Neuron
Article

Learning Precisely Timed Spikes

Raoul-Martin Memmesheimer,[1,2,6] Ran Rubin,[3,4,6] Bence P. Ölveczky,[2,5] and Haim Sompolinsky[2,3,5,*]
[1]Donders Institute, Radboud University, Nijmegen 6525, the Netherlands
[2]Center for Brain Science, Harvard University, Cambridge, MA 02138, USA
[3]Racah Institute of Physics
[4]The Edmond and Lily Safra Center for Brain Sciences
Hebrew University, Jerusalem 91904, Israel
[5]Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA 02138, USA
[6]Co-first author
*Correspondence: haim@fiz.huji.ac.il
http://dx.doi.org/10.1016/j.neuron.2014.03.026

SUMMARY

To signal the onset of salient sensory features or execute well-timed motor sequences, neuronal circuits must transform streams of incoming spike trains into precisely timed firing. To address the efficiency and fidelity with which neurons can perform such computations, we developed a theory to characterize the capacity of feedforward networks to generate desired spike sequences. We find the maximum number of desired output spikes a neuron can implement to be 0.1–0.3 per synapse. We further present a biologically plausible learning rule that allows feedforward and recurrent networks to learn multiple mappings between inputs and desired spike sequences. We apply this framework to reconstruct synaptic weights from spiking activity and study the precision with which the temporal structure of ongoing behavior can be inferred from the spiking of premotor neurons. This work provides a powerful approach for characterizing the computational and learning capacities of single neurons and neuronal circuits.

INTRODUCTION

Throughout the CNS, neuronal communication is largely carried out by the propagation of action potentials or spikes. The fundamental computation of single neurons is the transformation of incoming spike trains into appropriate spike output. For many biologically relevant tasks, temporal precision in the neuronal responses is essential, for instance, when neurons signal the onset times of salient features in sensory stimuli or when circuits control precisely timed sequences of movements. Thus, it is important to understand the extent to which neurons can map input spike patterns to output spike trains and the constraints imposed by synaptic connectivity, the neurons' electrical integration properties, and their spike generation mechanism.

Here we characterize the capacity of spiking neurons to implement desired transformations between input and output spike patterns. We evaluate the maximum number and length of mappings that can be implemented by a neuron and describe its dependence on neuronal time constants and input and output firing statistics.

Perceptual and motor skills requiring precise timing are often acquired through learning, suggesting that experience-dependent synaptic plasticity mechanisms can train neuronal circuits to learn new associations between pairs of input spike patterns and desired output spike sequences.

Here we present a simple, efficient, and biologically plausible neuronal learning algorithm capable of training neurons to generate desired spike trains with a specified temporal tolerance, in response to their associated inputs. We demonstrate the utility of this learning rule by applying it to three challenging problems. First, we show that it can be used as a data analysis tool for reconstructing synaptic connections from observed spike patterns. Second, we use it to study the temporal information about an ongoing vocal behavior embedded in the spiking patterns of the songbird motor cortex and to model the decoding of this information by downstream neurons. Third, we show that it allows learning of multiple stable, precisely timed patterns of spikes in networks with recurrent topology.

RESULTS

A well-known simplified neural model that performs input-output transformations is the Perceptron (Rosenblatt, 1962; Minsky and Papert, 1988), according to which at each time bin a neuron performs a weighted linear sum of its incoming spikes and generates an output spike if the net synaptic potential is above threshold. Indeed, the theory of the Perceptron and its celebrated learning algorithm have been invoked in the study of various neuronal systems, including sensorimotor learning in the cerebellum (Marr, 1969; Albus, 1971; Brunel et al., 2004; Clopath et al., 2012) and associative memory in cortex (Gardner, 1988; Chapeton et al., 2012). As will be shown below, properties derived using this static linear-threshold model are problematic, because this model ignores fundamental features of neuronal dynamics, such as the integration of incoming spikes over time, the absence of a natural discretization of the signals into time bins, and the membrane potential reset after an output spike.

A neuron model that incorporates these features and maintains some analytical tractability is the Leaky Integrate-and-Fire (LIF) neuron. It consists of linear spatiotemporal

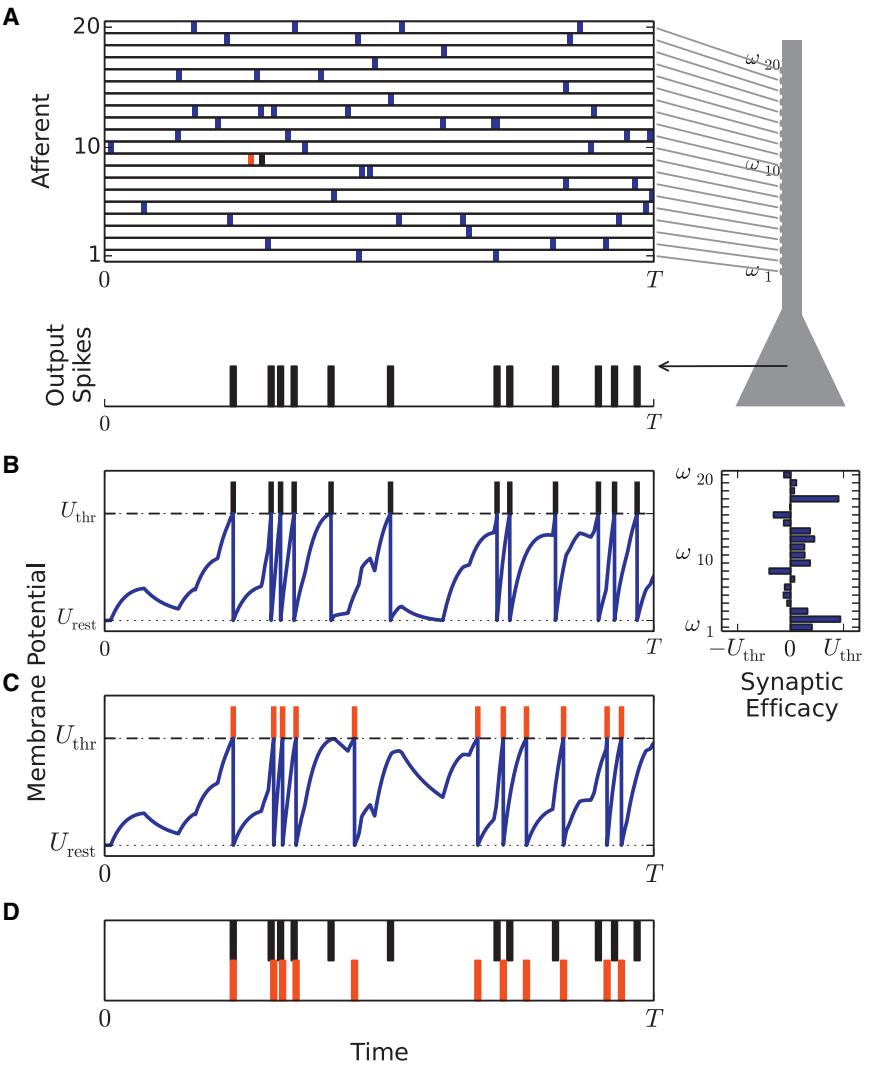

Figure 1. Neuronal Architecture

(A) An example input pattern (top) and output spikes (bottom). Each row represents the activity of one input afferent as a function of time. Vertical bars depict the timing of the spikes. The bottom time axis depicts the neuron's output spikes in the response to the input pattern.

(B) Voltage trace (left) and synaptic weights (right) of an LIF neuron integrating the input spikes in (A) according to Equation 2. Output spike times are marked by vertical bars.

(C) Sensitivity of voltage to spike reset times. The same neuron is driven by the same input spikes as in (B) except for the spike of input afferent 9, which is shifted 20 ms backward in time (black and red spikes in A). The shifted input induces at first only a small difference in subsequent spike times. However, after an omitted output spike, the missing reset results in substantial perturbations of the subsequent spike sequence.

(D) Output spike times in (B) (black) compared to output spike times in (C) (red).

summation of incoming spikes and output spike generation by threshold crossing followed by membrane potential reset (Gerstner and Kistler, 2002; Dayan and Abbott, 2005). To date, the capacity of LIF neurons to learn associations of input-output spike trains is not known, nor is there an algorithm for learning such associations with guaranteed convergence. Here we fill this gap by introducing theories and learning algorithms that allow the exploration of spike sequence generation and learning in feedforward as well as recurrent networks of LIF neurons.

Neuronal Architecture

We consider an LIF neuron with N spiking input afferents (Figure 1A). Each afferent, i, emits spikes at a set of times, $\{t_i\}$, and contributes to the total membrane potential a postsynaptic potential (PSP), $\omega_i x_i(t)$, where ω_i is the afferent's synaptic efficacy and $x_i(t)$ is given by

$$x_i(t) = \sum_{t_i<t} u(t - t_i) \,, \qquad \text{(Equation 1)}$$

with $u(t) = U_0(e^{-(t/\tau_m)} - e^{-(t/\tau_s)})$; U_0 is chosen such that the maximal value of $u(t)$ is 1, and τ_m and τ_s denote the membrane and synaptic time constants, respectively. Thus, the total PSP is $\boldsymbol{\omega}^T\mathbf{x}(t) = \sum_{i=1}^{N} \omega_i x_i(t)$. Spikes are generated when the potential crosses threshold (Figure 1B) as described below.

Implementing Precise Spike Time Input-Output Associations

How can we characterize the ability of a neuron to transform a given set of input spike trains, $\{t_i\}$, to a sequence of "desired" output spikes, $\{t_d\}$ (Figure 2A)? In the simple Perceptron approximation, the neuron performs a linear threshold operation on inputs that arrive in discrete time bins. In each time bin t, the total PSP obeys $\boldsymbol{\omega}^T\mathbf{x} > U_{thr}$ if there is an output spike in that bin and $\boldsymbol{\omega}^T\mathbf{x} < U_{thr}$ otherwise. This allows for application of theoretical results and learning algorithms appropriate for linear threshold devices (Bressloff and Taylor, 1992; Brunel et al., 2004; Clopath et al. 2012).

There are several major problems with this approach. First, the Perceptron approximation incorporates the linear dependence of the subthreshold potential on the synaptic weights but ignores the highly nonlinear dependence of the suprathreshold potential. The nonlinearity is a consequence of the fact that in a real neuron output spikes cause the neuron's membrane potential to undergo reset. Indeed, small variations in the time of one output spike can cause subsequent spikes to vanish or spurious spikes to appear. This sensitivity can drastically affect spike generation at later times (Figures 1C and 1D). Second, a set of weights that satisfy all constraints on a discrete set of times does not necessarily prevent errors at other times, as the neuron's integration is inherently continuous in time. This

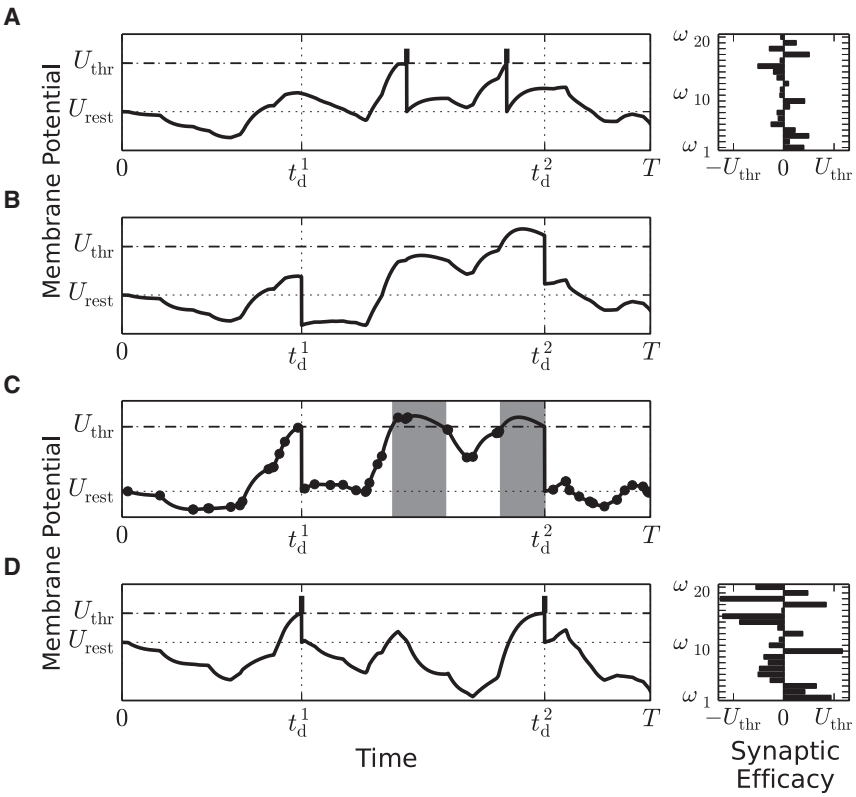

Figure 2. HTP Method

(A) Voltage trace (left) and synaptic weights (right) of an LIF neuron integrating the input spikes in Figure 1A according to Equation 2. Output spike times are marked by vertical bars. Desired spike times are denoted as t_d^1 and t_d^2.

(B) High-threshold dynamics. During the weight update process, the neuron does not spike at threshold crossing and membrane potential resets are forced at the desired times.

(C) High-threshold dynamics after the projection of the synaptic weights (see Results and Experimental Procedures). The membrane potential is at the threshold exactly at the desired times. Circles depict the timing of input spikes. Times at which the membrane potential is above threshold (error) are depicted in gray.

(D) Voltage trace (left) and synaptic weights (right) after successful learning. Spikes are emitted at the desired times and the membrane potential is below threshold at all other times.

problem is aggravated by the above-mentioned inherent sensitivity to the precise spike times, due to spike reset. Third, a naive application of Perceptron theory would predict that the capacity of a neuron to generate spike sequences is inversely proportional to the number of time bins, which implies a vanishing capacity in the continuous time limit (Bressloff and Taylor, 1992; Clopath et al., 2012). As we will see below, the problems of spike reset and continuous time also invalidate the simple Perceptron learning rule as a model of learning in spiking networks.

To incorporate the effect of spike reset, the suprathreshold potential, relative to the neuron's resting potential, can be written as

$$U(t) = \boldsymbol{\omega}^T \mathbf{x}(t) - U_{thr} x_{reset}(t). \qquad \text{(Equation 2)}$$

The second term in Equation 2 is the contribution due to the potential reset, given by

$$x_{reset}(t) = \sum_{t_{spike}<t} u_r(t - t_{spike}), \qquad \text{(Equation 3)}$$

where $\{t_{spike}\}$ are the output spike times, and $u_r(t) = e^{-(t/\tau_m)}$ models the effect of the potential reset. A spike and postspike reset are generated whenever $U(t) = U_{thr}$ and the second term in Equation 2 enforces $U(t) = 0$ immediately after an output spike. We further demand that at the time of spiking dU/dt be positive, as action potentials are suppressed when depolarization of the voltage is too slow.

We now formalize the demands on the neuron as follows: the synaptic weight vector $\boldsymbol{\omega}$ that implements a mapping between a set of precisely timed input spikes and desired output spikes at times $\{t_d\}$ must obey the following conditions: (1) $U(t_d) = U_{thr}$, (2) $dU/dt(t_d) > 0$ for all t_d, and (3) $U(t) < U_{thr}$ at all times except t_d. Utilizing the fact that for a given set of output spike times we can substitute t_d for t_{spike} in Equation 3, the potential $U(t)$ (Equation 2) is rendered to be linearly related to $\boldsymbol{\omega}$ and to the PSP input vectors $\mathbf{x}(t)$. Hence, the constraints (1)–(3) can be viewed as a combination of linear equalities (1) and linear inequalities (2 and 3). Since time is continuous, the neuron must obey an infinite number of inequalities (of type 3). Nevertheless, as will be shown below, due to the strong temporal correlations in the inputs, the effective number of inequalities is finite, and the neuron possesses considerable capacity of generating precisely timed spike sequences. The above description provides a geometric characterization of the "decision surface" of a spiking neuron as implementing linear equalities and inequalities in the PSP input vector space.

How many associations between input and output spike trains can a neuron implement? Since an exact analytical evaluation of this capacity is difficult, we estimated the capacity using extensive computer simulations. To do this, we used the above geometric insight to develop an efficient algorithm denoted as High-Threshold Projection (HTP) method for finding a set of synaptic weights that implement given maps of input-output spike trains in LIF neurons. First, to turn off the nonlinear effect of threshold crossings at the wrong times, we increase the firing threshold, essentially preventing the neuron from spiking except at desired times when potential resets are enforced (Figure 2B). Second, a projection operation is used at each trial to ensure that the weight vector $\boldsymbol{\omega}$ implements the equality constraints (1) throughout the weight update process (Figure 2C; Experimental Procedures). Lastly, the remaining inequality constraints are solved by a Perceptron-like error-based update rule (Experimental Procedures). Importantly, we have proved that the HTP

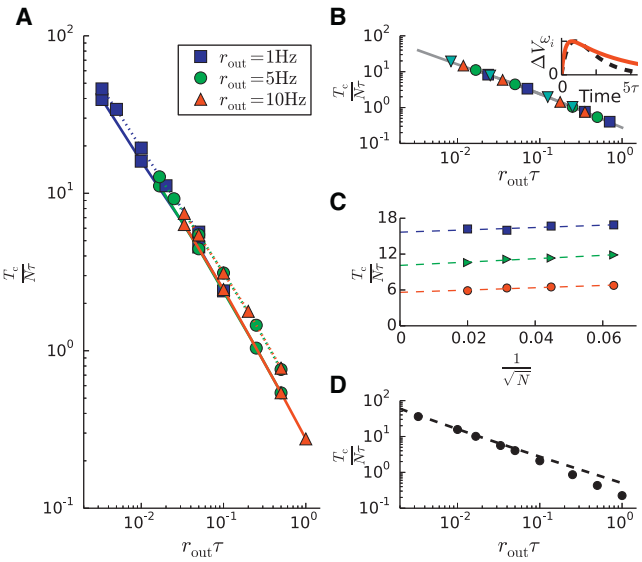

Figure 3. Capacity of the LIF Neuron

(A) Symbols connected by solid lines depict the capacity of LIF neurons for random input-output streams of precisely timed spikes for several values of the mean output rate r_{out} and the neuron's time constant $\tau = \sqrt{\tau_m \tau_s}$ with a fixed input rate r_{in} = 5 Hz. Different points with the same symbol correspond to different values of τ but a single value of r_{out} (legend). Capacity is measured by the total length of a single learnable spike sequence in units of τ, per synapse. The results indicate that the capacity exhibits scaling properties: it is a function only of the mean number of output spikes in time τ, i.e., $r_{out}\tau$. Symbols connected by dotted lines depict the capacity of the FP learning algorithm with $r_{out}\varepsilon = 10^{-2}$ (see Results).

(B) Independence of the capacity of τ_m/τ_s. The LIF neuron's capacity is plotted versus $r_{out}\tau$ for different values of the ratio τ_m/τ_s (inverted triangles, triangles, circles, and squares correspond to τ_m/τ_s = 16, 8, 4 and 2, respectively). r_{out} = 5 Hz and r_{in} = 5 Hz are kept constant. Gray lines in the background show the LIF capacity from (A) for comparison. The inset depicts the postsynaptic potential kernel $u(t)$ (see Equation 2) with τ_m/τ_s = 2 (dashed line) and τ_m/τ_s = 16 (solid line).

(C) Capacity in the large N limit. Capacity exhibits a finite size correction, linear in $1/\sqrt{N}$ (top to bottom: $r_{out}\tau$ = 0.01, 0.017, 0.033).

(D) Circles depict the capacity for large N extrapolated from the results in (C). Dashed line depicts the theoretical prediction derived from the mapping to an effective discrete time Perceptron (Results). See also Figure S1.

algorithm converges after a finite number of updates to a set of weights that implements exactly the desired input-output spike time mapping if such a solution exists (Figure 2D; Experimental Procedures). This allows us to explore the capabilities of the LIF neuron to perform such tasks and their dependence on the number of synapses and other parameters of the system.

Capacity of LIF Neurons

To assess the capacity of a neuron to store input-output associations, we consider the commonly used benchmark of random inputs and outputs (Gardner, 1988; Monasson and O'Kane, 1994; Rubin et al., 2010). Here, the spike times for each input neuron and the desired output spike times are randomly chosen with mean rates r_{in} and r_{out}, respectively (Experimental Procedures). We define the capacity as the maximal combined duration, T, of the sequences that can be learned. For simplicity, we consider a single long input-output sequence. Importantly, we find that for large systems, the capacity is proportional to the number of input synapses (Figure 3C). A key parameter is the PSP correlation time, which we define as $\tau = \sqrt{\tau_m \tau_s}$ (Rubin et al., 2010). We find that the capacity per synapse, expressed in units of τ, is a function of a single parameter, the mean number of output spikes within time τ, $r_{out}\tau$. Thus,

$$\frac{T}{N\tau} = \alpha_c(r_{out}\tau). \qquad \text{(Equation 4)}$$

In particular, α_c increases with decreasing $r_{out}\tau$ (Figure 3). This is in contrast to previous studies (Bressloff and Taylor, 1992; Clopath et al., 2012) that suggested that the temporal input correlation does not necessarily affect the capacity of the spiking neuron. Although the analytic calculation of α_c for our problem is difficult, it can be estimated for low output firing rates using an analogy to the capacity of the Perceptron. First, we discretize time into bins of length τ and consider different bins as uncorrelated. Second, we note that the effective dimensionality of our problem is $N_{eff} = N - n_{spikes} \approx N - r_{out}T$ due to equalities (1) above. Similarly, the effective number of patterns with -1 labels is T/τ (inequalities [3] above) and the number of positive labels is $n_{spikes} \approx r_{out}T$ (inequalities [2]). We can now estimate the capacity of the LIF neuron as the capacity of the Perceptron with the same dimensionality and label statistics as above (Supplemental Experimental Procedures). This estimate yields a very good agreement with the simulation results, especially for $r_{out}\tau \lesssim 0.1$ (Figure 3D). For $r_{out}\tau \ll 1$, the capacity is predicted to diverge as $\alpha_c \approx (2\, r_{out}\tau\, |\log r_{out}\tau|)^{-1}$. Importantly, for $r_{out}\tau \gtrsim 1$, this approximation breaks down because the interaction between neighboring output spikes due to reset cannot be neglected.

As Equation 4 and Figures 3A and 3B imply, for large systems the capacity is invariant to changes in the input rate variable $r_{in}\tau$, as well as the PSP shape parameter τ_m/τ_s. Additionally, we find that the capacity of learning a single long pattern with duration T is very similar to that of learning multiple patterns of short durations of the same combined duration T (Figure S1A).

The capacity results stated above are valid in the regime where, although each input afferent may fire at low rates, the total number of incoming spikes within time τ is large. This regime is relevant for neurons in cortex and many subcortical structures, which receive inputs from thousands of synaptic afferents. We observe a decrease in capacity when the total number of inputs within time τ is $O(1)$ (data not shown).

So far, we have characterized the neuron's capacity in terms of the maximum duration of spike sequences that it can implement. Alternatively, we can ask how many desired output spikes, n_{spikes}, a neuron can implement. This quantity can be readily computed by multiplying Equation 4 by the output spike rate parameter, $r_{out}\tau$ (Figure S1C). As a concrete example, with τ = 10 ms, the mean sequence duration that can be implemented ranges from 50 ms per synapse for output firing rates (r_{out}) of 0.5 Hz to 2 ms per synapse for r_{out} = 100 Hz. This is equivalent to a maximum mean number of precise output spikes varying from 0.14 to 0.28 spikes per synapse for output firing rates of 0.5 Hz to 100 Hz, respectively.

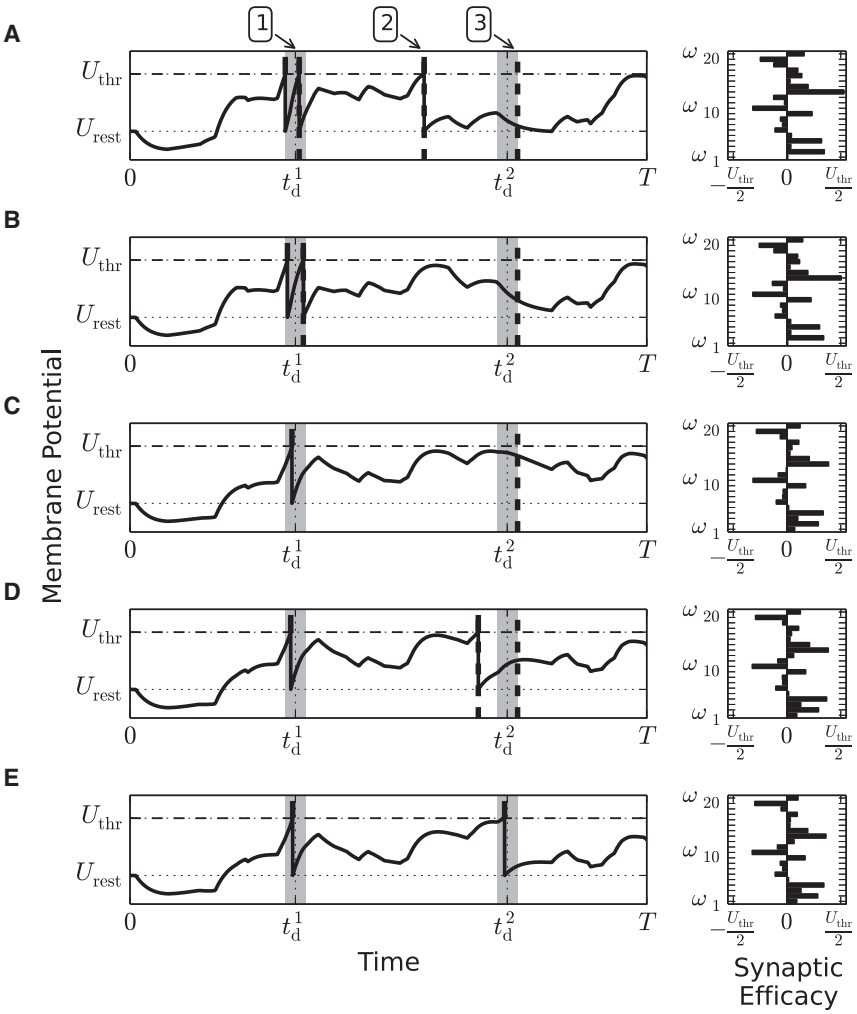

Figure 4. Finite Precision Learning

(A–D) Examples of voltage traces (left) along with the synaptic weights (right) calculated for the inputs of Figure 1A, at different stages of learning. Output spikes are denoted by black bars, tolerance windows are depicted by gray areas, and errors are marked by dashed lines. In (A), there are three errors: undesired spikes both inside (1) and outside (2) the tolerance windows and a missed spike at the second desired time (3). Learning is always performed with respect to the first error only.

(E) After successful learning, the neuron emits one output spike within each tolerance window.

Learning Spike Sequences with Finite Precision

The idealistic task of learning output spike sequences with perfect precision allows for theoretical insights into the parameters affecting the capabilities of a spiking neuron. The above theory relied on forcing the reset times to the desired output spike times, transforming the problem to that of fulfilling linear constraints imposed on the synaptic weights. However, incorporating such forced resets in a plausible synaptic learning rule is problematic since it would require a mechanism that overrules the natural spiking dynamics of the neuron during learning. Furthermore, realistic tasks should include finite temporal precision in the output spikes. Since, in this situation, reset times are not specified precisely in advance, there is no way of forcing them to be at fixed desired times during learning. Thus, a viable model of neuronal learning must incorporate a desired degree of precision and overcome the nonlinear effects of the potential reset.

We have developed a learning algorithm for Finite Precision (FP) tasks. The neuron is required to generate output spikes at specified times up to a tolerance window of width ε (Figure 4A). During learning, the output neuron generates spikes according to the LIF dynamics (Equations 2 and 3) in response to the given input spikes, without forcing resets at the desired spike times. To avoid nonlinear accumulation of errors due to interaction between output spikes, synaptic modification is initiated only by the first error in each trial. If this error consists of an output spike outside the tolerance window or more than one spike inside it, weights are depressed according to $\Delta\omega_i \propto -x_i(t_{\mathrm{err}})$ with t_{err} being the time of the extra spike. Conversely, if the first error consists of a failure to spike within a tolerance window, weights are potentiated according to $\Delta\omega_i \propto x_i(t_{\mathrm{err}})$ with t_{err} being the end of the tolerance window (Figures 4B–4E). Learning from the first error ensures that synaptic modification does not have to take into account the nonlinear accumulations of erroneous reset times since, by construction, all output spikes prior to t_{err} are in the correct time within the tolerance window.

Since desired output spike times are not specified exactly, the task cannot be mapped into a set of linear constraints. Thus, the solution space is not necessarily convex (Rubin et al., 2010), and convergence proofs derived for linear problems are not applicable. However, we have proven that the FP algorithm converges to a solution in finite time, if there is a weight vector that performs the task, and, in addition, is stable in the sense that at any time it performs as desired even if past output spike times are perturbed within the allowed windows. These extra robustness requirements are expected to hold in the regime where $\varepsilon/\tau \ll 1$ and output spikes are nonbursting (Supplemental Experimental Procedures). Furthermore, we observed that the FP algorithm converges to a solution even when this additional condition does not hold (data not shown).

We have used the FP algorithm to estimate the effect of finite ε on learning random input-output spike trains. Figure 3A shows that the capacity yielded by the algorithm is higher than the more stringent task of exact spike times, as expected. As $r_{\mathrm{out}}\varepsilon$ decreases, the algorithm's capacity approaches that of the exact time result (Figure S1B), yielding additional support for the algorithm's convergence. The simplicity of the FP learning algorithm and the biological plausibility of using the intrinsic

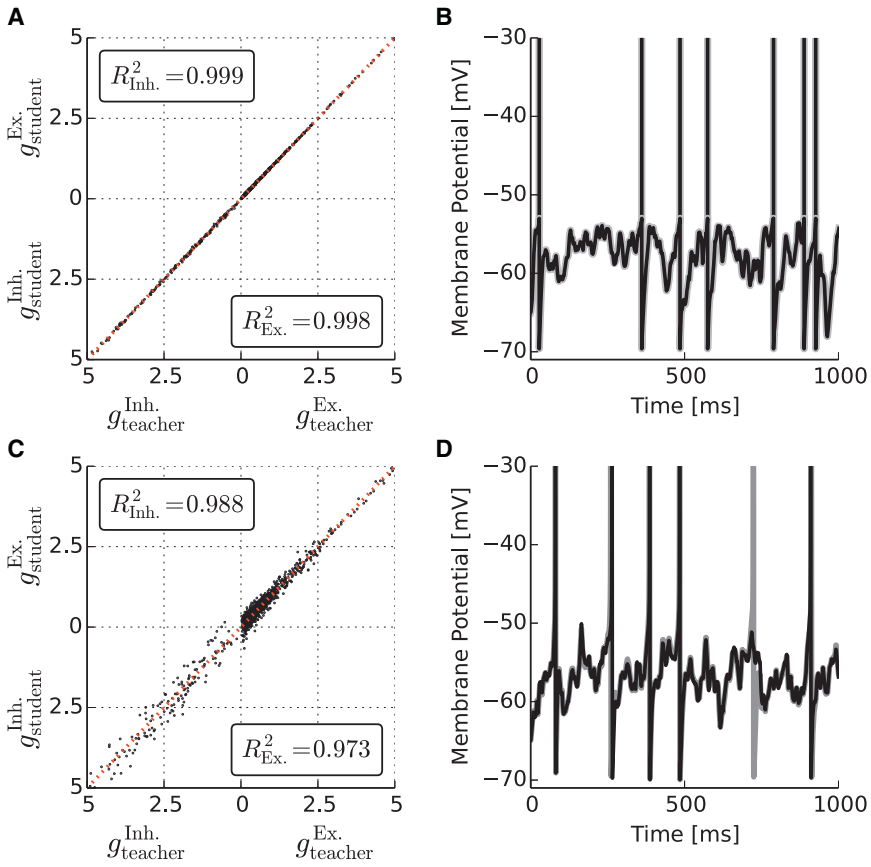

Figure 5. Synaptic Weights Reconstruction

(A and C) An example reconstruction of synaptic weights for an LIF neuron (A) and an HH neuron (C). Weights represent maximal synaptic conductances for excitatory synapses, $g^{\text{Ex.}}$, and inhibitory synapses, $g^{\text{Inh.}}$ (Experimental Procedures). Conductances are given in units of 1σ of the teacher's excitatory synapses' maximal conductances. Average R^2 values for excitatory and inhibitory synapses are given in legends.

(B) An example of voltage traces for the teacher (gray) and student (black) LIF neurons driven by the same input spike pattern.

(D) An example of voltage traces for the teacher HH neuron (gray) and student HH neuron (black) driven by the same input spike pattern (experimental procedures). See also Figure S2.

neuron's dynamics during learning make it an attractive candidate for learning tasks involving output spike sequences.

Reconstruction of Synaptic Weights

In recent years, new techniques, such as two-photon microscopy and multielectrode arrays, allow experimentalists to record the activity of multiple interconnected neurons in the brain simultaneously. Further technological developments promise to allow the recording of all neurons in a local circuit (Alivisatos et al., 2012). One goal of such dense activity mapping is to reverse engineer the underlying synaptic connectivity, complementing dense structural imaging (Lichtman and Denk, 2011).

Here we use the FP learning algorithm to reconstruct the efficacies of synaptic afferents of a model neuron, henceforth the "teacher," from measurements of its input-output spike trains. Input consists of multiple episodes of homogeneous Poisson-generated spike trains (Experimental Procedures). The synaptic weights are estimated by training a "student" LIF neuron to fire at the same times as the "teacher" neuron, using the FP algorithm with tolerance of a few milliseconds.

We first consider the simple case where both student and teacher obey the same LIF dynamics (Experimental Procedures). In this case, the learned weights converge to the weights of the teacher (Figure 5A) and the output spikes of the student neuron closely follow those of the teacher (Figure 5B). We then consider the more realistic setting of a teacher Hodgkin-Huxley (HH) neuron with conductance-based synapses (Experimental Procedures). In this case, the mapping between the biophysical parameters of the two models is unclear. Hence, the values of the LIF time constants τ_m and τ_s and the reset potential were chosen to minimize the distance between the true and reconstructed spike trains (Experimental Procedures). Similar optimization has been applied to determine an overall conductance scale as well as a driving force parameter, both of which are needed in the estimation of synaptic conductances from reconstructed current-based weights. Finally, optimizing the learning tolerance parameter yielded $\varepsilon = 7$ ms. Importantly, despite the model mismatch, our algorithm yields a very good estimate of the true synaptic weight vectors (Figure 5C; Experimental Procedures) and spike trains (Figure 5D; Experimental Procedures). Naturally, the quality of the reconstruction crucially depends on the amount of data, i.e., the total duration of input-output patterns (Figures S2A and S2B). As expected, we find that the quality of reconstruction increases with the total duration of input-output patterns used for training. For the reconstruction presented in Figure 5, we used 3 s of input-output pattern per synapse. However, reconstruction quality is reasonable ($R^2 \approx 0.9$) even for a total duration of 1 s per synapse.

The above results assume that the spiking activity of all of the neuron's presynaptic afferents is known. However, in most experimental settings the activity of some of the input afferents may not be observable. To test the effect of unobserved input afferents, we performed the reconstruction while only considering input spikes from a fraction of the teacher neuron's input afferents (Figures S2C and S2D; Supplemental Experimental Procedures). Importantly, we find that the FP algorithm achieves reasonable reconstruction quality ($R^2 \approx 0.9$) even when approximately 20% of the teacher neuron's input afferents are not used for training the student neuron.

Interestingly, reconstruction of the inhibitory synapses is less affected by short training sequences (Figures S2A and S2B) or by missing inputs (Figures S2C and S2D). This is due to the

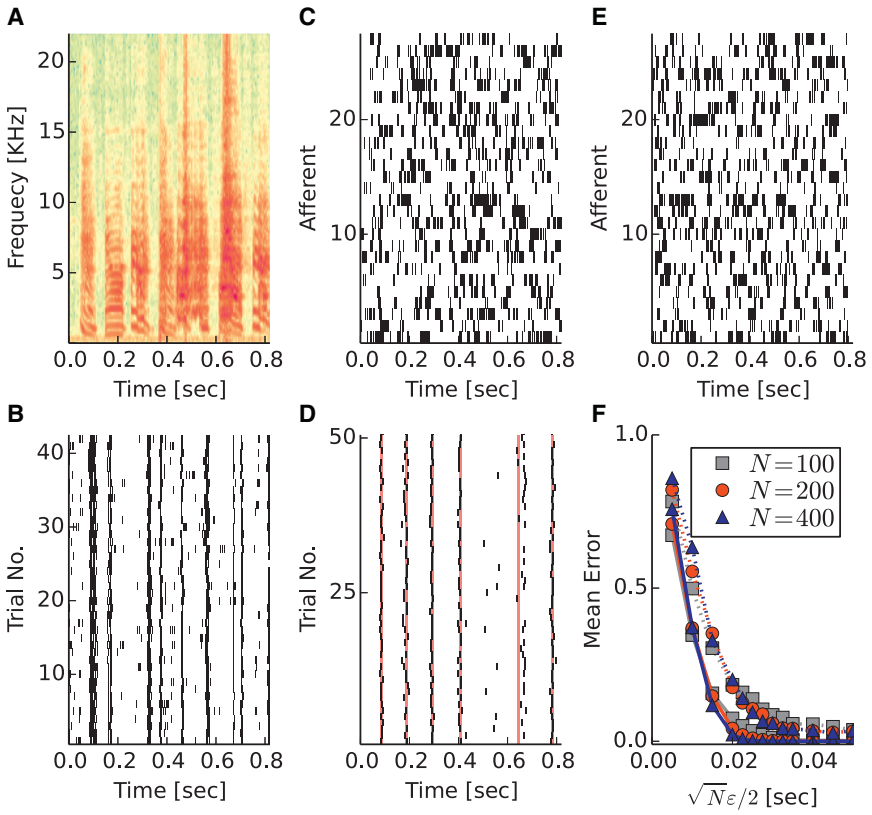

Figure 6. Reading the Neural Code in RA

(A) A spectrogram of a bird's repeating song motif.

(B) The spikes recorded from a single neuron in RA during multiple renditions of the song motif. Recorded spikes were aligned according to the onset times of the song's syllables (Experimental Procedures).

(C) An example input pattern. Each row depicts the spike train from a randomly chosen trial, of one of 27 recorded neurons from the first bird.

(D) Response of the readout neuron to previously unseen input patterns. The neuron was trained to fire six spikes near syllable onsets. Tolerance windows around the desired times are depicted in red (ε = 0.01 s). Most of the output spikes are in or near the tolerance windows. Similar performance was observed for input patterns constructed from the neurons of the second bird.

(E) An example input pattern from 27 synthetic neurons constructed from the recorded neurons (see Results). Pattern looks similar to that of the recorded neurons (C).

(F) Dependence of mean training (solid) and generalization (dashed) errors on the precision requirement, $\varepsilon/2$, and RA synthetic population size, N. Errors for different N collapse to the same curve when plotted against the scaled variable $\sqrt{N}\varepsilon/2$. Training and generalization errors are evaluated by averaging over a binary error variable (where error of 1 is attributed to any trial in which an error of any type occurred). See also Figure S3.

fact that inhibitory synapses in the present model are, in general, stronger than excitatory ones (Experimental Procedures).

Reading Out Temporal Information about Ongoing Motor Output from Recordings in Songbird Motor Cortex

The FP algorithm is a general and biologically plausible learning algorithm that we believe could support a variety of learning processes. Since it trains neurons to transform their inputs into a temporally specific output, it is particularly well suited for associating reproducible, temporally rich, sensory stimuli or motor behaviors with a clock (or timing) signal (Buonomano and Laje, 2010). To test the FP algorithm on such tasks using real spike trains, we turned to zebra finches, songbirds that learn a complex and temporally precise vocal output.

The timing of the bird's song is controlled by premotor nucleus HVC. Projection neurons in HVC represent time by firing sparse bursts of spikes at particular time points in the song (Hahnloser et al., 2002; Long et al., 2010), timing information that is relayed to the rest of the song system. Interestingly, the HVC timing network is activated both when the bird sings and when it listens to a song (Prather et al., 2008). Thus, the sparse time keepers in HVC could serve as "teachers" for downstream "student" neurons receiving sensory afference and/or motor efference. Such "student" neurons could learn to associate patterns of sensory or motor input with particular time points in the behavioral sequence, thus providing a potential substrate for sensorimotor learning.

We tested the feasibility of this idea by training a model neuron to fire at specific time points in the song in response to motor efference coming from RA, a motor cortex analog brain region that encodes the motor program underlying song. In particular, we were interested in the precision with which such a model neuron could reconstruct the timing of song elements. Functionally, this may be relevant since RA sends an efference copy to the anterior forebrain pathway (AFP) (Goldberg and Fee, 2012), a basal ganglia thalamocortical circuit essential for song learning (Bottjer et al., 1984) that also receives auditory and timing (HVC) input. Recent results suggest that the input from RA to the AFP may play an important role in song learning (Charlesworth et al., 2012). Hence, the precision with which song timing can be decoded from the activity of RA neurons may place constraints on the learning process.

The neural data we used came from recordings in two young adult zebra finches (91 and 111 days after hatch at the start of recording; 21 and 27 single units, respectively). Each neuron was recorded for at least 15 renditions of the song motif, and the spike trains were time warped with respect to the simultaneously recorded song (Figures 6A and 6B; Experimental Procedures).

We constructed an input pattern for a readout LIF neuron by randomly stacking trials from each recorded neuron in the same bird (Figure 6C; Experimental Procedures). We trained the LIF neuron to fire at desired times around the onset of syllables. The results (Figure 6D; Experimental Procedures) show that the task can be successfully implemented with a pool as small as 20–30 neurons.

To yield a quantitative estimate of the temporal precision of RA, we used the recorded neurons to construct a large pool of synthetic RA neurons. Specifically, we used the following

generative model: the activity of each synthetic neuron consisted of a stereotyped sequence of bursts, the number and structure of which were identical to those of one of our sampled neurons, while their onset times were drawn from a uniform distribution. Thus, by randomly permuting short segments of single neuron spike trains (Experimental Procedures), we generated a large number of synthetic neurons from each recorded neuron (Figure 6E).

We trained a readout neuron to fire at a single, randomly chosen time point in the song, within a tolerance ε, using the spike trains of the synthetic population. To assess the temporal accuracy of the readout neuron, we measured the mean readout error as a function of ε, the population size N, and the correlation time of the LIF neuron τ (Figure 5F, legend). Here the role of ε is analogous to discriminability in psychometric curves. For each N and τ, the error drops to zero with increasing ε as the task becomes easier. This drop occurs at a characteristic time scale, $\bar{\varepsilon}(\tau, N)$ (Figure 6F), which decreases with N, indicating that with increasing population size the information about time becomes more accurate. The results are consistent with $\bar{\varepsilon}(\tau, N) = \varepsilon_0(\tau) N^{-1/2}$, as expected from Fisher information theory of population coding in pools with no noise correlations (Seung and Sompolinsky, 1993). The prefactor $\varepsilon_0(\tau)$ can be interpreted as the mean single neuron temporal "imprecision" when filtered with time constant τ. Interestingly, $\varepsilon_0(\tau)$ decreases with τ, indicating that better accuracy can be achieved by considering the instantaneous firing activity of the input neuron (Figure S3; Supplemental Experimental Procedures). For the smallest τ in our simulations (τ = 5 ms), the single neuron imprecision parameter $\varepsilon_0(\tau)$ reaches a level of ~18 ms. Thus, to achieve a precision of 1 ms, an LIF readout neuron with integration time τ = 5 ms needs inputs from an RA population of size $N \approx 300$. For comparison, we have applied a maximal likelihood estimator of time along the song to segments of duration Δ of firing of N synthetic RA neurons. The root-mean-square error of this estimator behaves in large N as $\delta t(\Delta, N) = \delta_0 (N \bar{r} \Delta)^{-1/2}$ where $\bar{r} \approx 72$ Hz is the mean firing rate of an RA neuron and $\delta_0 \approx$ 7 ms (Supplemental Experimental Procedures). δ_0 can be interpreted as the mean temporal imprecision of a single spike. For a window size of Δ = 5 ms, this results in a single neuron imprecision $\delta_0 (\bar{r} \Delta)^{-1/2} \approx 12$ ms, compared to 18 ms in the LIF readout (Figure S3D). This indicates that the amount of temporal information extracted by a readout neuron with integration time of a few milliseconds is not far below the bound set by an ideal observer using spikes arriving in a temporal window of a similar size (Figures S3A and S3D).

Finite Precision Learning in Recurrent Networks

Training a recurrent network of neurons to reproduce a desired input-output mapping is more involved than training a single neuron in a feedforward architecture. In a recurrent network, each neuron has to generate its desired output in response to input spikes that are themselves dependent on the neuron's synaptic weights in a highly nonlinear fashion. The problem of supervised learning of dynamic trajectories in deterministic nonspiking recurrent networks has been addressed before (Jaeger and Haas, 2004; Sussillo and Abbott, 2009; Rumelhart et al., 1986; Laje and Buonomano, 2013). Here we propose FP learning for spiking networks with both recurrent and feedforward sets of plastic synapses. The recurrent network is required to generate desired spike patterns within a given tolerance, in response to patterns of spiking inputs from the afferent neurons. To highlight the role of the recurrent dynamics, we chose the desired spike patterns to persist long after the termination of the transient afferent activity (Figure 7A). In each training trial, the first error triggers synaptic modification according to the FP learning rule in the synapses of the erroneous neuron (Figure 7A).

FP learning is able to successfully implement the required dynamics. To demonstrate this, we trained a fully connected recurrent network to produce a specific periodic activity cycle in response to one input pattern (Figure 7B) and another periodic cycle in response to a different input pattern (Figure 7C). The network was only trained to produce two cycles of each pattern; however, after learning, the network continues to produce the periodic pattern in a stable manner even beyond this time (Figures 7B and 7C; Experimental Procedures). Interestingly, FP learning tends to find solutions that generate the required patterns as stable trajectories of the network dynamics, robust to small perturbations (Experimental Procedures; Figures S4A and S4C). In contrast, learning spike patterns in recurrent networks with the HTP algorithm, which does not use the natural spiking dynamics during training, often yields unstable solutions, even if stable solutions exist. As an example, we used the HTP algorithm to train a network to reproduce the spike times learned by the FP algorithm. Even when the learning of all the neurons converged successfully, the dynamics of the network was unstable and the network was unable to recall the trained pattern (Experimental Procedures; Figure S4A).

The ability of the FP learning algorithm to generate a desired periodic attractor in the recurrent network critically depends on the duration of the training sequence. For short training sequences, the network finds "transient" solutions, i.e., solutions that do not produce the required dynamics beyond the training sequence. When the length of the training sequence increases beyond a critical value, which we term the "learning horizon," there is a sharp transition to "infinite time" solutions, which permanently produce the required periodic activity pattern (Figure S4D). The learning horizon depends only weakly on the number of neurons in the network (Supplemental Experimental Procedures; Figure S4E) but may depend on the period of the required pattern T_p and the membrane integration time τ. We observe two qualitatively different regimes. For values of T_p/τ of order 1 or less, the learning horizon in units of τ does not depend on T_p/τ (Figure S4E). In this regime, the temporal correlations in the membrane potentials extend over several periods of the desired activity and the network must learn a sufficiently long sequence to account for these correlations. In contrast, for $T_p/\tau > 2$, the learning horizon in units of τ increases with increasing T_p/τ (Figure S4E). In the limit of $T_p/\tau \gg 1$, we expect the learning horizon to be approximately a single period, since the membrane dynamics during the first period and the beginning of the second period repeats itself in subsequent periods.

We have also applied FP learning to the problem of learning delay line architectures from an initial recurrent connectivity. Such architectures are useful for working memory tasks (White et al., 2004; Ganguli et al., 2008; Goldman, 2009; Harvey et al.,

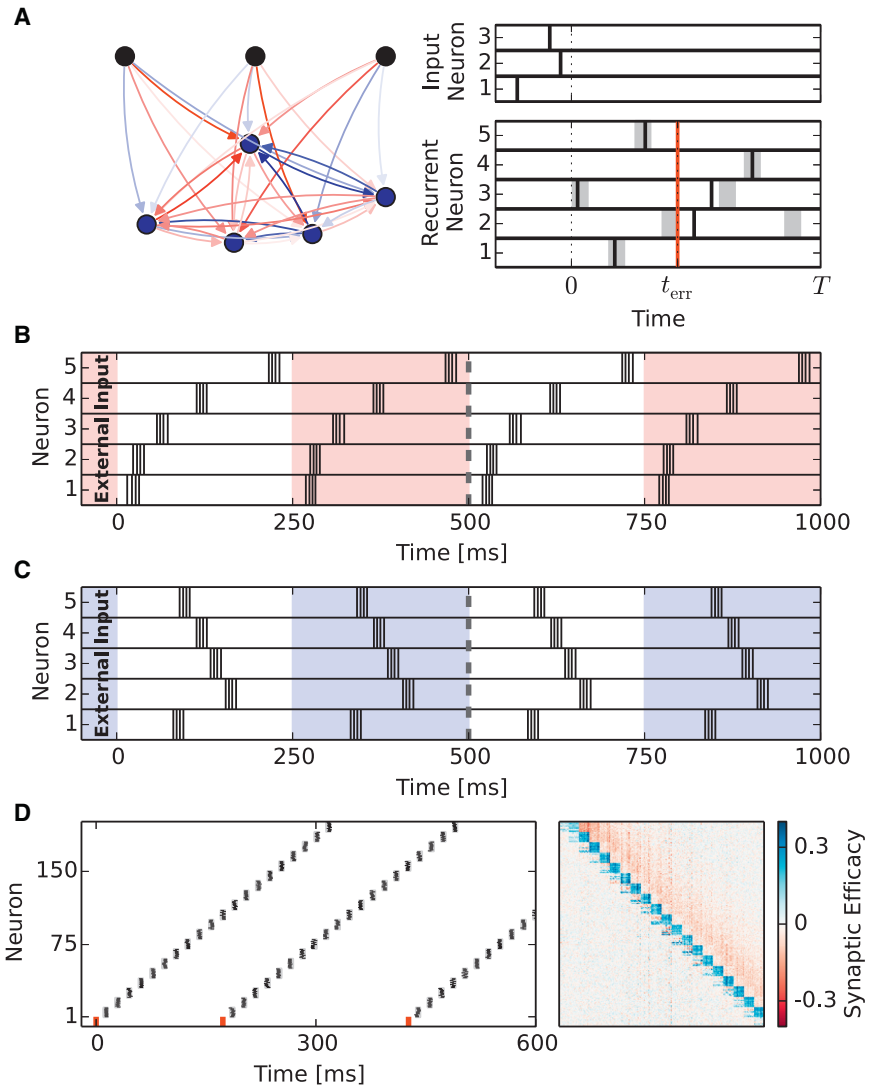

Figure 7. Finite Precision Learning in Recurrent Networks

(A) Network architecture and task. Input neurons (black) are connected via feedforward connections to a recurrent network (blue). Inhibitory and excitatory synaptic connections are depicted as red and blue arrows, respectively. Color intensity is proportional to the synaptic efficacy. The recurrent network's neurons are required to spike within the tolerance windows depicted in gray (right, bottom). Finite Precision learning is performed on feedforward and recurrent synapses; in each trial, only the synapses of the neuron responsible for the first error are modified according to Equation 6, with respect to the first error's time, t_{err}.

(B and C) Example of two stable, periodic patterns implemented by a recurrent network. Periods of the required output spikes are depicted by the shaded areas. The spike trains of the first five neurons are displayed (vertical lines); spike trains of the remaining neurons are similar. Each pattern is initiated by a specific sequence of external input spikes (External Inputs I and II). The spike times of the recurrent neurons are learned up to time T = 500 ms (dashed vertical line). After learning, the required patterns extend indefinitely. See also Figures S4A and S4B.

(D) Learned delay line memory in a recurrent network. Left: the network's activity after training. The network responds to random external input (red spikes, bottom) by successive synchronous spiking of 20 groups of 10 neurons (black dots). Desired tolerance windows are depicted in gray. Right: the learned synaptic connectivity matrix.

2012). The recurrent network learns to retain the timing of external stimulations (which has refractory Poisson statistics) by responding with successive synchronous spiking in assigned neuron groups (Figure 7D, left). We find that 95% of the trained networks were able to perform well even for sequences of inputs not seen during training (Experimental Procedures). This generalization ability implies that the resultant architecture resembles that of a delay line. Indeed, the learnt synaptic matrix has a pronounced feedforward structure, with excitatory connections from one group to the next and inhibition from earlier groups (Figure 7D, right). There are also weak excitatory and inhibitory connections within groups, while the remaining recurrent connections are close to zero.

DISCUSSION

Previous work (Marr, 1969; Bressloff and Taylor, 1992; Brunel et al., 2004; Clopath et al., 2012) used the theory and learning algorithm of the Perceptron to evaluate the capability of a neuron to generate desired spike trains. These studies, however, used discrete time bins and found that correlations between inputs in nearby time bins do not necessarily affect the neuron's capacity. Thus, the predicted capacity would depend on the time discretization. In addition, these studies did not address the nonlinearity associated with spike reset. In this work, we have introduced a geometric characterization of the computation performed by spiking networks using the LIF model. Using our HTP algorithm, we were able to show for the first time that the capacity of a continuous time spiking neuron is extensive, namely the total duration of output spikes patterns that can be learned by the neuron scales linearly with the number of synapses.

The setting of random input and output spike trains in an LIF network has a number of parameters, input and output firing rates, durations and number of input-output sequences, number of afferents, and synaptic and membrane time constants. Here we show that in a broad range of biologically relevant parameter values, the capacity depends only on a small subset of them and obeys a simple scaling property, as depicted in Equation 4 and Figure 3.

The FP learning algorithm, applicable both to feedforward and recurrent network topologies, addresses several challenges in

learning precise spike sequences. It incorporates a temporal tolerance parameter and circumvents the nonlinear error accumulation due to reset by learning only from the first error in each learning episode. This is reminiscent of the recently developed FORCE algorithm for nonspiking networks (Sussillo and Abbott, 2009; Laje and Buonomano, 2013), where, during training, the network is forced to produce the required dynamics, within some small error. Another desirable feature of the FP learning algorithm is that when the tolerance window is small, the capacity obtained by it approaches the rigorous capacity of neurons required to generate exact spike timing. Finally, learning a desired sequence does not in general guarantee that the sequence will be reproduced in a stable manner, a potentially severe problem in recurrent architecture. We have observed that the recurrent network solutions found by the FP rule are stable to small perturbations (Figure S4) and exhibit generalization abilities (Figure 7D and Figure S4C). Presumably, the explicit incorporation of the tolerance windows introduces variability in the spike timing during training, which forces the learning dynamics to converge onto a stable solution, similar to the stabilizing effects of noise on learning in nonspiking recurrent networks (Jaeger and Haas, 2004; Jaeger et al., 2007; Sussillo and Abbott, 2009). The performance of FP learning in recurrent networks and its scaling with network size and integration time constant deserve further studies.

Supervised, spike-time-based learning algorithms using stochastic neuronal dynamics have been proposed with continuous time (Xie and Seung, 2004; Fiete and Seung, 2006; Pfister et al., 2006) and discrete time (Brea et al., 2013). More closely related to our work are recent heuristic algorithms for deterministic spiking neurons with continuous time dynamics (Ponulak and Kasiński, 2010; Florian, 2012; Mohemmed et al., 2012; Xu et al., 2013). However, the convergence and capacity of these learning rules have been demonstrated only in limited examples.

A previous spike-time-based model, the Tempotron (Gütig and Sompolinsky, 2006, 2009; Rubin et al., 2010), requires a neuron to learn to classify input spike patterns by firing or not firing during the pattern presentation. Although the FP model allows some freedom in the timing of spikes, the FP learning algorithm differs substantially from the Tempotron learning algorithm. First, unlike the Tempotron, we do not allow more than one spike within the tolerance window. Second, the present learning incorporates temporal patterns consisting of more than one spiking window. To guarantee convergence under additional constraints (Experimental Procedures), weight potentiation is tagged to the end of the tolerance window and not to the maximum of the potential as in the Tempotron. Thus, our FP learning is appropriate for modest values of ε where these constraints are expected to be met in many biologically relevant scenarios. Furthermore, other tagging scenarios within the tolerance windows do not degrade performance substantially (data not shown). For large tolerance windows, learning from the maximum potential is expected to be superior.

Our learning algorithm offers a simple method for reconstruction of synaptic weights from observed spike times. In particular, it uses a standard neuron model with a clear biophysical interpretation, incorporating full voltage reset, and synaptic currents with finite time constant and a strength characterized by a single amplitude. The simple reconstruction example shown here with feedforward networks can also be used in a recurrent circuit. Recent studies of connectivity reconstruction from spiking activity used more complex stochastic generative models for the circuit (Pillow et al., 2008; Gerwinn et al., 2010). Their biophysical interpretation is less clear, particularly since these models typically represent each synapse by a modifiable temporal filter. Other LIF-based experimental procedures for weight reconstruction are restricted to pulse-like synaptic currents (Monasson and Cocco, 2011; Van Bussel et al., 2011; Memmesheimer and Timme, 2006), assumed constant external input (Van Bussel et al., 2011), or are limited to very small networks with restrictive reset schemes (Makarov et al., 2005).

The realization of FP learning by a biological system requires the presence of a supervisory signal that detects the first error produced by the network in each episode. The most plausible mechanism would be comparing the network output with an internally stored template of the desired sequence. The restriction of learning to the first error can be relaxed to multiple learning events as long as they are well separated such that nonlinear interactions between the corresponding errors are minimized. Biologically, this can be implemented by refractoriness in the error signals.

Most current models of neurons' computational capacity are based on averaging inputs and outputs over long time windows. These models neglect the dynamic features of neuronal integration and spiking as well as the potential coding of information in the spike times. Precisely timed patterns of spikes carrying sensory information have been experimentally found in various neural systems (Kayser et al., 2009; Jones et al., 2004; Johansson and Birznieks, 2004; Gollisch and Meister, 2008). In the mammalian motor cortex, their occurrence correlates with internal cognitive states and task performance (Riehle et al., 1997; Putrino et al., 2010) and in the motor cortex of songbirds, they govern the song generation process (Yu and Margoliash, 1996; Leonardo and Fee, 2005). Our work shows that simple circuits of spiking neurons can robustly implement and learn temporally precise codes under biologically realistic conditions.

EXPERIMENTAL PROCEDURES

Implementing Precise Spike Time Input-Output Associations

For convenience, we adopt here the notation in which the threshold and reset are represented by an $N + 1$-th component of the input vector $\mathbf{x}(t)$, $x_{N+1}(t) = -1 - \sum_{t_d<t} u_r(t - t_d)$ and an $N + 1$-th component of $\boldsymbol{\omega}$, $\omega_{N+1} = U_{\text{thr}}$. With this notation, the dynamics can be expressed as

$$U(t) - U_{\text{thr}} = \boldsymbol{\omega}^T \mathbf{x}(t) = \sum_{i=1}^{N+1} \omega_i x_i(t) \,. \qquad \text{(Equation 5)}$$

$U(t)$ must satisfy the following sets of constraints: (1) $U(t_d) = U_{\text{thr}}$, (2) $\mathrm{dU/dt}(t_d) > 0$, and (3) $U(t) < U_{\text{thr}}$ at all times other than t_d. Note that, in order to ensure a robust solution, we demand strict inequality in (3). In the definition of $x_{N+1}(t)$, the reset times are fixed at the desired times. This makes $U(t)$ a linear function of $\boldsymbol{\omega}$, hence, (1)–(3) are linear constraints. Note that once these linear constraints are fulfilled, the neuron will spike at and only at t_d. We thus treat (1)–(3) as defining the space of solutions for $\boldsymbol{\omega}$.

Denoting the total number of desired output spikes by n_{spikes}, the equality constraints (1) define an $N_{\text{eff}} \geq N - n_{\text{spike}}$ dimensional linear subspace of all vectors $\boldsymbol{\omega}$ that are orthogonal to all n_{spikes} vectors $\mathbf{x}(t_d)$ (the permitted subspace). Assuming for simplicity that the vectors $\mathbf{x}(t_d)$ are linearly independent,

the subspace is defined by a projection matrix $P = I - X(X^TX)^{-1}X^T$ where X is the $(N+1) \times n_{spikes}$ matrix defined by $\mathbf{x}(t_d)$. Thus, a solution weight vector must lie inside the permitted subspace and additionally obey the set of linear inequalities (2) and (3).

In the Supplemental Experimental Procedures, we prove that within the permitted subspace, all solutions are robust, i.e., a small change in the solution weight vector $\boldsymbol{\omega}$ does not invalidate the solution. The existence of this "finite margin" property is important for robustness and learning (Vapnik, 2000).

HTP Algorithm

The HTP algorithm imposes constraints (1) at all times by applying the projection matrix P. To implement (3) in Perceptron-like learning, the continuous time errors associated with violation of (3) need to be appropriately subsampled. HTP uses an efficient bootstrap process in which a subsample of the errors associated with (3) and the errors associated with (2) induce Perceptron-like synaptic modification in the permitted subspace. The algorithm and its convergence proof are described in detail in the Supplemental Experimental Procedures.

Capacity of LIF Neurons

Random Input-Output Patterns

For each input afferent, spike trains of duration T_{input} were drawn from a homogeneous Poisson process with rate $r_{in} = 5$ Hz. Desired output spikes were not allowed between $t = 0$ and $t = \tau_m$. To maintain the mean output firing rate r_{out}, we drew desired spike times for $t \in (\tau_m, T_{input})$ from a homogeneous Poisson process with rate $r_{out}/(1 - (\tau_m/T_{input}))$.

Capacity Measurements

In Figure 3 and Figure S1, capacity was measured by increasing the total duration of the input and estimating the point at which the probability of convergence within $n_{iter} = 5 \times 10^7$ iterations is 1/2. One iteration is the presentation of all the patterns in the current set. The convergence probability was estimated by the fraction of converged simulations out of $n_{sim} = 50$ simulations. For the data shown, we have verified that the maximum number of iterations does not affect the capacity substantially.

Analytical Estimation of Capacity

The detailed derivation of the theoretical capacity estimation that is depicted as the dashed line in Figure 3D is given in the Supplemental Experimental Procedures.

Learning Spike Times with Finite Precision

FP Learning Algorithm

The LIF dynamics can be expressed as in Equation 5 with $x_{N+1}(t) = -1 - \sum_{t_{spike}<t} u_r(t - t_{spike})$, where t_{spike} are the threshold crossing times given the input and the weights. The task of the neuron is to spike once within each spiking window $[t_d - \varepsilon/2, t_d + \varepsilon/2]$. Our FP learning algorithm consists of a sequence of learning trials, in each of which one pattern is presented. After each trial, weights are modified according to

$$\Delta\omega_i = \pm \eta x_i(t_{err}), \qquad \text{(Equation 6)}$$

where t_{err} is the time of the first error, $\eta > 0$ is a constant learning rate and +/− is used for a missed/undesired spike error, respectively. If the first error consists of an output spike outside the tolerance window or more than one spike inside it, t_{err} is the time of the extra spike. If the first error consists of a failure to spike within a tolerance window, t_{err} is the end of the tolerance window (Figures 4B–4E).

Reconstruction of Synaptic Weights

Input Patterns

The input layer consisted of $N = 1{,}000$ input neurons. The spike times of each input neuron were randomly sampled from homogeneous Poisson processes with mean firing rate $r_{in} = 10$ *Hz*. The duration of each input pattern was chosen to be $T_{input} = 1$ s.

Distribution of the Teacher's Synaptic Conductances

Maximal synaptic conductances for the teacher neurons, $g_i^{teacher}$, where randomly chosen from a log-normal distribution with shape parameter $\sigma = 0.97$ (Song et al., 2005). Eighty percent of the synapses were chosen to be excitatory and 20% inhibitory. To maintain the excitation-inhibition balance, we chose the scale of the distribution of inhibitory conductances to be five times the scale of the distribution of excitatory conductances (Heiss et al., 2008). The two scale parameters were adjusted to ensure a mean output firing rate of ~10 Hz given the synaptic input.

For the LIF teacher neuron, synaptic efficacies (in units of voltage), $\omega_i^{teacher}$, were derived from the synaptic conductances according to $\omega_i^{teacher} = \Delta g_i^{teacher}$, where Δ is the driving force ($\Delta = 55$ mV for excitatory and $\Delta = -25$ mV for inhibitory synapses).

Teacher Neuron Dynamics

For the teacher LIF neuron, we used $\tau_m = 15$ ms and $\tau_s = 5$ ms. For the dynamics of the HH neuron we used a Wang-Buzsaki neuron (Wang and Buzsáki, 1996). The synaptic input current was modeled as

$$I_{synaptic}(t) = (E_i - U(t)) \sum_{i=1}^{N} g_i \sum_{t_i<t} e^{-\frac{t-t_i}{\tau_s}}, \qquad \text{(Equation 7)}$$

where $U(t)$ is the membrane potential at time t and g_i and E_i are the maximal conductance and the reversal potential of the synapse of afferent i, respectively. E_i was taken to be $E_{ex} = 0$ mV for excitatory synapses and $E_{in} = -80$ mV for inhibitory synapses. The synaptic time constant was taken to be $\tau_s = 5$ ms.

Learning

Initial weights for the student neuron were randomly chosen as described for the LIF teacher neuron. For the results in Figure 5, learning was performed on a batch of 3,000 input patterns. The teacher output spikes were taken as desired spikes and learned with a tolerance window of size ε centered around the desired time. An adaptive learning rate was used according to the protocol suggested in Barkai et al. (1995) with parameters $A = 0.005$ and $\lambda = 0.01$. A maximal number of $n_{iter} = 10^6$ pattern presentations were performed. To evaluate the quality of the synaptic weights reconstruction, we measured R^2 values for the teacher's excitatory and inhibitory synapses separately. R^2 was defined as $R^2 = 1 - \langle (g_i^{student} - g_i^{teacher})^2 \rangle / \mathrm{var}(g_i^{student})$ where $\langle x \rangle$ and $\mathrm{var}(x)$ are the empirical mean and variance of x respectively (calculated over the relevant set). The R^2 values presented in Figure 5 and Figure S2 are averaged over the weight reconstruction of 100 teacher neurons.

Learning from an LIF Teacher

When learning spikes from an LIF teacher, the student neuron's time constants were the same as the teacher neuron's. The tolerance window size was taken to be $\varepsilon = 3$ ms. After training the synaptic weights were normalized with respect to the neuron's threshold.

Learning from an HH Teacher

When learning from an HH neuron, the student's LIF dynamics were slightly modified: after an output spike, the LIF membrane potential was reset to $-AU_{thr}$. $A > 0$ implements the afterhyperpolarization displayed by the Wang-Buzsaki neuron after an action potential. This modification does not affect the convergence properties of the FP algorithm. The parameters of the student dynamics, τ_m, τ_s, and A were optimized to yield the best weights reconstruction as described below.

Learning was performed with tolerance window size ε, its value was also optimized (see below). Since for HH teacher learning does not converge to zero error, we take the time average of the weights during the second half of training as reconstructed weights.

To convert the current-based reconstructed weights to synaptic conductances, we use

$$g_i^{student} = \frac{1}{\tau_s} \frac{\omega_i^{student}}{U_{thr}} \frac{V}{(E_i^{student} - E_0)} \qquad \text{(Equation 8)}$$

with $E_i^{student} = E_{ex}$ for $\omega_i^{student} > 0$ and $E_i^{student} = E_{in}$ for $\omega_i^{student} < 0$. V is a global scale parameter and E_0 controls the ratio between the driving force of excitatory and inhibitory synapses.

The values of τ_m, τ_s, A, ε, V, and E_0 were optimized to minimize the error in spiking activity between the true teacher's spikes and the spikes generated by a student HH neuron using the reconstructed conductances (Equation 8). To measure the error, we used the spike distance metric $D^{spike}[q]$ proposed in Victor and Purpura (1997) with $q^{-1} = 75$ ms. Mean $D^{spike}[q]$ per desired output spike was estimated using 25 untrained input patterns for

each teacher neuron. For each set of parameters, the performance was averaged over 100 teacher neurons. Optimal performance ($D^{spike}[q] \approx 0.12$ per spike) was found for τ_m = 14 ms, τ_s = 6 ms, A = 0.6, ε = 7 ms, V = 30 mV, and $E_0 = -54.3$ mV.

Reading Out Temporal Information about Ongoing Motor Output from Recordings in Songbird Motor Cortex

Neuronal Data

Experimental procedures for the recording from single units in RA of singing songbirds have been previously published (Ölveczky et al., 2011). Recorded spike times were time warped using a piecewise linear transformation matching the onset of the syllables in each song rendition to a template song with average syllable onset times (Ölveczky et al., 2011). Syllable onsets were determined by thresholding the log power of the acoustic signal and verifying that all onsets were detected correctly by comparing them to the song's spectrogram.

Input Patterns

Input patterns were constructed by randomly selecting a single recorded spike train from each neuron. In Figure 6F, mean errors are averaged over n_{sim} = 100 simulations. For each simulation, the recordings of each neuron were randomly partitioned: 80% were used to construct training patterns and the remaining 20% were used to construct patterns for testing generalization error. $P_{train} = 20N$ training patterns were used, where N is the number of input neurons. Generalization error was estimated using P_{gen} = 200 test patterns.

Synthetic Neurons

Synthetic neurons were created according to the following procedure. First the PSTHs of the recorded neurons were calculated with time bin size of 3 ms. The PSTHs were then segmented by detecting upcrossings above a threshold level of $r_{thr} = 0.125 \cdot r_{max}$ where r_{max} is the maximal firing rate of the recorded neuron. The order of the resulting time segments was then randomly shuffled. Each recorded spike train was used to generate a single spike train of the new synthetic neuron by shifting the spikes in each time segment according to the time segment's new position. This procedure generates neurons in which the number and structure of the "parent" neuron's activity bursts are preserved but the temporal modulation of the firing is different.

Learning Parameters

Learning was performed using the FP algorithm. Initial weights were randomly drawn from a standard normal distribution, initial threshold was U_{thr} = 1, and the learning rate was taken to be η = 0.01. In each simulation, n_{iter} = 25,000 iterations were performed. If the learning did not converge after n_{iter} iterations, the weight vector for which the training distance was minimal was selected. Training distance was defined as the mean distance, $D^{spike}[q]$, between the LIF output spikes and the desired times over the entire training set, with q^{-1} = 0.075 s.

In Figure 5D training was performed on P_{train} = 400 training patterns. In this task τ = 0.02 s and ε = 0.01 s were found to yield optimal performance.

FP Learning in Recurrent Networks

Network and Learning Parameters

Figures 7B and 7C illustrate pattern recall in a recurrent, fully connected network of N = 100 recurrent neurons and N_{ext} = 10 external neurons. The network stores two periodic patterns, both with period T_p = 250 ms. In each pattern, each neuron emits one burst of four spikes with uniformly distributed random starting time within the period. The interspike interval within a burst is t_{ISI} = 5 ms, the tolerance is $\varepsilon = t_{ISI}$. Each pattern is initiated by a specific sequence of external input spikes of length t_{init} = 50 ms. Within the initialization sequence, each external neuron emits one spike at a random time drawn from a uniform distribution on $[-t_{init}, 0]$. The spike times of the recurrent neurons are learned up to time $T = 2T_p$. Recurrent connection strengths as well as strengths of afferents from external neurons are modified according to recurrent FP learning. Initial weights are normally distributed with mean zero and SD 0.5; all initial thresholds are set to 1. Membrane and synaptic time constants are τ_m = 20 ms and τ_s = 5 ms, respectively. The learning rate is η = 1. HTP learning in Figure S4B uses the same initial weights and learning rate.

Stability of Learned Patterns

We verify the stability of the learned periodic orbits by continuing the recall beyond the learning horizon and by perturbing the network. For details, see the Supplemental Experimental Procedures.

Delay Line Learning

Learning of the delay line architecture (Figure 7D) was performed with a network of N = 200 identical LIF neurons with τ_m = 40 ms and τ_s = 10 ms. The network was divided into 20 groups of ten neurons each. The groups were trained to spike synchronously in succession after an external input spike from a single input afferent, with Δt = 16 ms temporal difference and 6 ms tolerance window size, generating delay line memory of duration T_{DL} = 320 ms. Input spikes were drawn from a Poisson process with rate $2/T_{DL}$ and refractory period 0.5 T_{DL}. The training set consisted of 400, 4 s long, input spike trains. Initial synaptic weights were drawn from a zero mean normal distribution with SD 0.1; initial thresholds were chosen to be 1 and η = 0.1. Generalization performance was tested on 100, 200 s long, input spike trains for 50 networks trained with different input patterns and initial weights. In general, we observe no more than about ten missing/additional spikes in the entire network within the 200 s long pattern. In two networks, we observed a few trials in which the error in the spiking activity causes the network activity to diverge.

SUPPLEMENTAL INFORMATION

Supplemental Information includes Supplemental Experimental Procedures and four figures and can be found with this article online at http://dx.doi.org/10.1016/j.neuron.2014.03.026.

AUTHOR CONTRIBUTIONS

R.-M.M. and R.R. contributed equally to this paper. R.-M.M., H.S., and R.R. wrote the manuscript, developed the learning algorithms, the convergence proofs, and the applications. R.R. and R.-M.M. performed the numerical simulations. H.S. supervised the work. B.P.O. provided the single unit recordings from songbird and contributed to analyzing and interpreting the data. All authors discussed the results and implications, and contributed to the manuscript.

ACKNOWLEDGMENTS

We thank R. Gütig, Y. Burak, and P. Tiesinga for helpful discussions. We thank Timothy M. Otchy for sharing the RA recordings. Work is supported in part by the Gatsby Charitable Foundation, the Israel Defense Ministry (MAFAT), the Max Planck Hebrew University Center, and the Swartz Foundation.

Accepted: March 18, 2014
Published: April 24, 2014

REFERENCES

Albus, J.S. (1971). A theory of cerebellar function. Math. Biosci. *10*, 25–61.

Alivisatos, A.P., Chun, M., Church, G.M., Greenspan, R.J., Roukes, M.L., and Yuste, R. (2012). The brain activity map project and the challenge of functional connectomics. Neuron *74*, 970–974.

Barkai, N., Seung, H.S., and Sompolinsky, H. (1995). Local and global convergence of on-line learning. Phys. Rev. Lett. *75*, 1415–1418.

Bottjer, S.W., Miesner, E.A., and Arnold, A.P. (1984). Forebrain lesions disrupt development but not maintenance of song in passerine birds. Science *224*, 901–903.

Brea, J., Senn, W., and Pfister, J.P. (2013). Matching recall and storage in sequence learning with spiking neural networks. J. Neurosci. *33*, 9565–9575.

Bressloff, P.C., and Taylor, J.G. (1992). Temporal sequence storage capacity of time-summating neural networks. J. Phys. Math. Gen. *25*, 833.

Brunel, N., Hakim, V., Isope, P., Nadal, J.P., and Barbour, B. (2004). Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cell. Neuron *43*, 745–757.

Buonomano, D.V., and Laje, R. (2010). Population clocks: motor timing with neural dynamics. Trends Cogn. Sci. *14*, 520–527.

Chapeton, J., Fares, T., LaSota, D., and Stepanyants, A. (2012). Efficient associative memory storage in cortical circuits of inhibitory and excitatory neurons. Proc. Natl. Acad. Sci. USA *109*, E3614–E3622.

Charlesworth, J.D., Warren, T.L., and Brainard, M.S. (2012). Covert skill learning in a cortical-basal ganglia circuit. Nature *486*, 251–255.

Clopath, C., Nadal, J.P., and Brunel, N. (2012). Storage of correlated patterns in standard and bistable Purkinje cell models. PLoS Comput. Biol. *8*, e1002448.

Dayan, P., and Abbott, L. (2005). Theoretical Neuroscience: Computational and Mathematical Modeling of Neural systems. (Cambridge: MIT Press).

Fiete, I.R., and Seung, H.S. (2006). Gradient learning in spiking neural networks by dynamic perturbation of conductances. Phys. Rev. Lett. *97*, 048104.

Florian, R.V. (2012). The chronotron: a neuron that learns to fire temporally precise spike patterns. PLoS ONE *7*, e40233.

Ganguli, S., Huh, D., and Sompolinsky, H. (2008). Memory traces in dynamical systems. Proc. Natl. Acad. Sci. USA *105*, 18970–18975.

Gardner, E. (1988). The space of interactions in neural network models. J. Phys. A *21*, 257–270.

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

Gerwinn, S., Macke, J.H., and Bethge, M. (2010). Bayesian inference for generalized linear models for spiking neurons. Front Comput Neurosci *4*, 12.

Goldberg, J.H., and Fee, M.S. (2012). A cortical motor nucleus drives the basal ganglia-recipient thalamus in singing birds. Nat. Neurosci. *15*, 620–627.

Goldman, M.S. (2009). Memory without feedback in a neural network. Neuron *61*, 621–634.

Gollisch, T., and Meister, M. (2008). Rapid neural coding in the retina with relative spike latencies. Science *319*, 1108–1111.

Gütig, R., and Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing-based decisions. Nat. Neurosci. *9*, 420–428.

Gütig, R., and Sompolinsky, H. (2009). Time-warp-invariant neuronal processing. PLoS Biol. *7*, e1000141.

Hahnloser, R.H., Kozhevnikov, A.A., and Fee, M.S. (2002). An ultra-sparse code underlies the generation of neural sequences in a songbird. Nature *419*, 65–70.

Harvey, C.D., Coen, P., and Tank, D.W. (2012). Choice-specific sequences in parietal cortex during a virtual-navigation decision task. Nature *484*, 62–68.

Heiss, J.E., Katz, Y., Ganmor, E., and Lampl, I. (2008). Shift in the balance between excitation and inhibition during sensory adaptation of S1 neurons. J. Neurosci. *28*, 13320–13330.

Jaeger, H., and Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science *304*, 78–80.

Jaeger, H., Lukosevicius, M., Popovici, D., and Siewert, U. (2007). Optimization and applications of echo state networks with leaky-integrator neurons. Neural Netw. *20*, 335–352.

Johansson, R.S., and Birznieks, I. (2004). First spikes in ensembles of human tactile afferents code complex spatial fingertip events. Nat. Neurosci. *7*, 170–177.

Jones, L.M., Depireux, D.A., Simons, D.J., and Keller, A. (2004). Robust temporal coding in the trigeminal system. Science *304*, 1986–1989.

Kayser, C., Montemurro, M.A., Logothetis, N.K., and Panzeri, S. (2009). Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns. Neuron *61*, 597–608.

Laje, R., and Buonomano, D.V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. Nat. Neurosci. *16*, 925–933.

Leonardo, A., and Fee, M.S. (2005). Ensemble coding of vocal control in birdsong. J. Neurosci. *25*, 652–661.

Lichtman, J.W., and Denk, W. (2011). The big and the small: challenges of imaging the brain's circuits. Science *334*, 618–623.

Long, M.A., Jin, D.Z., and Fee, M.S. (2010). Support for a synaptic chain model of neuronal sequence generation. Nature *468*, 394–399.

Makarov, V.A., Panetsos, F., and de Feo, O. (2005). A method for determining neural connectivity and inferring the underlying network dynamics using extracellular spike recordings. J. Neurosci. Methods *144*, 265–279.

Marr, D. (1969). A theory of cerebellar cortex. J. Physiol. *202*, 437–470.

Memmesheimer, R.M., and Timme, M. (2006). Designing the dynamics of spiking neural networks. Phys. Rev. Lett. *97*, 188101.

Minsky, M., and Papert, S. (1988). Perceptrons: Expanded Edition. (Cmbridge: MIT Press).

Mohemmed, A., Schliebs, S., Matsuda, S., and Kasabov, N. (2012). Span: spike pattern association neuron for learning spatio-temporal spike patterns. Int. J. Neural Syst. *22*, 1250012.

Monasson, R., and Cocco, S. (2011). Fast inference of interactions in assemblies of stochastic integrate-and-fire neurons from spike recordings. J. Comput. Neurosci. *31*, 199–227.

Monasson, R., and O'Kane, D. (1994). Domains of solutions and replica symmetry breaking in multilayer neural networks. Europhys. Lett. *27*, 85–90.

Ölveczky, B.P., Otchy, T.M., Goldberg, J.H., Aronov, D., and Fee, M.S. (2011). Changes in the neural control of a complex motor sequence during learning. J. Neurophysiol. *106*, 386–397.

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

Pillow, J.W., Shlens, J., Paninski, L., Sher, A., Litke, A.M., Chichilnisky, E.J., and Simoncelli, E.P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature *454*, 995–999.

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

Prather, J.F., Peters, S., Nowicki, S., and Mooney, R. (2008). Precise auditory-vocal mirroring in neurons for learned vocal communication. Nature *451*, 305–310.

Putrino, D., Brown, E.N., Mastaglia, F.L., and Ghosh, S. (2010). Differential involvement of excitatory and inhibitory neurons of cat motor cortex in coincident spike activity related to behavioral context. J. Neurosci. *30*, 8048–8056.

Riehle, A., Grün, S., Diesmann, M., and Aertsen, A. (1997). Spike synchronization and rate modulation differentially involved in motor cortical function. Science *278*, 1950–1953.

Rosenblatt, F. (1962). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. (Washington, DC: Spartan Books).

Rubin, R., Monasson, R., and Sompolinsky, H. (2010). Theory of spike timing-based neural classifiers. Phys. Rev. Lett. *105*, 218102.

Rumelhart, D.E., Hintont, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. Nature *323*, 533–536.

Seung, H.S., and Sompolinsky, H. (1993). Simple models for reading neuronal population codes. Proc. Natl. Acad. Sci. USA *90*, 10749–10753.

Song, S., Sjöström, P.J., Reigl, M., Nelson, S., and Chklovskii, D.B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. PLoS Biol. *3*, e68.

Sussillo, D., and Abbott, L.F. (2009). Generating coherent patterns of activity from chaotic neural networks. Neuron *63*, 544–557.

Van Bussel, F., Kriener, B., and Timme, M. (2011). Inferring synaptic connectivity from spatio-temporal spike patterns. Front. Comput. Neurosci. *5*, 3.

Vapnik, V. (2000). The Nature of Statistical Learning Theory. (New York: Springer).

Victor, J., and Purpura, K. (1997). Metric-space analysis of spike trains: theory, algorithms and application. Network *8*, 127–164.

Wang, X.J., and Buzsáki, G. (1996). Gamma oscillation by synaptic inhibition in a hippocampal interneuronal network model. J. Neurosci. *16*, 6402–6413.

White, O.L., Lee, D.D., and Sompolinsky, H. (2004). Short-term memory in orthogonal neural networks. Phys. Rev. Lett. *92*, 148102.

Xie, X., and Seung, H.S. (2004). Learning in neural networks by reinforcement of irregular spiking. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. *69*, 041909.

Xu, Y., Zeng, X., and Zhong, S. (2013). A new supervised learning algorithm for spiking neurons. Neural Comput. *25*, 1472–1511.

Yu, A.C., and Margoliash, D. (1996). Temporal hierarchical control of singing in birds. Science *273*, 1871–1875.

Neuron, Volume *82*

Supplemental Information

Learning Precisely Timed Spikes

Raoul-Martin Memmesheimer, Ran Rubin, Bence P. Ölveczky, and Haim Sompolinsky

SUPPLEMENTAL DATA

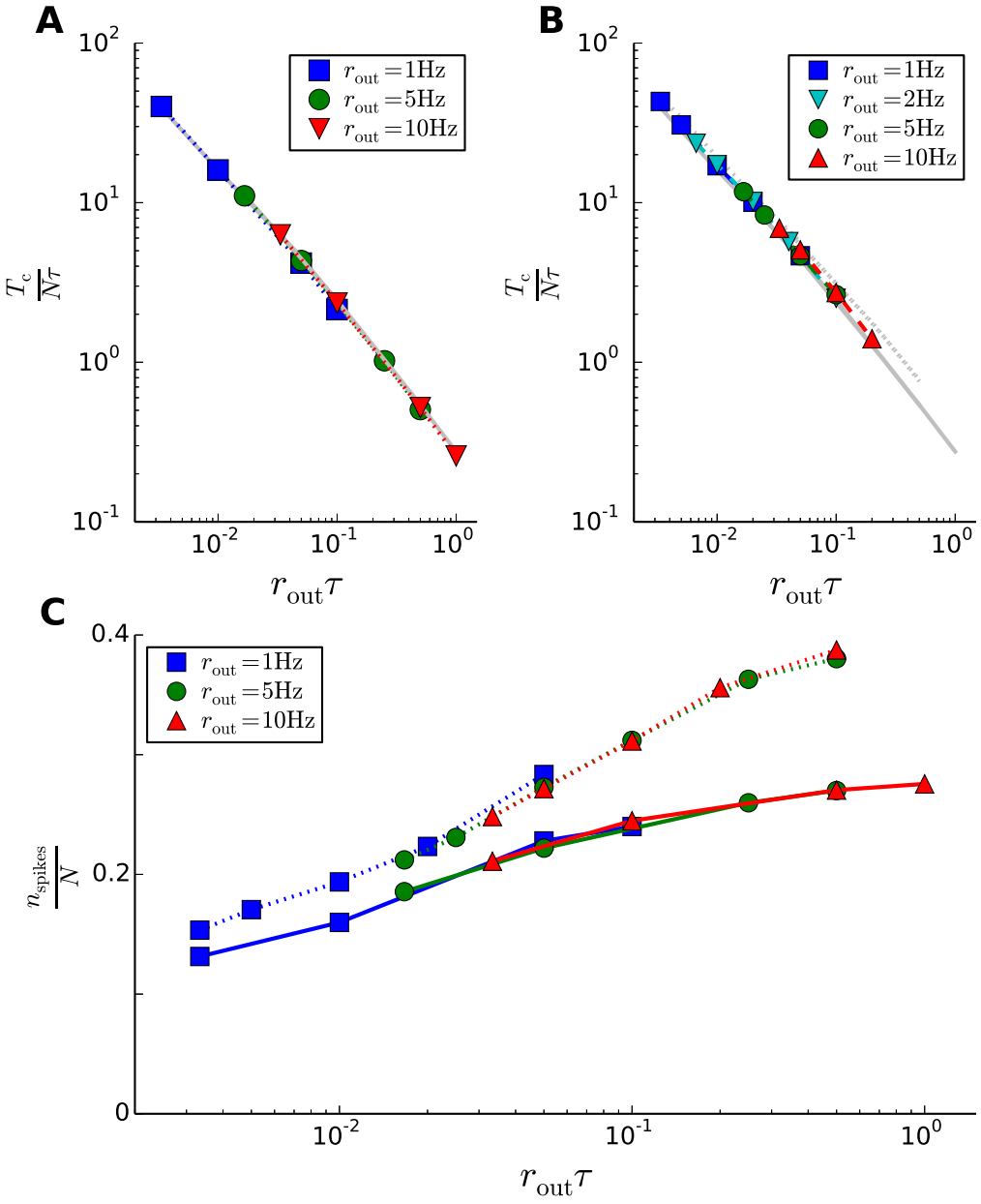

Figure S1: **Capacity (related to Figure 3).** In the top panels gray solid lines depict the results for HTP learning of a single long association (**Figure 3A**). (**A**) Capacity of LIF neurons for learning multiple input-output associations each of duration $T = 1\,\text{sec}$. A slight reduction in capacity is observed for $\tau/T = 0.1$. (**B**) Capacity of the Finite Precision algorithm to learn random input-output associations with $r_{\text{out}}\varepsilon = 10^{-3}$. Gray dotted lines depict the results for the Finite Precision algorithm with $r_{\text{out}}\varepsilon = 10^{-2}$ (**Figure 3A**). (**C**) Capacity presented as the average maximal number of desired output spikes the neuron can implement per synapse vs. $r_{\text{out}}\tau$. Solid lines depict the capacity obtained with the HTP method. Dashed lines depict the maximal number of desired output spikes learnable by the Finite Precision learning algorithm with $r_{\text{out}}\varepsilon = 10^{-2}$. The data presented here is identical to the data presented in **Figure 3A**.

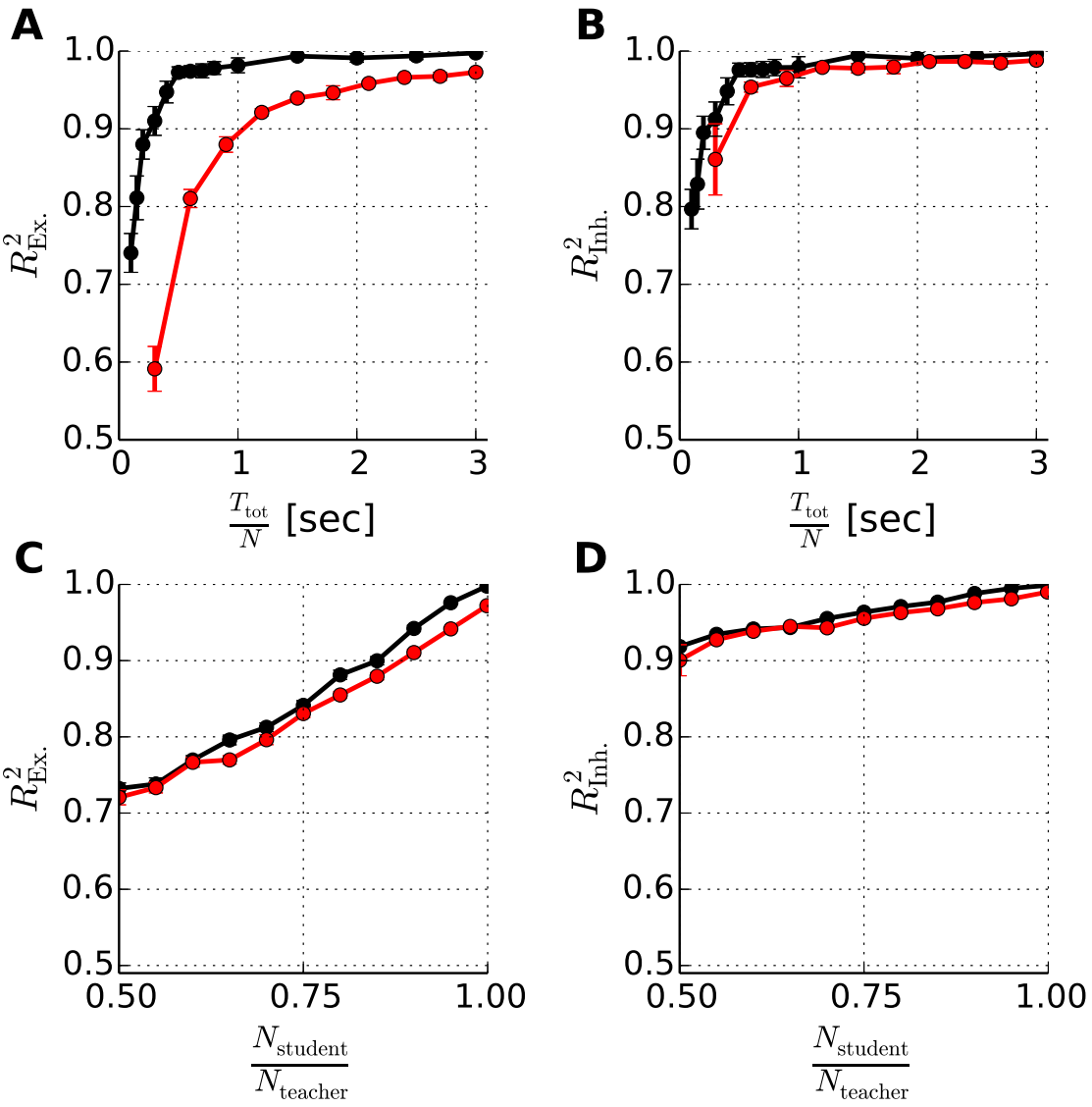

Figure S2: **Synaptic weights reconstruction (related to Figure 5).** R^2 values (experimental procedures) for the reconstructed weights as a function of the total duration of the training set (**A** - **B**) and as a function of the fraction of the number of synapses used during training (**C** - **D**) (see supplemental experimental procedures for details). (**A**) and (**C**) depict the results for the teacher's excitatory synapses. (**B**) and (**D**) depict the results for the teacher's inhibitory synapses. All panels depict the results for both weight reconstruction of an LIF teacher (black) and weight reconstruction of an HH teacher (red). Error bars depict two times standard error of the mean over 100 trials.

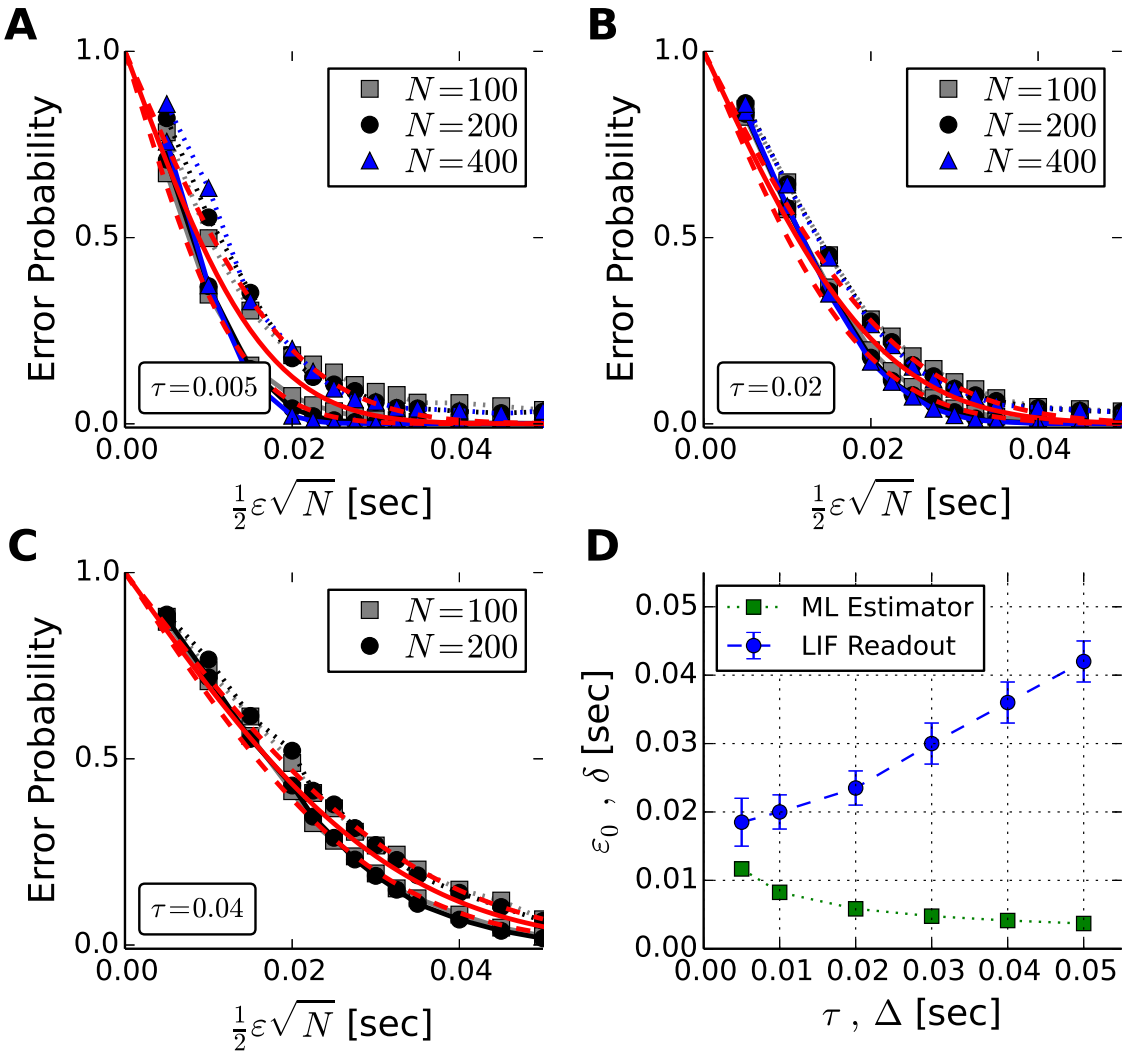

Figure S3: **Errors of precisely timed responses to RA activity (related to Figure 6).** (**A**), (**B**) and (**C**) depict error probability as a function of the scaled required precision for different τ (inset, in sec). The red lines depict a complementary error function scaled to fit the training and generalization error probabilities according to erfc $\left(\frac{\sqrt{N}\varepsilon}{2\varepsilon_0}\right)$. The solid line is taken to estimate ε_0 and the dashed lines yield the estimation error. (**D**) Estimated ε_0 as a function of the neuronal time constant τ (blue) compared to the scaled root mean square error, $\delta = \delta_0/\left(\bar{r}\Delta\right)^{\frac{1}{2}}$, of a maximum likelihood (ML) estimator as a function of the observed time window size, Δ (green). See Results and Supplemental Experimental Procedures for details.

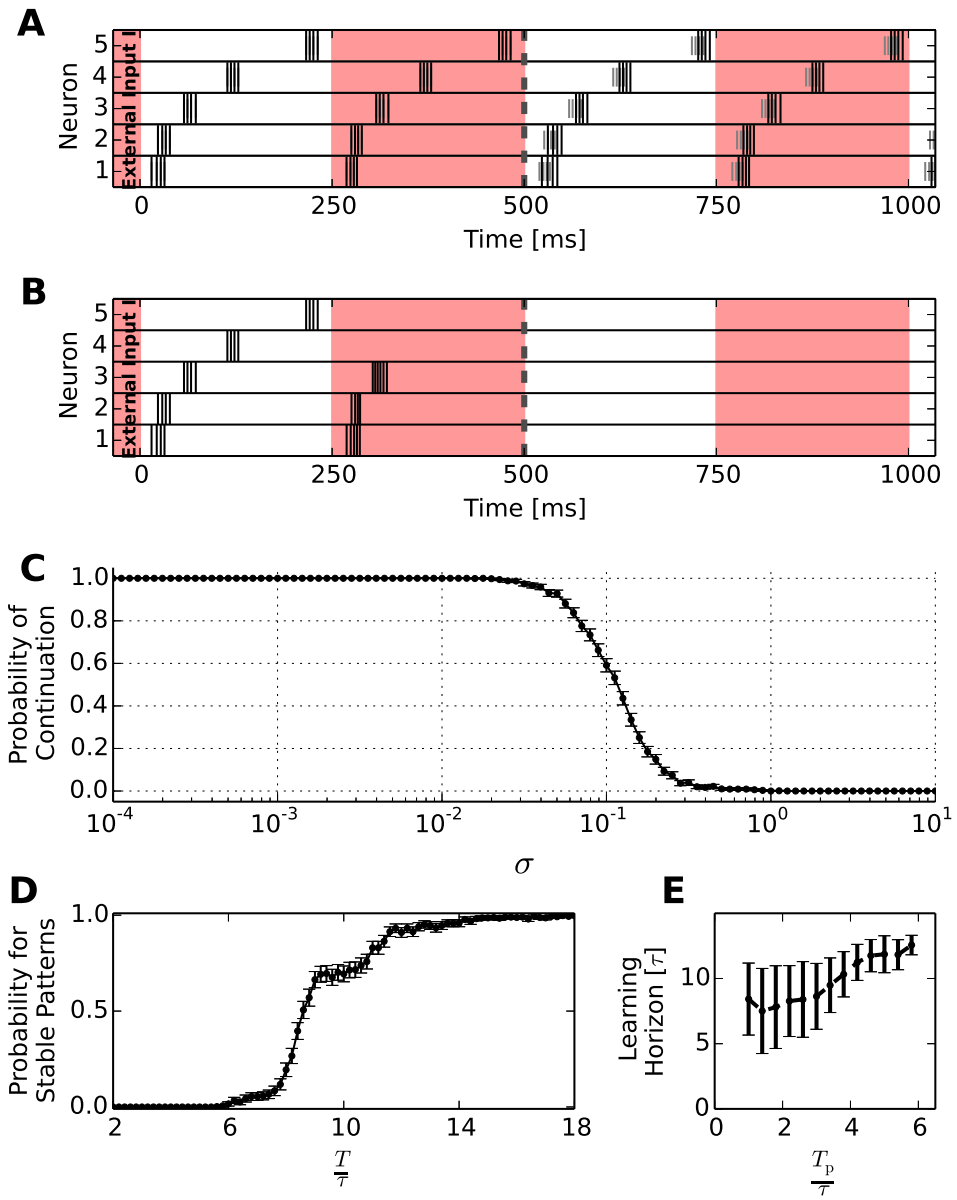

Figure S4: **Stability of the recurrent periodic pattern and estimation of the learning horizon (related to Figure 7B-C).** (**A**) Pattern (black spikes) continuing after a perturbation at $t = 0.5$s. The spiking dynamics quickly returns to its original pattern (unperturbed dynamics: gray background spikes) only shifted in time. (**B**) Network dynamics following the use of the HTP algorithm to train neurons to fire at the spike times learned by the FP algorithm with precision higher than 10^{-10}ms. Due to accumulation of numerical error, the pattern deviates from the learned orbit already before the termination of the trained pattern. (**C**) Probability of continuation after perturbation as in (A), versus perturbation size σ (in units of the neurons' threshold potential). Error-bars indicate two times standard error of the mean. (**D**) Probability of converging to a stable desired periodic activity pattern vs. the training sequence duration T. Error bars depict the standard error of the mean. (**E**) Estimation of the learning horizon vs. the period of the desired activity. See Results and Supplemental Experimental Procedures for details.

SUPPLEMENTAL EXPERIMENTAL PROCEDURES

Implementing Precise Spike Time Input-Output Associations

Robustness of solutions within the permitted subspace. Here we prove that within the permitted subspace, all solutions are robust, i.e., a small change in the solution weight vector $\boldsymbol{\omega}$ does not invalidate the solution. Constraints (2) ($\frac{\mathrm{d}U}{\mathrm{d}t}(t_\mathrm{d}) > 0$) imply that there is a finite margin $\delta > 0$ such that $\frac{\mathrm{d}U}{\mathrm{d}t}(t_\mathrm{d}) > \delta$. Constraints (3) ($U(t) < U_\mathrm{thr}$, for $t \neq t_\mathrm{d}$) do not appear to provide a finite margin since $U(t)$ becomes arbitrarily close to U_thr as $t \to t_\mathrm{d}$. However, due to the shape of the PSP, the voltage can have at most one maximum between successive input spikes. Denoting by t_last the last input spike before t_d, if $U(t_\mathrm{last}) < U_\mathrm{thr}$ and constraints (1) ($U(t_\mathrm{d}) = U_\mathrm{thr}$) and (2) are met, $U(t)$ cannot cross threshold in the interval $(t_\mathrm{last}, t_\mathrm{d})$. Thus, constraints (3) are equivalent to: (3') $U(t) < U_\mathrm{thr}$ for all times except in the intervals $(t_\mathrm{last}, t_\mathrm{d}]$. Condition (3') implies that there exists a finite margin $\delta' > 0$ such that $U(t) < U_\mathrm{thr} - \delta'$ in the relevant intervals. Thus, within the permitted subspace, a small enough change in any solution vector does not invalidate constraints (3') and (2).

HTP Algorithm. The algorithm is described in detail below (Algorithm 1).

Convergence proof for the HTP algorithm. As in the Perceptron, given a solution $\boldsymbol{\omega}^\star$, there exist $\delta > 0$ such that $\boldsymbol{\omega}^{\star T}\boldsymbol{x}(t) \leq -\delta$ for all t not in $(t_\mathrm{last}, t_\mathrm{d}]$ and $\boldsymbol{\omega}^{\star T}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t_\mathrm{d}) \geq \delta$ for all t_d. Using the facts that $\boldsymbol{\omega}^{\star T}\boldsymbol{x}(t) = \boldsymbol{\omega}^{\star T}P\boldsymbol{x}(t)$ and $\boldsymbol{\omega}^{\star T}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t_\mathrm{d}) = \boldsymbol{\omega}^{\star T}P\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t_\mathrm{d})$, we follow the convergence proof of the Perceptron (Rosenblatt, 1962; Minsky and Papert, 1988) and conclude that if $\boldsymbol{\omega}^\star$ exists, the total number of weight updates must remain finite. Since our algorithm stops if and only if no more errors occur, it must converge to zero error in finite time.

Modification of the threshold. In the HTP algorithm, the threshold U_thr changes during the course of the algorithm. The HTP algorithm can also be formulated with fixed threshold. As is the case for the Perceptron with fixed threshold (Brunel et. al.,

Algorithm 1 HTP algorithm.

1. **Initialization**

   (a) Initialize weights with $\boldsymbol{\omega} = P\boldsymbol{\omega}_0$ where $\boldsymbol{\omega}_0$ is an arbitrary weight vector.

   (b) Initialize the set of sub-sampled error patterns $\chi = \emptyset$

2. **Update of** χ

   (a) Present all the spike patterns to the neuron using the high threshold dynamics (Eq. 5 in Experimental Procedures)

   (b) Construct a set of error times, $\{t_{\mathrm{err}}\}$ consisting of (i) the time of maximal depolarization in every time segment in which the potential is suprathreshold not inside the intervals $(t_{\mathrm{last}}, t_{\mathrm{d}}]$; and (ii) every desired time at which the potential's slope is negative.

   (c) Stop if no errors where found, i.e. all equality and inequality constraints are satisfied.

   (d) Construct a set of labeled input patterns, $\{(\boldsymbol{\xi}, y)\}$, where $\boldsymbol{\xi} = P\boldsymbol{x}\left(t_{\mathrm{err}}\right)$, $y = -1$ if t_{err} is one of the maximal suprathreshold voltage times; $\boldsymbol{\xi} = P\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}\left(t_{\mathrm{err}}\right)$, $y = +1$ if $t_{\mathrm{err}} = t_{\mathrm{d}}$.

   (e) Add the constructed set to χ.

3. **Update of** ω

   (a) Present patterns in χ iteratively and update weights according to $\Delta\boldsymbol{\omega} = \eta y\boldsymbol{\xi}$ whenever sign $\left(\boldsymbol{\omega}^T\boldsymbol{\xi}\right) \neq y$ until all of them are correctly classified. $\eta > 0$ is a constant learning rate. Since all $\boldsymbol{\xi}$ are in the permitted subspace, the weight vector is kept within this subspace at all times.

4. Return to step 2.

2004; Clopath et. al., 2012), convergence to a solution is then guaranteed for sufficiently small η (not shown).

Capacity of LIF neurons

Analytical estimation of capacity. In Perceptron theory, capacity of random input-output associations is given by $\frac{P}{N} = \alpha_c^P\left(f\right)$ where N and P are the dimensionality of ω and the number of labeled patterns, respectively, and f is the fraction of $+1$ labeled

patterns. The function $\alpha_c^P(f)$ is given by

$$\alpha_c^P(f) = \frac{1}{f\int_x^\infty Dt\,(t-x)^2 + (1-f)\int_{-x}^\infty Dt\,(t+x)^2}\ , \tag{S1}$$

where $Dt = e^{-\frac{t^2}{2}}\frac{dt}{\sqrt{2\pi}}$ and x is given by the solution to:

$$f\int_x^\infty Dt\,(t-x) = (1-f)\int_{-x}^\infty Dt\,(t+x)\,. \tag{S2}$$

As explained in the Results, the equality constraints (1) diminish the effective dimensionality of our problem and since constraints (2) and (3) constitute two statistically distinct sets of inputs we can consider them as differently labeled sets that are required to be classified. Thus, in our case $N_{\text{eff}} = N - n_{\text{spikes}} \approx N - r_{\text{out}}T$, $P_{\text{eff}} \approx \frac{T}{\tau}(1 + r_{\text{out}}\tau)$, and $f_{\text{eff}} = \frac{n_{\text{spikes}}}{P_{\text{eff}}} \approx \frac{r_{\text{out}}\tau}{1+r_{\text{out}}\tau}$. Thus,

$$\frac{T}{N\tau} = \frac{\alpha_c^P(f_{\text{eff}})}{1 + r_{\text{out}}\tau + r_{\text{out}}\tau\alpha_c^P(f_{\text{eff}})}\ , \tag{S3}$$

which is depicted as the dashed line in **Figure 2D**. The behavior in the limit of small $r_{\text{out}}\tau$ is derived from the asymptotic behavior $\alpha_c^P \approx (-2f\log f)^{-1}$ as $f \to 0$.

Learning spike times with finite precision (FP)

Convergence of FP learning. In general, for spiking neural classifiers in which spike times are not precisely given, such as FP learning, the structure of the solution space is not necessarily convex and may even be comprised of many disjoint domains (Rubin et al., 2010). Thus, one cannot prove convergence by methods applied to linear separation problems, such as the Perceptron.

However, we can apply a Perceptron-like convergence proof in a subset of FP tasks. Given the inputs $\boldsymbol{x}(t)$ we define vectors $\boldsymbol{x}^{\pm}(t)$, which are the same as $\boldsymbol{x}(t)$ except in the reset term.

Specifically, $x_i^{\pm}(t) = x_i(t)$ for $i \leq N$, $x_{N+1}^{-}(t) = -1 - \left[\sum_{t_{\mathrm{d}'}-\varepsilon/2<t} u_{\mathrm{r}}(t - t_{\mathrm{d}'} + \varepsilon/2)\right]$, and $x_{N+1}^{+}(t) = -1 - \sum_{t_{\mathrm{d}'}+\varepsilon/2<t} u_{\mathrm{r}}(t - t_{d'} - \varepsilon/2)$. Thus, $\boldsymbol{x}^{-}(t)$ and $\boldsymbol{x}^{+}(t)$ represent the effect of the resets if output spike times were forced to be at their lowest and largest temporal boundaries, respectively. We now state the condition for convergence in the following theorem.

Theorem: The FP learning algorithm is guaranteed to converge to a weight vector $\boldsymbol{\omega}$ satisfying all the task's requirements within a finite number of iterations, if $\boldsymbol{x}^{-}(t)$ and $\boldsymbol{x}^{+}(t_{\mathrm{d}} + \varepsilon/2)$, are linearly separable, i.e., there exist $N+1$ dimensional $\boldsymbol{\omega}^{\star}$ and $\delta > 0$ that obey the linear inequalities (I) $\boldsymbol{\omega}^{\star T}\boldsymbol{x}^{-}(t) \leq -\delta$ for all t – this guarantees that no spikes occur outside the tolerance windows and that only one spike occurs within each of them – and (II) $\boldsymbol{\omega}^{\star T}\boldsymbol{x}^{+}(t_{\mathrm{d}} + \varepsilon/2) \geq \delta$ for all t_{d} which guarantees threshold crossing at each window.

Proof: Given a student LIF neuron with weight vector $\boldsymbol{\omega}$, we consider the change in $\boldsymbol{\omega}^{T}\boldsymbol{\omega}^{\star}$ due to a weight update during FP learning. If the error is an undesired spike, we have

$$\eta^{-1}\Delta\boldsymbol{\omega}^{T}\boldsymbol{\omega}^{\star} = -\boldsymbol{\omega}^{\star T}\boldsymbol{x}^{-}(t_{\mathrm{err}}) + \omega_{N+1}^{\star} \sum_{t_{\mathrm{d}}-\varepsilon/2<t_{\mathrm{err}}} \left[u_{\mathrm{r}}(t_{\mathrm{err}} - t_{\mathrm{d}} + \varepsilon/2) - u_{\mathrm{r}}(t_{\mathrm{err}} - t_{\mathrm{spike}})\right], \tag{S4}$$

where t_{spike} are the student's previous (correct) spike times within the tolerance windows around t_{d}. Since $\omega_{N+1}^{\star}$ (the teacher's threshold) is positive and $u_{\mathrm{r}}(t)$ is a monotonically decreasing function, the last term is non-negative, hence

$$\Delta\boldsymbol{\omega}^{T}\boldsymbol{\omega}^{\star} \geq -\eta\boldsymbol{\omega}^{\star T}\boldsymbol{x}^{-}(t_{\mathrm{err}}) > \eta\delta\ . \tag{S5}$$

Similarly, if the error is a missed spike, we have

$$\Delta\boldsymbol{\omega}^{T}\boldsymbol{\omega}^{\star} \geq \eta\boldsymbol{\omega}^{\star T}\boldsymbol{x}^{+}(t_{\mathrm{err}}) > \eta\delta\ . \tag{S6}$$

From Eqs. (S5),(S6) convergence in finite time follows as in the Perceptron convergence

proof.

Note that these conditions are sufficient for solving the FP task but they are not necessary conditions. It is easy to construct examples where a solution to the FP task exists while the above linear separability conditions for are not obeyed. How restrictive is this additional requirement? The following result can be proven: Given a set of desired times to be learned, there exists an $\varepsilon_{\max} > 0$ for which $\boldsymbol{x}^{-}(t)$ and $\boldsymbol{x}^{+}(t_{\mathrm{d}} + \varepsilon/2)$ are linearly separable for any $\varepsilon < \varepsilon_{\max}$ *iff* there exists a weight vector $\boldsymbol{\omega}^{\star}$ such that the membrane potential crosses the threshold only at the desired times, and its *subthreshold* potential, i.e., $V_{\mathrm{sub}}(t_{\mathrm{d}}) = \sum_{i=1}^{N} \omega_i^{\star} x_i(t_{\mathrm{d}})$ has a positive slope for all t_{d}. This last condition is more stringent than the condition that the slope of V is positive at t_{d}. Thus, the stability requirements are expected to hold for solutions of the FP task when $\varepsilon/\tau \ll 1$ and the desired output spike times are not bursty so that the neuron does not fire on the falling phase of the subthreshold potential. Finally, we note that in our simulations we have observed that the FP algorithm finds a solution to the task even in cases where the linear separability of $\boldsymbol{x}^{\pm}(t)$ is not obeyed. Thus, the algorithm has a general usefulness even beyond the conditions required for the proof of its convergence.

Learning of the threshold: Learning can also be implemented with constant U_{thr} (constant ω_{N+1}). Convergence as in the above theorem is then guaranteed for sufficiently small η (not shown).

Reconstruction of synaptic weights

Learning with unobserved input afferents. To simulate weight reconstruction of a neuron that the spiking activity of some of its input afferents is not observed, we activated the teacher neuron with $N_{\mathrm{teacher}} = 1000$ input afferents and trained the student neuron with only $N_{\mathrm{student}} < N_{\mathrm{teacher}}$ randomly chosen input afferents. **Figure S2C-D** depicts the reconstruction quality of the learned synaptic conductances as a function of $\frac{N_{\mathrm{student}}}{N_{\mathrm{teacher}}}$.

Reading out temporal information about ongoing motor output from recordings in songbird motor cortex

Estimation of ε_0 (Figure S3D). The single neuron imprecision, $\varepsilon_0(\tau)$, was estimated by first fitting a complementary error function, $\mathrm{erfc}\left(\frac{\sqrt{N}\varepsilon}{2\varepsilon_0}\right)$, to both the training and generalization errors separately, and then taking the mean of the two fitted parameters as the estimate of $\varepsilon_0(\tau)$ and the difference between them as the estimation error (see **Figure S3A-C**).

ML estimation of time (Figure S3D). The input to the ML estimator are the spike times of the synthetic RA neurons, within a time window of duration Δ, relative to the window's end time, t. To calculate the likelihood of observed spikes given time t, we assume that the spikes of each neuron are generated by independent non-homogeneous Poisson processes. For each neuron, the instantaneous firing rate is a piecewise constant function of time defined by its PSTH values in 3ms time bins. Estimation error was estimated for various population sizes ($N = 50 - 1600$) and temporal window sizes ($\Delta = 5 - 20$ms).

FP learning in recurrent networks

Stability of learned patterns. We verify the stability of the learned periodic orbits by continuing the recall beyond the trained duration, T, and by perturbing the network. The perturbation is implemented with an external input neuron that projects to the recurrent neurons with normally distributed weights (mean 0, standard deviation σ), and sends a single spike at a perturbation time randomly distributed within the first period after T. Continuation of the pattern is then checked by matching it after long times ($30T_\mathrm{p}$, where T_p is the period of the desired activity) to the desired pattern, allowing for phase shifts, and deviations within the tolerance ε. Probabilities and errors in **Figure S4C** are estimated over 1000 trials. The perturbation in **Figure S4A** was applied at time T with strength $\sigma = 0.1$.

Transition to stable learned periodic activity. Figure S4D displays the transition to infinite generation of a desired periodic pattern with period 3τ in a network of $N = 30$ neurons. Each neuron spikes once in a period. The neuron and learning parameters are $\tau_{\mathrm{m}} = 10\mathrm{ms}$, $\tau_{\mathrm{s}} = 2.5\mathrm{ms}$, ($\tau = 5\mathrm{ms}$), $\varepsilon = 1\mathrm{ms}$, and $\eta = 0.1$. **Figure S4E** shows the learning horizon for different values of the period T_{p}. The learning horizon is taken as the training period where 50% of the trials generate the desired dynamics beyond the training sequence, lower and upper ends of the error bars denote the training periods where this is the case for 40% and 60% of the trials. The total number of trials used per training period is 400. We observe similar results for $N = 50$ and $N = 100$ (data not shown).