$ with $\tau = 10$ ms (black: $c = 2$, blue: $c = 0$, red: $c = -10$) while keeping the synaptic response kernel $\epsilon(t) = \frac{1}{\tau_1 - \tau_2}(e^{-t/\tau_1} - e^{-t/\tau_2})$, with $\tau_1 = 10$ ms and $\tau_1 = 2$ ms. The initial value of the membrane potential before stimulation was set to the resting potential $u_0 = -5$. Inital weight was $w = 10$.

Fig. 5B: The data were fitted with a model given by the probability function $\rho(u) = g_0 \lfloor u - \theta \rfloor^2 + \nu_0$, and response kernel $\alpha(s) = \tau_\alpha^{-1} \exp(-s/\tau_\alpha)\Theta(s)$ (see paragraph "Link to the voltage-triplet rule"). The fitted parameters are given by: $g_0 = 0.94 \cdot 10^{-2}$ Hz, $\nu_0 = 7.9$ Hz, $\theta = -36.2$, $\tau_\alpha = 41.5$ ms, $w_0 = 1$, $\eta = 0.46$.

All simulations were performed with Mathematica on personal computers, except the simulations in Fig. 3, which were written in C and the fit in Fig. 5B, which was performed with Matlab.

3 Results

We studied the task of learning to spontaneously produce spiking activity with a given statistics: stimuli make neurons fire in a specific order (Fig. 1A target); in the absence of any stimulus, neurons spike spontaneously; before learning, the spontaneous activity patterns do not resemble the stimulus-evoked activity patterns (Fig. 1A before learning). The goal of learning is to change the network dynamics such that after learning the spontaneous activity patterns resemble the stimulus-evoked activity patterns (Fig. 1A after learning). To derive plasticity rules that solve this learning task, we chose the neural dynamics and minimized with respect to synaptic weights a distance measure between stimulus-evoked and spontaneous activity (see Materials and Methods and Fig. 1B). In this way, the learning rule is matched to the neural dynamics.

To study the proposed learning rules we first elaborate on deterministic target sequences learned with the simplest variant of our model. Deterministic target sequences are of behavioral relevance, since spatio-temporal spike pattern distributions are presumably often sharply peaked in the sense that the typical spike patterns closely resemble each other.

Deterministic target sequences

In the simplest conceivable form of our model the shape of the postsynaptic kernel ϵ is a unit rectangular for one delayed time step and no adaptation takes place. This stochastic McCulloch-Pitts neuron (Mc-

Culloch and Pitts, 1943) is of widespread usage in artificial neural networks (Baum and Wilczek, 1987; Düring et al., 1998; Barber, 2011).

In Fig. 2A (target) we show a sequence, which is learnable with only visible neurons. During learning, the stimuli activated the neurons repetitively in the order shown in the figure. After learning, spontaneous activity reproduced the target sequence including the transition from the last to the first spatial activity pattern. This fact is also reflected in the learning curve in Fig. 2A, which shows that the model distribution approximated the target distribution almost perfectly.

Which sequences are learnable with only visible neurons and which require hidden neurons? Since at each moment in time a neuron can be either active or silent, the total number of visible sequences is $2^{N_\mathrm{v}T}$, for a given number of visible neurons N_v and sequence length T. We can group these sequences using as criterion either linear separability or Markovianity (see Materials and Methods). It turns out that this grouping helps to formulate minimal architectural requirements, as summarized in Fig. 2C: linearly separable sequences can be learned with only visible neurons and synaptic connections between them, Markovian sequences require enough (at most $N_\mathrm{h} = T$) hidden neurons and at least synaptic connections from visible to hidden and hidden to visible, and non-Markovian sequences demand additionally connections between hidden neurons.

The sequence shown in Fig. 2A is linearly separable and thus can be learned with only visible neurons. The sequence in Fig. 2E is non-Markovian: state marked by a blue triangle occurs twice in the sequence. Furthermore, already the second state (red triangle), where all neurons are silent, renders the sequences not linearly separable. This is a consequence of our choice of coding (1: active, 0: silent) and adaptable parameters (only synaptic weights w_{ij}), which we used to account for biological constraints. This sequence can only be learned with appropriate recurrent connections of the hidden neurons. We show that both the batch and the online learning rule find appropriate synaptic weights by stochastic gradient descent on the upper bound $\mathcal{F}$ of the divergence $\mathcal{D}$ from target to model distribution. This bound becomes tight, i.e. $\mathcal{F} = \mathcal{D}$, towards the end of learning (see Materials and Methods), which is reflected in the diminishing distance between the $\mathcal{F}$- and $\mathcal{D}$- curves in Fig. 2E.

What is the typical size of the subset of linearly separable – and thus learnable – sequences for given N_v and T? In Fig. 2B we show the fraction of learnable sequences as a function of the relative sequence length T/N_v for different N_v. Below a relative sequence length of approximately $T/N_\mathrm{v} \approx 1.5$ the probability for the sequence to be linearly separable is very close to 1. The critical value $T/N_\mathrm{v} \approx 1.5$ is again a consequence of our choice of coding. For (1: active, -1: silent) coding we expect the critical relative sequence length to be at $T/N_\mathrm{v} \approx 2$ (Hertz et al., 1991). As a reference we compare this to a simple temporal Hebb rule $\Delta w_{ij} = \frac{1}{T}\sum_{t=1}^{T}(2x_i(t+1)-1)(2x_i(t)-1)$ as proposed in Hopfield (1982); Düring et al. (1998); Grossberg (1969); Herz et al. (1988); Gerstner et al. (1993); Horn et al. (2000). It is not surprising that this simple temporal Hebb rule, for which linear separability is not a sufficient condition for a pattern to be learnable, in general does not find synaptic weights to perform perfect recall: instead of solving an optimization problem (minimizing the divergence from target to model distribution), which leads to a learning rule that is matched to the dynamics, it is simply inspired by the symmetric Hebb rule used in Hopfield networks (Hopfield, 1982). Its weakness becomes even clearer when we consider correlated patterns (see Fig. 2D): the simple temporal Hebb rule performs well only for uncorrelated patterns whereas the probability of linearly separable sequences decreases slowly with increasing correlation.

Stochastic target

Some stimuli might evoke spike patterns that do not follow a sharply peaked distribution. Instead, the spike patterns look quite different each time a sample from the target distribution $P^*(x)$ is drawn. Even though they do not look similar, there might be some temporal dependencies in the sense that the spiking activity $x(t)$ at time t depends on earlier spiking activity states $x(t-k), k > 0$. Variability in the target patterns can arise due to many different reasons. Firstly, there is an extrinsic source of variability, as the same stimulus can be presented with small variations. Secondly, neurons and synapses are intrinsically noisy, so the neural network that provides the teaching signal will be subject to noise. Interestingly, in the proposed framework the level of noise during recall can be matched to the target variability through synaptic plasticity.

As we showed in Materials and Methods (second paragraph after Eq. 6) it is possible to implement a large class of stochastic target distributions, namely factorizable Markov chains $\prod_i P_w(v_i(t)|v(t-1),\ldots,v(t-k))$ of order k, which can be parametrized by $P_w(v_i(t)|v(t-1),\ldots,v(t-k))$ (c.f. Eq. 1 and Eq. 2). For Markovian targets ($k=1$) and a model without hidden neurons, the solution is unique due to the convexity of the KL divergence. In Fig. 3A we demonstrate in an example that the learning rule effectively leads to the solution. For $k > 1$ a model with only visible neurons fails to learn the target, whereas a model with sufficient hidden neurons equiped with our learning rule succeeds (Fig. 3B). Under the light of the remarkable capabilities of reservoir computing (Maass and Sontag, 2000; Lukoševičius and Jaeger, 2009; Rigotti et al., 2010) it is interesting to note that a random choice of static hidden weights obtained by reshuffling learned weights is not sufficient to learn the target (Fig. 3C).

Alpha-shaped kernels

So far we discussed the plasticity rule for a model with very simple dynamics. However, the neuron model in Eq. 1 allows for more realistic dynamics. In the examples shown in Fig. 4, a presynaptic spike evokes an alpha-shaped synaptic response ϵ (see Fig. 4B), which is felt by the postsynaptic neuron, and an adaptation κ, that prevents immediate follow-up spikes in the pre-synaptic cell. With this dynamics it is possible to learn a “synfire chain” (Fig. 4A) or a pattern of precise spike times (see Fig. 4B). Depending on the target distribution and the neural dynamics, timing errors may propagate and the pattern eventually becomes unstable (Diesmann et al., 1999). This fading of precision is less problematic with a rate code.

In Fig. 4C (target) the pattern of Fig. 2A is encoded with periods of high and low firing rate. The system can reasonably well learn this target with an alpha-shaped synaptic response ϵ_1 and adaptation kernel κ (Fig. 4, model with ϵ_1, blue learning curve). Fast synaptic responses ϵ_2 help to further stabilize the pattern and increase performance (Fig. 4, model with both ϵ_1 and ϵ_2, red learning curve). The motivation for using synaptic responses on two different timescales arises from the idea that fast connections establish attractors which encode for the different elements in the sequence and slow responses push the dynamics from one attractor to the next (Sompolinsky and Kanter, 1986). Neurons could implement fast and slow responses to one presynaptic spike with different types of synapses, e.g. fast AMPA synapses and slow NMDA synapses.

Biological plausibility

When learning is formulated with a computational goal in mind, like minimizing the difference between target and model distribution, it is far from obvious that the resulting learning rule is biologically plausible in the sense that it respects constraints of biological systems and is consistent with experimental data. Here we argue that the proposed learning rule in Eq. 12 is biologically plausible.

The learning rule in Eq. 12 respects causality, is implemented in an online fashion and depends on pre- and postsynaptic activity and a modulatory factor for hidden synapses. The pre-post term shares similarities with Spike-Timing Dependent Plasticity (STDP): Fig. 5A shows the STDP window predicted by the learning rule (see also (Pfister et al., 2006; Brea et al., 2011; Bi and Poo, 1998)). Note that this learning rule, which minimizes the divergence in Eq. 3 from target to model, predicts that the causal part of the STDP window should be matched to the dynamics of the synaptic transmission. The acausal part depends on the adaptation properties of the neuron. Interestingly, the learning rule is closely related to the voltage-triplet rule (Clopath et al., 2010; Pfister and Gerstner, 2006) (see Materials and Methods) and is compatible with the frequency dependence of STDP as observed by Sjöström et al. (2001) (see Fig. 5B).

For hidden synapses, the plasticity rule does not depend only on the pre- and postsynaptic activity, but is also modulated by a global factor. This global factor could be implemented by astrocytes for mainly three reasons. Firstly, it has been shown that astrocytes - which contact a large number of synapses - can modulate synaptic plasticity (Henneberger et al., 2010; Min and Nevian, 2012). Similarly, in our framework, the semi-global modulating factor (see Eq. 12) affects the plasticity of a large group of synapses (the hidden synapses). Secondly, according to our model, this global factor (see Eq. 12) acts at a slower time constant than the membrane time constant, which is consistent with the calcium dynamics of astrocytes (Di Castro et al., 2011). Finally, a causal pairing (pre-then-post) can lead either to long-term potentiation or long-term depression depending on the sign of the global factor $(r(t) - \bar{r}(t))$. This is in agreement with the study of Panatier et al. (2006) who showed that when D-serine - which is a NMDA co-agonist - is released by astrocytes, long-term depression can be turned into long-term potentiation. Unlike what is suggested by Panatier et al. (2006), this type of sliding threshold ($\bar{r}(t)$ in our case) is conceptually different from the one in the BCM learning rule (Bienenstock et al., 1982) since it is a global quantity whereas in the BCM learning rule the sliding threshold is a local quantity that depends on the history of the postsynaptic activity. Note that the proposed global factor depends on activity in visible neurons and affects hidden synapses only. Thus, if astrocytes are responsible for this signalling, they “need to know” which neurons are visible and which are hidden. This might be possible due to geometrical arrangement or chemical signalling.

4 Discussion

Learning a statistical model for temporal sequences with hidden units is a challenging machine learning task in itself. Here we considered an even harder problem in that we are interested in a biologically plausible learning rule that can solve this task. In order to be biologically plausible, the learning rule has to be local, causal and be consistent with experimental data. We derived a biologically plausible learning rule that minimizes by stochastic gradient descent an upper bound of the KL divergence from target to model distribution.

Because the proposed learning rule is minimizing an upper bound of the KL-divergence and not the

KL-divergence itself and because the tightness of the bound is not explicitly controlled - unlike in the Helmholtz framework (Dayan et al., 1995), the maxima of the two different objective functions could in principle be located at different places in the parameter space. It is however interesting to note that after some learning time, the bound becomes tighter, as can be seen on Fig. 2E where the KL-divergence between the proposal distribution $Q_w(h|v)$ and the posterior distribution $P_w(h|v)$ decreases. In fact, we can show that at the beginning of learning and at the end of learning of deterministic patterns, the bound is tight. Indeed, at the beginning of the learning all the weights (and in particular the weights towards visible neurons) are initialized to zero and therefore, the function $R_w(v|h)$ is independent of the hidden activity and thus $\mathcal{D}$ and $\mathcal{F}$ are identical. At the end of learning, for a deterministic pattern, the proposal distribution is a Dirac delta distribution and therefore $\mathcal{D} = \mathcal{F}$.

It is interesting to note that the learning rule for the hidden neurons in Eq. 12 appears formally similar to a reward-based learning rule (Pfister et al., 2006), but now the reward is an internally computed quantity and does not depend on an external reward. Loosely speaking, if the hidden units contribute to make the visible spikes likely, the synapses targeting those hidden units receive an internal reward signal $(r(t) - \bar{r}(t))$ (Eq. 12).

The formulation of learning as minimizing the KL-divergence (or, equivalently, maximizing the log-likelihood) or an upper bound thereof is common practice in machine learning. The novelty of our approach lies in the specific choice of the upper bound of the KL-divergence which relies on the assumption of conditional independence for the neural firing, i.e. given its membrane potential, the probability of firing of a neuron is conditionally independent of the firing of all other neurons at the same time. Even though this assumption is perfectly reasonable and widely used (e.g. in the GLM framework (Pillow and Latham, 2008)), it is the key assumption that allows the joint distribution over visible and hidden activity to be expressed as a product of a distribution from which it is easy to sample $(Q_w(h|v))$ and a function which is easy to calculate $(R_w(v|h))$. This stands in contrast to temporal Boltzmann Machines (Sutskever et al., 2009) where this assumption is not made and sampling usually involves running a Monte Carlo Markov Chain in each time step, which is hard to justify under the light of biological plausibility.

Another approach to learn a statistical model of spatio-temporal spike pattern with visible and hidden neurons is a generalization of the Expectation-Maximization algorithm proposed by Pillow and Latham (2008). Yet, as a version of the Helmholtz machine, the distribution, from which the hidden states are sampled, uses a different parametrization for storage (recognition; wake phase of learning) and recall (generation; sleep phase of learning) and assumes an acausal kernel, which renders the model unsuitable for a biologically realistic implementation. To circumvent the need of sampling the hidden states, $h \sim P_w(h|v)$ (see Eq. 4), Rezende et al. (2011) proposed to calculate explicitly the expectation under the posterior distribution $P_w(h|v)$. In order to achieve this, they had to assume a weak coupling between neurons and then approximate the true posterior distribution by a Gaussian Process on the membrane potential. Unfortunately, this weak coupling assumption is difficult to justify at the end of learning where individual weights can become large.

A drawback of the proposed learning rule for systems including hidden neurons is the potentially long learning time. The plasticity rule relies on stochastic gradient ascent by sampling hidden sequences $h \sim Q_w(h|v)$. As in any gradient descent method, the learning time depends on the learning rate and the initial condition: the learning time is short if at the beginning of learning the weights projecting onto hidden neurons are such that the samples $h \sim Q_w(h|v)$ help to quickly reduce the difference measure between target and model distribution and can be long otherwise. In some cases it is even possible to find initial weights that supersede any further learning of hidden weights. For example any possible sequence as discussed in Fig. 2C could be learned with a (hardwired) delay line of length T with $N_\mathrm{h} = T$ hidden neurons and no learning of the hidden weights. target distribution or in the presence of noise. Alternatively, the delay line could be implemented by a large enough number of randomly connected hidden neurons (Maass and Sontag, 2000; Lukoševičius and Jaeger, 2009; Rigotti et al., 2010) where the weights are chosen from a given distribution. The number of randomly connected hidden neurons needed might be very large to guarantee good solutions in any case. Since the goal of this paper was rather to demonstrate that the learning rule is capable of learning distributions which could not be learned with only visible neurons, we did not make use of elaborated choices of the initial conditions and started all simulations with inital weights $w_0 = 0$.

Our paper is not the first one to propose the idea that storage and recall should be matched. Indeed Sommer and Dayan (1998); Lengyel et al. (2005) already proposed that there should be a tight link between the plasticity rule and the neural dynamics. Interestingly their approach is complementary to the one presented here. Indeed, they start from a given plasticity rule and then derive the optimal (Bayesian) recall dynamics for this given plasticity rule. Here we are following the opposite path. We start from the neural dynamics and then derive the plasticity rule. Given the richness and the accuracy of existing neural models and given the absence of a canonical model of synaptic plasticity, we preferred to start from what is well known and derive predictions on what is largely unknown.

An interesting outcome of our model is that the learning rule for hidden synapses does not depend only on the pre- and postsynaptic activity, but is also modulated by a global factor. We argued in this paper that this global factor could be provided by astrocytes. To experimentally test this hypothesis we note that the global factor is predicted to depend on the voltage of the visible neurons. In particular, independently of the precise implementation of the model (be it by minimizing an upper bound on the KL divergence as presented here or by directly minimizing KL divergence itself as in Brea et al. (2011)), the global factor crucially depends on the presynaptic membrane potential at the time of the spike (see Eq. 10). So the key experimental step would be to show that astrocytic activity depends on the membrane potential of the presynaptic neuron at the time of the spike. This prediction seems plausible since it was found that a depolarization of the presynaptic membrane potential at the time of the spike causes a larger postsynaptic potential (Alle and Geiger, 2006; Shu et al., 2006). We suggest that astrocytes could have a similar sensitivity to the membrane potential at the time of the spike.

References

Ackley D, Hinton G, Sejnowski T (1985) A learning algorithm for boltzmann machines. *Cognitive science* 9:147–169.

Alle H, Geiger JRP (2006) Combined Analog and Action Potential Coding in Hippocampal Mossy Fibers. *Science* 311:1290–1293.

Amari SI (1972) Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers* 21:1197–1206.

Barber D, Agakov F (2002) Correlated sequence learning in a network of spiking neurons using maximum likelihood. *Tech. Report EDI-INF-RR-0149, School of Informatics, Edinburgh, UK* .

Barber D (2011) *Bayesian Reasoning and Machine Learning* Cambridge University Press.

Baum EB, Wilczek F (1987) Supervised learning of probability distributions by neural networks. *Neural Information Processing Systems* pp. 52–61.

Bi GQ, Poo MM (1998) Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience* 18:10464–72.

Bienenstock EL, Cooper LN, Munro PW (1982) Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *The Journal of Neuroscience* 2:32–48.

Brea J, Senn W, Pfister JP (2011) Sequence learning with hidden units in spiking neural networks. *Advances in Neural Information Processing Systems* 24:1422–1430.

Clopath C, Büsing L, Vasilaki E, Gerstner W (2010) Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nature Neuroscience* 13:344–352.

Dayan P, Hinton GE, Neal RM, Zemel RS (1995) The Helmholtz machine. *Neural Computation* 7:889–904.

Di Castro MA, Chuquet J, Liaudet N, Bhaukaurally K, Santello M, Bouvier D, Tiret P, Volterra A (2011) Local Ca2+ detection and modulation of synaptic release by astrocytes. *Nature Neuroscience* 14:1276–1284.

Diesmann M, Gewaltig MO, Aertsen A (1999) Stable propagation of synchronous spiking in cortical neural networks. *Nature* 402:529–33.

Düring A, Coolen A, Sherrington D (1998) Phase diagram and storage capacity of sequence processing neural networks. *Journal of Physics A: Mathematical and General* 31:8607–8621.

Elizondo DA (2006) The linear separability problem: some testing methods. *IEEE Transactions on Neural Networks* 17:330–344.

Gerstner W, Ritz R, van Hemmen JL (1993) Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biological cybernetics* 69:503–15.

Gerstner W, Kistler WM (2002) *Spiking neuron models: single neurons, populations, plasticity* Cambridge University Press.

Grossberg S (1969) On the serial learning of lists. *Mathematical Biosciences* 4:201–253.

Hahnloser RHR, Kozhevnikov AA, Fee MS (2002) An ultra-sparse code underlies the generation of neural sequences in a songbird. *Nature* 419:65–70.

Henneberger C, Papouin T, Oliet SHR, Rusakov DA (2010) Long-term potentiation depends on release of D-serine from astrocytes. *Nature* 463:232–236.

Hertz J, Krogh A, Palmer RG (1991) *Introduction to the theory of neural computation* Addison-Wesley Longman Publishing Co., Inc.

Herz A, Sulzer B, Kühn B, Van Hemmen JL (1988) The Hebb rule: Storing static and dynamic objects in an associative neural network. *Europhys. Lett.* 7:663–669.

Hinton G, Dayan P, To A, Neal R (1995) The Helmholtz machine through time. *Proceedings of the ICANN* pp. 483–490.

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79:2554–2558.

Horn D, Levy N, Meilijson I, Ruppin E (2000) Distributed synchrony of spiking neurons in a Hebbian cell assembly. *Advances in Neural Information Processing Systems* 12:129–135.

Lengyel M, Kwag J, Paulsen O, Dayan P (2005) Matching storage and recall: hippocampal spike timing–dependent plasticity and phase response curves. *Nature Neuroscience* 8:1677–1683.

Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. *Computer Science Review* 3:127–149.

Maass W, Sontag ED (2000) Neural systems as nonlinear filters. *Neural computation* 12:1743–72.

Maass W, Markram H (2002) Synapses as dynamic memory buffers. *Neural Networks* 15(2):155–161.

McCulloch WS, Pitts WH (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5:115–133.

Min R, Nevian T (2012) Astrocyte signaling controls spike timing-dependent depression at neocortical synapses. *Nature Neuroscience* 15:746–753.

Mishchenko Y, Paninski L (2011) Efficient methods for sampling spike trains in networks of coupled neurons. *The Annals of Applied Statistics* 5:1893–1919.

Panatier A, Theodosis DT, Mothet JP, Touquet B, Pollegioni L, Poulain Da, Oliet SHR (2006) Glia-derived D-serine controls NMDA receptor activity and synaptic memory. *Cell* 125:775–84.

Pfister JP, Gerstner W (2006) Triplets of spikes in a model of spike timing-dependent plasticity. *Journal of Neuroscience* 26:9673–9682.

Pfister JP, Toyoizumi T, Barber D, Gerstner W (2006) Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Computation* 18:1318–1348.

Pillow J, Latham P (2008) Neural characterization in partially observed populations of spiking neurons. *Advances in Neural Information Processing Systems* 20:1161–1168.

Ponulak F, Kasiński A (2010) Supervised learning in spiking neural networks with resume: Sequence learning, classification, and spike shifting. *Neural Computation* 22:467–510.

Rezende DJ, Wierstra D, Gerstner W (2011) Variational learning for recurrent spiking networks. *Advances in Neural Information Processing Systems* 24:136–144.

Rigotti M, Rubin DBD, Wang XJ, Fusi S (2010) Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Frontiers in computational neuroscience* 4:24.

Shu Y, Hasenstaub A, Duque A, Yu Y, Mccormick DA (2006) Modulation of intracortical synaptic potentials by presynaptic somatic membrane potential. *Nature* 441:761–765.

Sjöström PJ, Turrigiano GG, Nelson SB (2001) Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32:1149–64.

Sommer FT, Dayan P (1998) Bayesian retrieval in associative memories with storage errors. *IEEE Transactions on Neural Networks* 9:705–713.

Sompolinsky H, Kanter I (1986) Temporal association in asymmetric neural networks. *Physical Review Letters* 57:2861–64.

Sutskever I, Hinton GE, Taylor G (2009) The recurrent temporal restricted boltzmann machine. *Advances in Neural Information Processing Systems* 21:1601–1608.

Xu S, Jiang W, Dan Y, Poo Mm (2012) Activity recall in a visual cortical ensemble. *Nature Neuroscience* 15:449–455.

Fig. 1 Task and model description. **A** Left: Stimuli evoke activity patterns x (ticks) with probabilities given by a target distribution $P^*(x)$ (curves). Middle: In the absence of stimuli the network spontaneously generates activity patterns x distributed according to a model distribution $P_{w_0}(x)$ with synaptic strength parameters w_0. Right: With learning, the distribution of spontaneous activity patterns $P_w(x)$ approaches the target distribution. Hidden neurons (in gray) help to support the desired activity patterns in the visible neurons. **B** Neural dynamics define the model distribution $P_w(x)$ trough a spike reponse kernel ϵ, an adaptation kernel κ and the spike probability ρ. Minimizing the divergence $\mathcal{D}(P^*(x)\|P_w(x))$ from the target to the model distribution leads to a plasticity rule for the synaptic strengths w that is matched to the recall dynamics.

Fig. 2 Learnability of deterministic target sequences with a network of McCulloch-Pitts neurons. **A** Linearly separable sequence can be learned with only visible neurons. **B** The fraction of learnable sequences with only visible neurons for different relative sequence lengths and number of neurons: in red for a learning rule that finds an optimal solution if the sequence is linearly separable, in blue for the asymmetric Hebb rule discussed in Results ($N_\mathrm{v} = 40, 60, 80$, light to dark). The dots indicate the mean, the error bars the standard deviation of the posterior distribution with flat prior. **C** For a given number of visible neurons N_v and sequence length T, the number of possible deterministic target sequences is $2^{N_\mathrm{v}T}$. These sequences can be classified by either linear separability or Markovianity (see Materials and Methods). Linearly separable sequences can be learned with only visible neurons, non-linearly separable but Markovian sequences require sufficient hidden neurons and appropriate synaptic weights from visible to hidden and from hidden to visible neurons, and non-Markovian sequences require additionally connections between hidden neurons. **D** The simple asymmetric Hebb rule (blue curve; discussed in Results) does often not find weights that allow perfect recall if the pattern is temporally correlated. The fraction of learnable patterns (red curve) decreases only slowly with increasing correlation length α. **E** This target sequence is obviously not linearly separable and even non-Markovian: it contains an activity gap in the second time step (highlighted with red triangle) and in time steps 5 and 9 (highlighted with blue triangles) the same spatial pattern is followed by two different patterns in time steps 6 and 10. After training, a network with only visible neurons only recalls the gap faithfully when initialized with the first time step and only reproduces the states up to the ambiguous transition faithfully when initialized with the third time step just after the gap. Shown is in each case an overlay of 10 recalls. With the batch algorithm in Eq. 8 and the online plasticity rule in Eq. 12 the sequence can be learned. The states left of the green line were clamped for recall. With the online rule, the system learned to periodically produce the sequence and no further clamping of initial states was required during recall.

Fig. 3 Stochastic targets learned with McCulloch-Pitts dynamics. In all the simulations the target distribution was parametrized by target weights w^*. **A** Left: for a Markovian target with only visible neurons ($k = 1$) the weights can be learned with a model with only visible neurons or a model with visible (1-5) and hidden (6-20) neurons (upper-left submatrix: visible to visible connections). Middle: empirical frequency of subsequent activity patterns $v(t)$ and $v(t + 1)$. For dots on the black line the model frequency matches the target frequency. (black: only visible, red: with hidden) Right: the learning curve is similar for both models. **B** A $k = 3$ target cannot

be accurately learned with only visible neurons but can be learned with a model that includes hidden neurons. The target was implemented with connectivities such that the first group of hidden neurons (6-10) receives strong excitatory input from the visible and the second group of hidden neurons (11-15) receives strong excitatory input from the first group of hidden neurons. The learned hidden weights do not need to be the same as the target weights. **C** With increasing complexity ($k = 1$ to $k = 4$) a model with only visible neurons (black) or static hidden weights (blue) is not sufficient to accurately learn the target. The static hidden weights were obtained by reshuffling the weights learned in a simulation with plastic hidden weights. A model with 15 plastic hidden neurons (red) performs well up to the capacity limit $k = 4$. Learning stopped after 10^7 target samples, even when learning did not converge. Dots indicate the median, error bars the first and third quartile of the results from 16 simulations with different targets.

Fig. 4 Target distributions learned with alpha-shaped response kernels. **A** The target distribution $P^*(x) = P_{w^*}(x)$ has the same parametrization as the model with excitatory weights from groups of ten neurons projecting to subsequent groups of ten neurons and inhibitory weights otherwise. **B** Precisely timed spike patterns can be learned. **C** The sequence of Fig. 2A is encoded with periods of high or low firing rate. The target was generated by applying strong input current square pulses. Refractoriness sets an upper bound to the maximal firing rate. In recalls the model system was clamped to the target up to the green line.

Fig. 5 The learning rule shares similarities with Spike-Timing Dependent Plasticity (STDP). **A** The STDP window for different adaptation kernels (on the right in black, blue and red): the synaptic weight change Δw of a positive synaptic strength $w > 0$ after a pairing protocol on an interval of 200 ms, in which both neurons spike once with a separating interval of $\Delta t = t_{post} - t_{pre}$. Note that the shape of the curve in the causal part $\Delta t \geq 0$ is determined by the response kernel-shape ϵ and in the acausal part $\Delta t < 0$ depends on the adaptation kernel κ. **B** Experimental (black) and model (red) weight change induced by the pairing protocol described in Sjöström et al. (2001). Every 10 seconds, bursts of 5 pairs of pre- and postsynaptic spike with a relative timing of $\Delta t = t_{post} - t_{pre} = 10$ ms (solid lines) and $\Delta t = -10$ ms (dashed line) are induced at different pairing repetition frequencies. 15 such bursts are elicited (which give a total of 75 pairs) for all repetition except at 0.1 Hz. At this frequency, only 50 pairs are elicited (see Experimental Procedures in Sjöström et al. (2001)). It is assumed that at the beginning of the simulation, the weight is set to 1 and updated only after the induction protocol in order to mimic the time lag between induction and expression in the experiments.

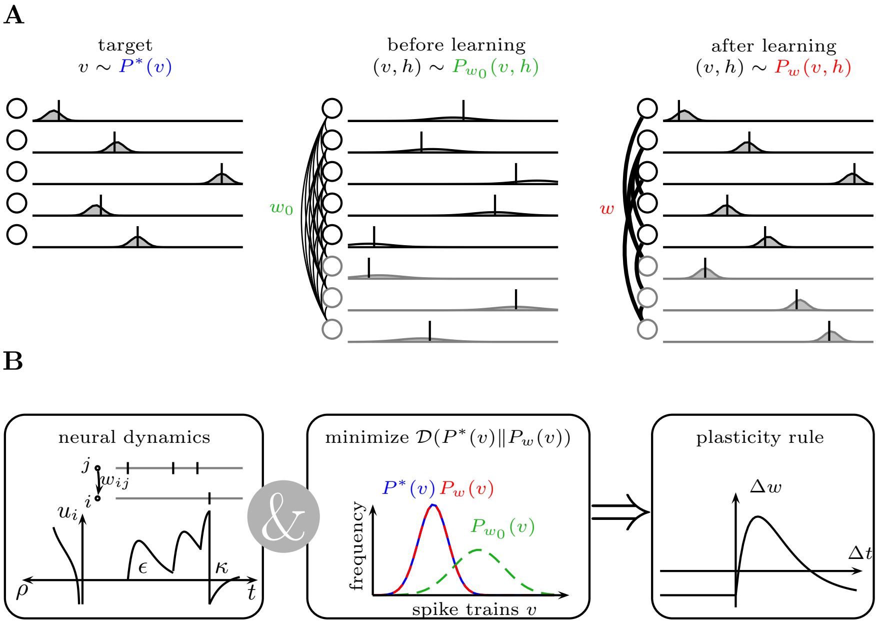
A
target
$v \sim P^*(v)$
before learning
$(v,h) \sim P_{w_0}(v,h)$
w_0
after learning
$(v,h) \sim P_w(v,h)$
w
B
neural dynamics
j
w_{ij}
i
u_i
ρ
ϵ
κ
t
&
minimize $\mathcal{D}(P^*(v) \| P_w(v))$
$P^*(v)$ $P_w(v)$
$P_{w_0}(v)$
frequency
spike trains v
plasticity rule
Δw
Δt

Figure 1

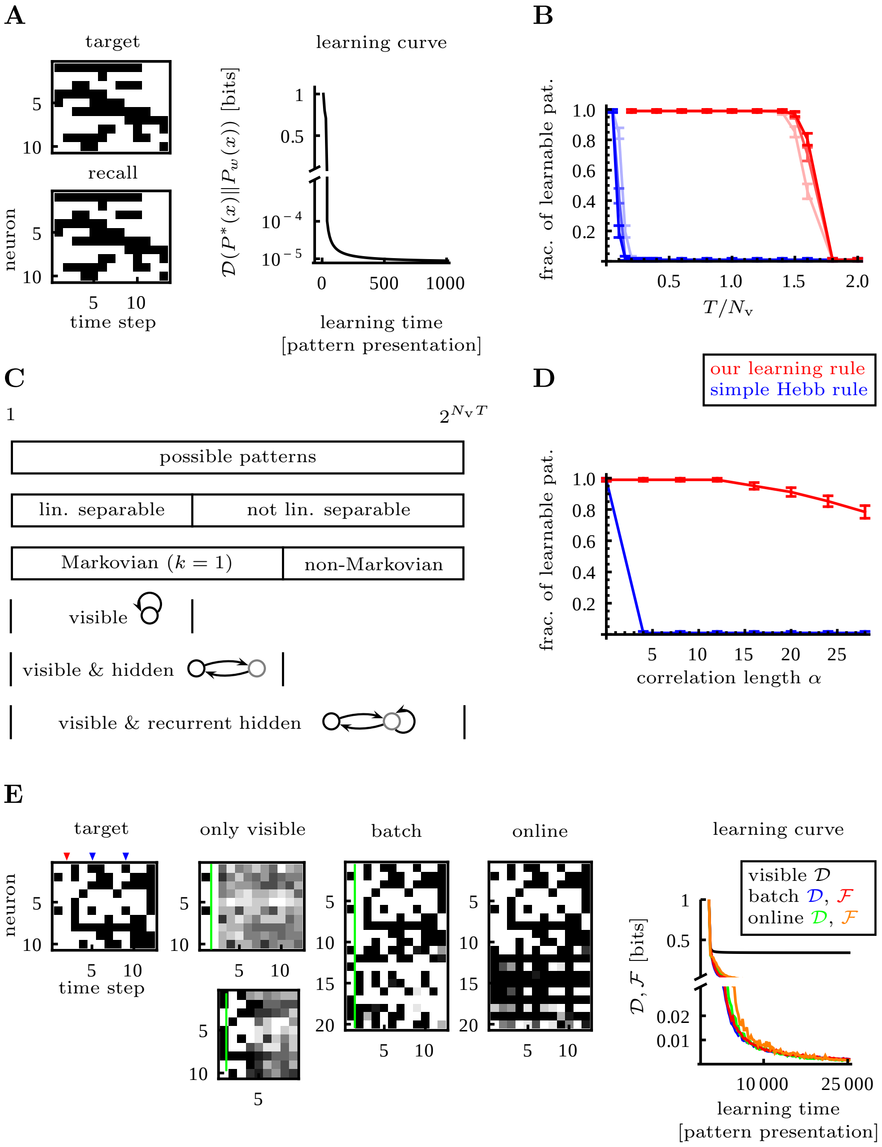
A
target
5
10
recall
neuron
5
10
5 10
time step
learning curve
$\mathcal{D}(P^*(x)||P_w(x))$ [bits]
1
0.5
10^{-4}
10^{-5}
0 500 1000
learning time
[pattern presentation]
B
frac. of learnable pat.
1.0
0.8
0.6
0.4
0.2
0.5 1.0 1.5 2.0
T/N_v
C
1
$2^{N_\mathrm{v}T}$
possible patterns
lin. separable
not lin. separable
Markovian ($k = 1$)
non-Markovian
visible
visible & hidden
visible & recurrent hidden
D
our learning rule
simple Hebb rule
frac. of learnable pat.
1.0
0.8
0.6
0.4
0.2
5 10 15 20 25
correlation length α
E
target
neuron
5
10
5 10
time step
only visible
5
10
5 10
5
10
5
batch
5
10
15
20
5 10
online
5
10
15
20
5 10
learning curve
visible $\mathcal{D}$
batch $\mathcal{D}$, $\mathcal{F}$
online $\mathcal{D}$, $\mathcal{F}$
$\mathcal{D}, \mathcal{F}$ [bits]
1
0.5
0.02
0.01
10 000 25 000
learning time
[pattern presentation]

Figure 2

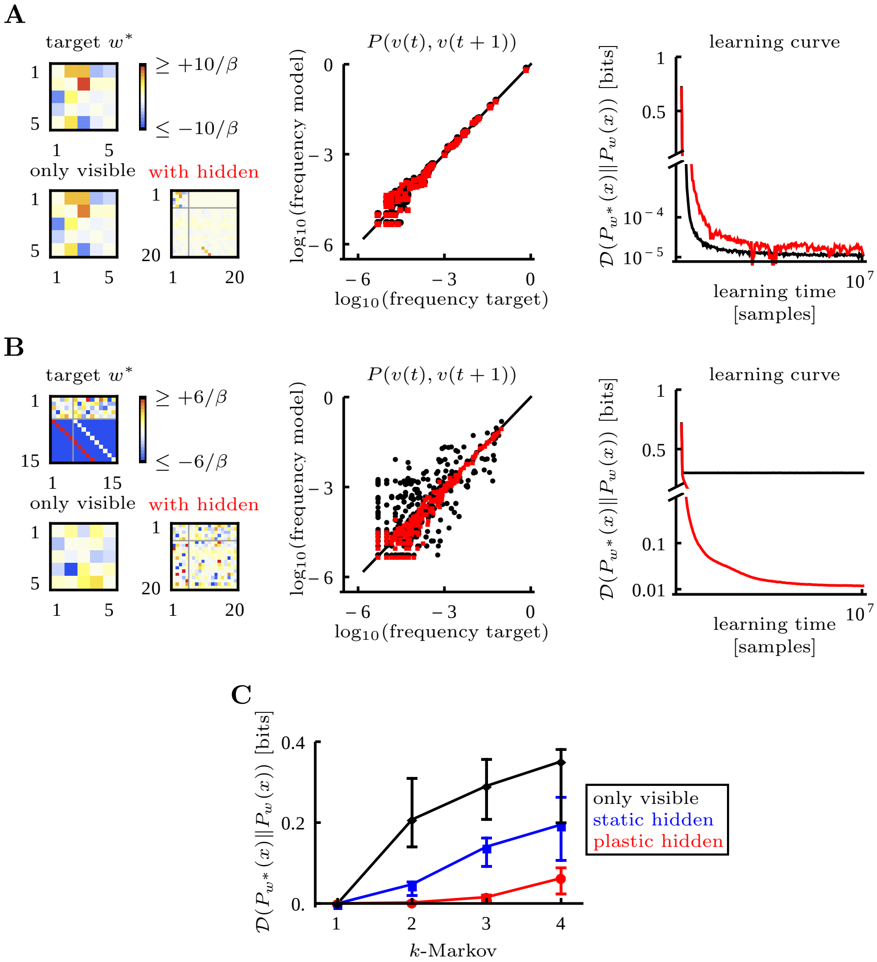
A
target w^*
1
5
1
5
$\geq +10/\beta$
$\leq -10/\beta$
only visible
with hidden
1
5
1
5
1
20
1
20
$P(v(t), v(t+1))$
$\log_{10}$(frequency model)
0
−3
−6
−6
−3
0
$\log_{10}$(frequency target)
learning curve
$\mathcal{D}(P_{w^*}(x) \| P_w(x))$ [bits]
1
0.5
10^{-4}
10^{-5}
learning time [samples]
10^7
B
target w^*
1
15
1
15
$\geq +6/\beta$
$\leq -6/\beta$
only visible
with hidden
1
5
1
5
1
20
1
20
$P(v(t), v(t+1))$
$\log_{10}$(frequency model)
0
−3
−6
−6
−3
0
$\log_{10}$(frequency target)
learning curve
$\mathcal{D}(P_{w^*}(x) \| P_w(x))$ [bits]
1
0.5
0.1
0.01
learning time [samples]
10^7
C
$\mathcal{D}(P_{w^*}(x) \| P_w(x))$ [bits]
0.4
0.2
0.
1
2
3
4
k-Markov
only visible
static hidden
plastic hidden

Figure 3

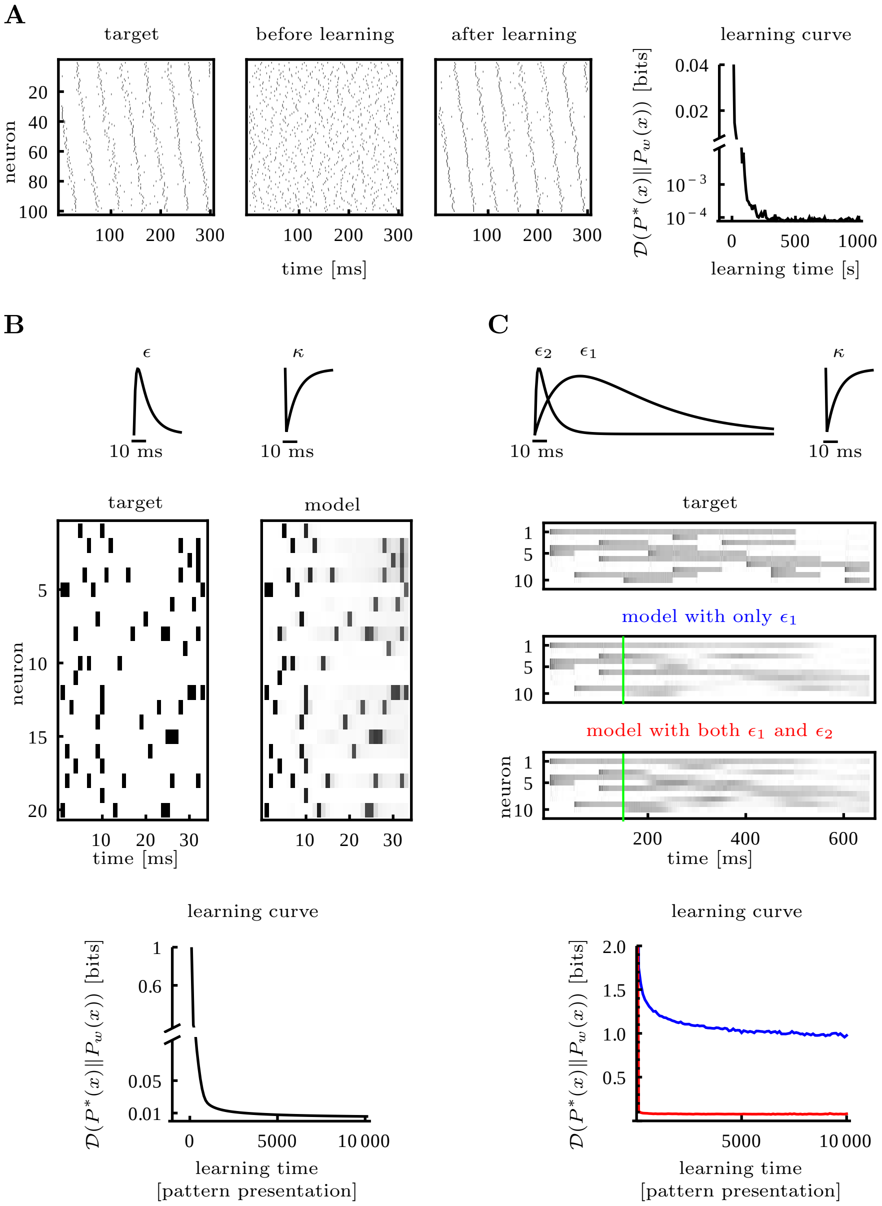
A
target
before learning
after learning
learning curve
neuron
20
40
60
80
100
100
200
300
time [ms]
$\mathcal{D}(P^*(x)\|P_w(x))$ [bits]
0.04
0.02
10^{-3}
10^{-4}
0
500
1000
learning time [s]
B
ϵ
κ
10 ms
target
model
neuron
5
10
15
20
10
20
30
time [ms]
learning curve
1
0.6
0.05
0.01
0
5000
10 000
learning time
[pattern presentation]
C
ϵ_2
ϵ_1
κ
10 ms
target
model with only ϵ_1
model with both ϵ_1 and ϵ_2
neuron
1
5
10
200
400
600
time [ms]
learning curve
2.0
1.5
1.0
0.5
5000
10 000
learning time
[pattern presentation]

Figure 4

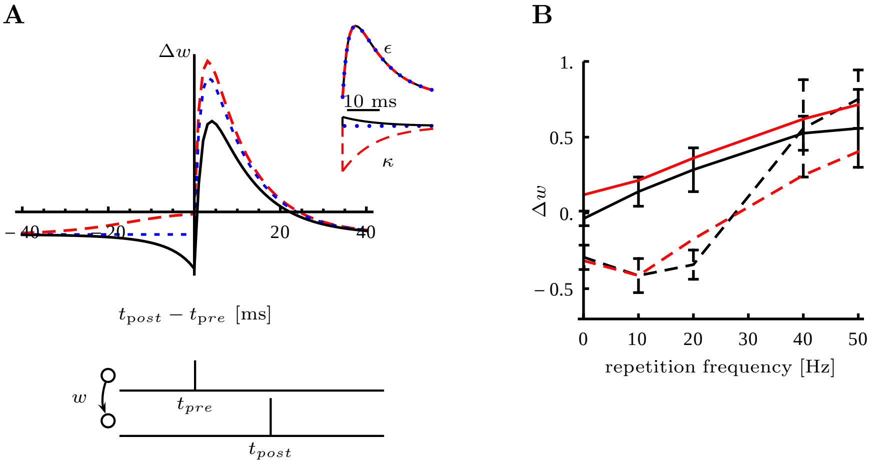
A
Δw
ϵ
10 ms
κ
− 40
20
20
40
$t_{post} - t_{pre}$ [ms]
w
t_{pre}
t_{post}
B
1.
0.5
0.
− 0.5
Δw
0
10
20
30
40
50
repetition frequency [Hz]

Figure 5