The Chronotron: A Neuron that Learns to Fire Temporally Precise Spike Patterns

Răzvan V. Florian

Center for Cognitive and Neural Studies (Coneural),
Romanian Institute of Science and Technology,
Str. Cireşilor nr. 29, 400487 Cluj-Napoca, Romania

Email: florian@coneural.org

Abstract

In many cases, neurons process information carried by the precise timings of spikes. Here we show how neurons can learn to generate specific temporally precise output spikes in response to input spike patterns, thus processing and memorizing information that is fully temporally coded, both as input and as output. We introduce two new supervised learning rules for spiking neurons with temporal coding of information (chronotrons), one that is analytically derived and highly efficient and one that has a high degree of biological plausibility. For the latter rule, synaptic changes are proportional to the synaptic currents at the timings of actual and target spikes. We study these learning rules in computer simulations where we train integrate-and-fire neurons. Both learning rules allow neurons to fire spikes at the desired timings, with sub-millisecond precision. We show how chronotrons can learn to classify their inputs, by firing identical, temporally coded spike trains for different inputs belonging to the same class. When the input is noisy, the classification also leads to noise reduction. We also compute lower bounds for the memory capacity of chronotrons. The chronotrons can model neurons in oscillatory networks that encode information in the phases of spikes relative to the background oscillation.

There is increasing evidence that the precise timing of spikes, and not only the neural firing rate, represents information in the brain (Bohte, 2004; Tiesinga et al., 2008). For example, temporally structured multicell spiking patterns, organized into frames, were observed in hippocampus and cortex, and were associated to memory traces (Nádasdy et al., 1999; Ji and Wilson, 2007). In the olfactory bulb, spike latencies represent sensory input strength and identity (Margrie and Schaefer, 2003; Junek et al., 2010). The coding of information in the phases of spikes relative to a background oscillation has been observed in many brain regions, including the visual and prefrontal cortices and the hippocampus (Lee et al., 2005;

Jacobs et al., 2007; Fries et al., 2007; Montemurro et al., 2008; Siegel et al., 2009; Rutishauser et al., 2010).

Learning in neural networks that represent information through a firing rate code has been thoroughly studied (Hertz et al., 1991); however, we have lacked efficient, theory-supported learning rules for spiking neurons with temporal coding of information. The tempotron, a model of a spiking neuron endowed with a specific learning rule, has shown how a neuron can give a binary response to information encoded in the precise timings of the afferent spikes (Gütig and Sompolinsky, 2006; Florian, 2008; Urbanczik and Senn, 2009). But the tempotron's output represents information through the existence or the lack of an output spike during a predefined period. The timing of the tempotron's output spikes is arbitrary and does not carry information. Because of this change in the representation of information, the output of a tempotron cannot serve as the input of another tempotron. By contrast, the ReSuMe learning rule (Ponulak, 2005; Ponulak and Kasiński, 2010) allows supervised learning of spiking neural codes where the output is also temporally coded, but this rule is heuristically defined and, as we will show, less efficient than the analytically derived rule introduced here.

Here we present two new supervised learning rules for spiking neurons, which allow such neurons to process information that is encoded, for both input and output, in the precise timings of spikes. We show how single neurons can perform classification of input spike patterns into multiple categories, using a temporal coding of information with sub-millisecond precision. The first learning rule that we introduce here is analytically derived and highly efficient. The other learning rule is heuristic, but is more biologically plausible, because synaptic changes depend directly on the synaptic currents at the timings (actual and target) of the postsynaptic spikes.

We consider the problem of training a spiking neuron by changing its parameters, such that, for a given input, its output is as close as possible to some given target spike train. Multiple such associations must be performed with a single set of neural parameters. Information is represented in both the input and the output through the precise timings of spikes. We call a neuron that solves this problem a chronotron. In order to solve the chronotron problem, appropriate learning rules should be defined. Here we focus on learning rules that change the synaptic efficacies of the neuron, although other neural parameters can also be trained.

Materials and Methods

Neural model

Our analysis uses the Spike Response Model (SRM) of spiking neurons, which reproduces with high accuracy the dynamics of the complex Hodgkin-Huxley neural model while being amenable to analytical treatment (Gerstner and Kistler, 2002). The integrate-and-fire neuron is a particular case of the SRM. We consider that the arrival of the f-th presynaptic spike on the synapse j of a neuron at the moment t_j^f leads to a postsynaptic potential (PSP) whose value as a function of the time t is the product of synaptic efficacy w_j and a normalized kernel $\epsilon_j(t, t_j^f)$, i.e. $w_j\, \epsilon_j(t, t_j^f)$. We denote as λ_j the total normalized synaptic PSP, resulting from the contribution of past presynaptic spikes, $\lambda_j(t) = \sum_f \epsilon_j(t, t_j^f)$. The membrane potential u of the neuron is determined by the integration of the PSPs generated by presynaptic spikes, and also by a term η that models the refractoriness caused by the last spike fired by the studied neuron:

$$u(t) = \eta(t) + \sum_j w_j \lambda_j(t). \tag{1}$$

When the membrane potential reaches the firing threshold θ, a spike is fired and the membrane potential is reset.

Graphical illustration of the chronotron problem

We denote as n the chronotron's number of synapses. We consider the n-dimensional vectors $\boldsymbol{w}$, having as components the synaptic efficacies w_j, and, respectively, $\boldsymbol{\lambda}(t)$, having as components the normalized total synaptic PSPs $\lambda_j(t)$. After each postsynaptic spike, the normalized PSPs are reset to 0 and thus the trajectory of $\boldsymbol{\lambda}(t)$ in the n-dimensional space starts, after each of these spikes, from the origin of the space. Eq. 1 can be then rewritten in vectorial form as $u(t) = \eta(t) + \boldsymbol{w} \cdot \boldsymbol{\lambda}(t)$. The neuron fires a spike when $u(t) = \theta$, or $\boldsymbol{w} \cdot \boldsymbol{\lambda}(t) = \theta - \eta(t)$. The magnitude of the projection of the $\boldsymbol{\lambda}$ vector on the $\boldsymbol{w}$ vector is $(\boldsymbol{w} \cdot \boldsymbol{\lambda})/|\boldsymbol{w}|$. Thus, the neuron fires a spike when the magnitude of the projection of $\boldsymbol{\lambda}$ on $\boldsymbol{w}$ reaches $\left(\theta - \eta(t)\right)/|\boldsymbol{w}|$, i.e. $\boldsymbol{\lambda}$ reaches a spike-generating hyperplane which is perpendicular on the vector $\boldsymbol{w}$ and at a distance $\left(\theta - \eta(t)\right)/|\boldsymbol{w}|$ of the origin (Fig. 1). The chronotron problem can be understood as the problem of setting the spike-generating hyperplane, by changing $\mathbf{w}$, such that it intersects the trajectory of $\boldsymbol{\lambda}(t)$ at exactly those timings when we want spikes to be fired.

Analytical derivation of the E-learning rule

The Victor & Purpura (VP) metric (Victor and Purpura, 1996) defines the distance between two spike trains as the minimum cost required to transform one into the other. This is the sum of the costs assigned to either insertion of spikes, removal of spikes or shifting the timing of spikes. The cost of adding or deleting a single spike is set to 1, while the cost of shifting a spike by an amount Δt is $\sigma(|\Delta t|/\tau_q)$, where σ is a positive, increasing function with $\sigma(0) = 0$, and τ_q is a positive time constant that is a parameter of the metric.

For a given input, the trained neuron fires at the moments t^f, where f represents the index of the spike in the spike train. The ordered set of the spikes in the spike train fired by the neuron is $\mathscr{F} = \{t^1, t^2, ...\}$. The target spike train that the neuron should fire for that input is $\tilde{\mathscr{F}} = \{\tilde{t}^1, \tilde{t}^2, ...\}$. In a transformation of minimal cost, according to the VP metric, of the actual spike train $\mathscr{F}$ into the target one $\tilde{\mathscr{F}}$, the operations involved are the following: removal of spikes (that are not previously moved); insertion of spikes (at their target timings, so that they are not moved after insertion); and shifting of spikes toward their target timings. We denote as $\mathscr{F}^*$ the subset of $\mathscr{F}$ that represents the spikes that should be eliminated; and as $\tilde{\mathscr{F}}^*$ the subset of $\tilde{\mathscr{F}}$ that represents the timings of target spikes at which new spikes should be inserted into $\mathscr{F}$. The spikes in the actual spike train that are not eliminated, $\mathscr{F} - \mathscr{F}^*$, are in a one-to-one correspondence with the spikes in the target spike train for which a correspondent is not inserted, $\tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*$, and they should be moved towards their targets. The existing algorithm that computes the VP distance between two given spike trains (Victor and Purpura, 1996) can be extended

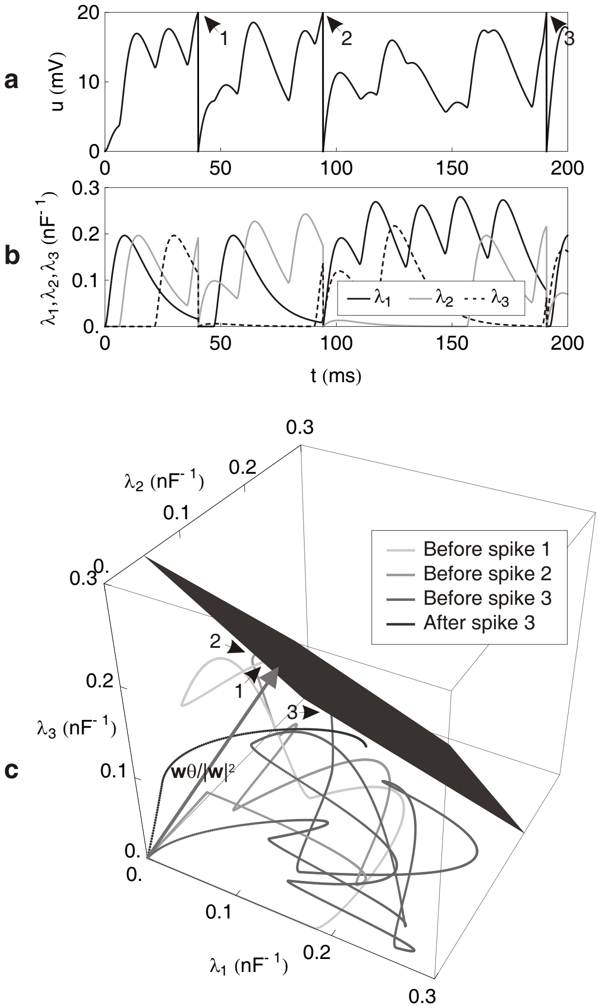

Figure 1: A graphical illustration of the chronotron problem for a neuron with $n = 3$ synapses and $\eta(t) = 0$. (a) The dynamics of the membrane potential u. The numbered arrows indicate the timings when the membrane potential reaches the firing threshold and spikes are fired. (b) The dynamics of the three components of $\boldsymbol{\lambda}$. (c) The trajectory of $\boldsymbol{\lambda}$. Spikes are generated when the trajectory reaches the spike-generating hyperplane, which is here the black plane, perpendicular on $\mathbf{w}/|\mathbf{w}|$ and at a distance $\theta/|\mathbf{w}|$ of the origin. The chronotron problem is solved by adjusting the location of the spike-generating hyperplane, through changes in $\mathbf{w}$, such that the timings of the fired spikes are the target ones. The numbered arrows indicate the generation of spikes at the timings when the spike-generating hyperplane is reached.

in order to also compute the sets $\mathscr{F}^*$, $\tilde{\mathscr{F}}^*$ and their complements (Supplementary Material).

E-learning aims to minimize the following error function:

$$E = \sum_{t^f \in \mathscr{F}^*} u(t^f) - \sum_{\tilde{t}^f \in \tilde{\mathscr{F}}^*} u(\tilde{t}^f) + \gamma_d \sum_{\substack{(t^f, \tilde{t}^g) \\ t^f \in \mathscr{F} - \mathscr{F}^* \\ \tilde{t}^g \in \tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*}} \sigma(|t^f - \tilde{t}^g| / \tau_q), \tag{2}$$

where γ_d is a positive parameter. Because the creation or deletion of spikes or changes in their classification in either $\mathscr{F}^*$ or $\mathscr{F} - \mathscr{F}^*$ lead to discontinuous changes of E, gradient descent can only be ensured piecewisely. The synaptic changes that aim to minimize the error function through piecewise gradient descent are $\Delta w_j \sim -\partial E / \partial w_j$. By performing the derivation with $\sigma(x) = x^2/2$ and after some approximations (Supplementary Material), we get the E-learning rule:

$$\Delta w_j = \gamma \Bigg[\sum_{\tilde{t}^f \in \tilde{\mathscr{F}}^*} \lambda_j(\tilde{t}^f) - \sum_{t^f \in \mathscr{F}^*} \lambda_j(t^f) + \frac{\gamma_r}{\tau_q^2} \sum_{\substack{(t^f, \tilde{t}^g) \\ t^f \in \mathscr{F} - \mathscr{F}^* \\ \tilde{t}^g \in \tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*}} (t^f - \tilde{t}^g)\, \lambda_j(t^f) \Bigg], \tag{3}$$

where γ is the learning rate, a positive parameter, and γ_r another positive parameter.

I-learning

The I-learning rule is defined by

$$\Delta w_j = \gamma \,\mathrm{sign}(w_j) \left[\sum_{\tilde{t}^f \in \tilde{\mathscr{F}}} I_j(\tilde{t}^f) - \sum_{t^f \in \mathscr{F}} I_j(t^f) \right], \tag{4}$$

where γ is the learning rate, a positive parameter, and I_j is the synaptic current on the synapse j (Supplementary Material).

Computer Simulations

The trained neuron was a classical leaky integrate-and-fire neuron, with synaptic currents modeled as a difference of two exponentials (Supplementary Material). The neuron had a membrane time constant $\tau_m = 10$ ms, a capacity $C = 2.5$ nF, a firing threshold $\theta = 20$ mV and a reset potential equal to the resting potential, $u_r = 0$. The time constants that define the dynamics of the synaptic currents were $\tau_s = 5$ ms and $\tau_r = 1.25$ ms. We used an event-driven simulation (D'Haene et al., 2009) where the timing of input spikes were represented with machine precision and the timing of the trained neuron's spikes were computed with a precision of 10^{-5} ms.

The neuron received inputs through n synapses. In Figs. 3 and 4 we used $n = 500$, while in Fig. 5 n was variable. At the beginning of learning experiments, synaptic efficacies were generated randomly, distributed uniformly between 0 and w_m. In Figs. 3 and 4 we used $w_m = 2{,}000/n$ pC, while in Fig. 5 we used $w_m =$

$1{,}000/n$ pC. The neuron was trained to learn p input patterns by firing a predetermined output spike train for each of the input patterns. The length T of the input patterns was of 200 ms. An input pattern consisted of the ensemble of the n input spike trains coming through the n synapses during the interval T. The input spike trains consisted, for each of the n synapses, of one spike generated at a random timing, distributed uniformly between 0 and T. During learning, the input patterns were presented sequentially, in batches consisting of the p patterns, and the synapses were modified at the end of the batches according to the learning rules. A presentation of one input pattern and the simulation of the output of the trained neuron corresponding to this input is called a trial. Each batch of presentations of the p patterns (trials) is called an epoch. At the beginning of each trial, the membrane potential of the neuron was reset to $0.8\ \theta$. The learning rates that we used were, for E-learning: $\gamma = 1{,}250/(n\ p)$ pC nF in Fig. 3, $\gamma = 2{,}500/(n\ p)$ pC nF in Fig. 4, and $\gamma = 5{,}000/(n\ p)$ pC nF in Fig. 5. We also used $\gamma_r = 15$ ms and $\tau_q = 10$ ms. For I-learning, the learning rates were $\gamma = 5/p$ ms in Figs. 3 and 4, and $\gamma = 20/p$ ms in Fig. 5.

In the experiments presented in Figs. 4 and 5, we trained the neuron to perform classifications by setting its target output to be the same for several different, randomly generated, inputs. The number c of the different outputs was the number of categories into which the neuron classified the p input patterns. In Fig. 4, $c = 1$ and in Fig. 5, $c = 3$. We assigned equal number of patterns into each category, such that p was an integer multiple of c. The output used a phase-coded representation of the information. The target spike train for each category $k \in \{1, \ldots, c\}$ consisted of one spike at $k\ T/(c+1)$. We considered that an input-output association was learned correctly by the trained neuron if the number of the actual output spikes was the one in the target spike train and each of the output spikes was fired within less than 1 ms of the target timing. We considered that the chronotron was able to learn correctly a particular setup if, during each of 500 realizations of the setup with different, random, initial conditions, all input-output mappings were learned correctly in no more than 10,000 epochs. For each realization of the experiments, both the input patterns and the initial synaptic efficacies were generated randomly. In Fig. 5, for various values of the number of inputs n we increased the load $\alpha = p/n$ until the chronotron was not able to learn correctly all the 500 random realizations of the setup. The capacity for a particular setup was the maximum load for which the chronotron was able to learn correctly that setup, lower than the first load for which the chronotron was not able to learn correctly the setup.

In Fig. 4, we considered that the actual spike matched the target one if there was exactly one actual spike and its timing was within τ_q of the target timing. The probability P_m that the fired spikes matched the target ones was the number of patterns within a trial for which the actual spikes matched the target one, divided by the total number of patterns, $p = 10$.

Methods are presented in more detail in the Supplementary Material.

Results

Understanding and Illustrating the Chronotron Problem

We investigated how a spiking neuron that uses temporal coding (a chronotron) can learn to change its parameters, such that, for a given input, its output is as close as possible to some given target spike train. The chronotron problem can be illustrated graphically by considering a space having the same number of dimensions n as the number of afferent synapses of the neuron. In this space, the n synaptic efficacies w_j define a vector $\mathbf{w}$ and the normalized synaptic PSPs λ_j define a vector $\boldsymbol{\lambda}$. The vector $\boldsymbol{\lambda}(t)$ moves around this space, in time, according to the dynamics of the PSPs, while $\mathbf{w}$ changes on much larger timescales than $\boldsymbol{\lambda}$. The neuron fires when $\boldsymbol{\lambda}(t)$ touches a hyperplane determined by $\mathbf{w}$ (Materials and Methods; Fig. 1). The chronotron problem can be understood as the problem of setting the spike-generating hyperplane, by changing $\mathbf{w}$, such that it intersects the trajectory of $\boldsymbol{\lambda}(t)$ at exactly those timings when we want spikes to be fired.

Similar optimization problems can usually be solved by defining an error function and then changing the parameters to be optimized, through methods like gradient descent, which minimize this error function. The differences between the actual spike train fired by the neuron for a particular input and, respectively, the target spike train can be measured using spike train metrics such as the Victor & Purpura (VP) distance (Victor and Purpura, 1996). The VP distance is defined as the minimum cost for transforming one spike train into the other by creating, removing or moving spikes (Victor and Purpura, 1996). However, one cannot derive an efficient learning rule using directly this distance, because the terms corresponding to spikes that should be created or removed are constant and do not reflect how creating or removing these spikes depends on the plastic parameters. In order to solve this issue, we used a new error function, which is a modification of the VP distance. Instead of constant cost terms for the independent spikes that have to be created or removed, our error function changes the VP distance by including terms that depend on the value of the membrane potential of the neuron at the timings of these spikes. This allows these terms to be differentiated piecewisely with respect to the plastic parameters (Materials and Methods; Supplementary Material).

E-learning

The resulting learning rule works by modifying each synaptic efficacy j by terms that depend on the total normalized PSP λ_j. For all output spikes that should be eliminated, each synaptic efficacy needs to be decreased with a term proportional to the value of λ_j at the moments of these spikes. For all target spikes that the neuron should fire, for which a spike should be created, each synaptic efficacy should be increased with a term proportional to λ_j at the moments of these target spikes. For all actual spikes that are close to their target positions and should be moved towards them, each synaptic efficacy needs to change with a term proportional to the value of λ_j at the moments of the actual spikes, multiplied by the temporal difference between actual and target spikes. We call this learning rule E-learning, because it results from the pursuit of minimization of an error function (Fig. 2; Materials and Methods; Supplementary Material).

Our learning rule aims to minimize the error function by performing piecewise gradient descent. The inherent discontinuities introduced in the error function by

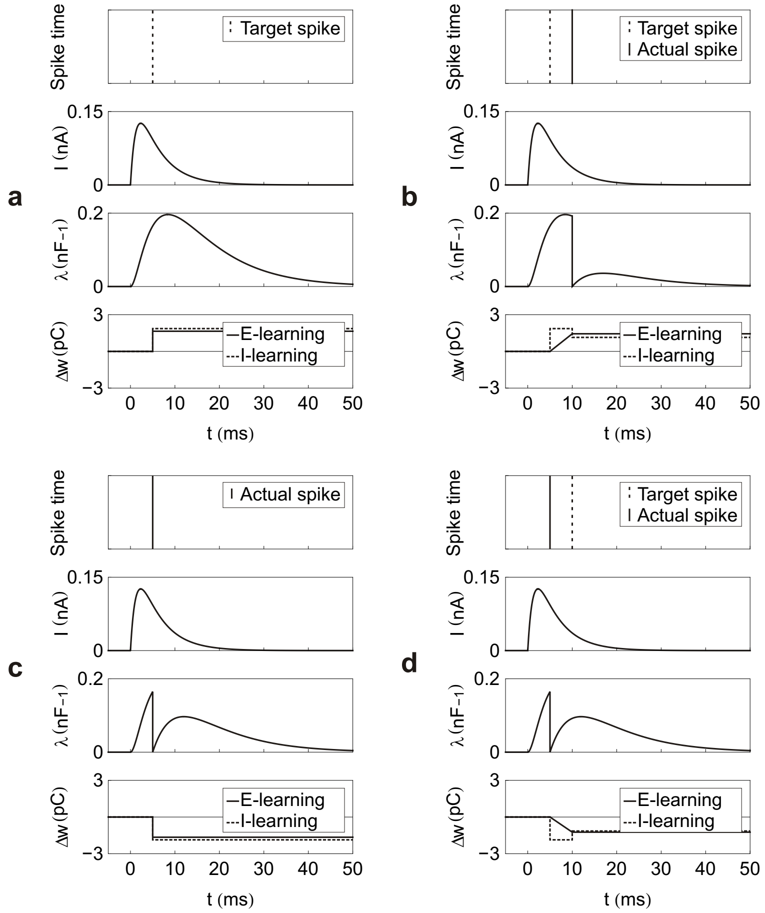

Figure 2: A graphical illustration of the plastic changes implied by the learning rules under several settings of actual and target spikes. The graphs show the spike timings and, for one synapse, the dynamics of the synaptic current I, the total normalized PSP λ and the synaptic changes Δw implied by the two learning rules. It is considered that one input spike arrives at this synapse at $t = 0$. The synaptic changes are shown to be localized temporally along the events that cause them; the actual application of the synaptic changes can be delayed with respect to these events. (a) One independent target spike and no actual spike. (b) A pair of matching target and actual spikes, the actual one following the target one. (c) One independent actual spike and no target spike. (d) A pair of matching target and actual spikes, the target one following the actual one.

creation or removal of spikes or by creation or breaking of matching pairs of actual and target spikes (Supplementary Material, Fig. S1) lead to the possibility that the error function is not minimized at each and every step during learning, since the discontinuous changes may lead to both increases or decreases of the error function. However, the terms that reflect in the error function spikes that should be created or removed ensure that the membrane potential is increased or, respectively decreased at the corresponding timings, such that the number of spikes becomes the desired one and the actual spikes are close to the target ones. Because the learning rule uses approximations (Supplementary Material), it is possible that gradient descent is not ensured, not even piecewisely. Since the learning rule does not ensure the minimization of the error function at each and every step, it can be considered a heuristic rule, although it is analytically derived. However, as the simulations have shown (see below), E-learning is much more efficient for chronotron training than the other available learning rules, having a much higher memory capacity. It is possible to devise a continuous error function that is then properly derivable, yielding a proper gradient descent. However, the continuous error function will be much more complex than the current one, yielding a complex learning rule. This will encumber an intuitive understanding of the learning rule, as it is possible with E-learning. A learning rule based on a continuous error function will be presented elsewhere.

I-learning

The second learning rule that we developed is heuristic and is inspired by both the E-learning rule and the existing ReSuMe learning rule (Ponulak, 2005; Ponulak and Kasiński, 2010) (Supplementary Material). As in ReSuMe, actual and target output spikes lead to synaptic changes of equal amplitude but of opposite signs, such that when the actual spike train corresponds to the target one the terms cancel out and synapses become stable. In ReSuMe, synaptic changes depend exponentially on the relative timings of pairs of pre- and postsynaptic spikes, as in some models of spike-timing-dependent plasticity. In contrast, here we consider that synaptic changes depend on the value of the synaptic current at the timings of spikes. This learning rule is thus biologically-plausible, since it depends on quantities that are directly available to the synapse. Target spikes determine synaptic potentiation, while actual spikes lead to synaptic depression. We call this synaptic current-dependent rule I-learning (Fig. 2; Materials and Methods; Supplementary Material).

Performance of the Learning Rules

We have studied these rules in computer simulations involving integrate-and-fire neurons having synaptic currents modeled as a difference of two exponentials. Both learning rules allow a neuron to perform accurate input-output mappings. Fig. 3 illustrates learning of a mapping between one input pattern (the spike trains coming through all input synapses) and one output spike train. The learning rules perform a descent in the landscape defined by the VP or E distance (Supplementary Material, Fig. S1).

We studied next setups where both input and output information were encoded temporally in the phases of spikes: both input and output spike trains consisted of one spike per trial (latency patterns), and the timing of this spike represented the information (phase coding). The length of spike patterns (and of one simulation trial) was 200 ms. This models experimentally-observed situations where phase locking of spikes relative to a theta rhythm is associated to encoding and memorizing of information (Margrie and Schaefer, 2003; Lee et al., 2005; Jacobs et al., 2007; Siegel et al., 2009; Rutishauser et al., 2010).

Fig. 4 illustrates learning of a mapping between 10 different input patterns and one output spike train consisting of one spike at the middle of the trial interval. For example, for E-learning, in 99.9% of the trials of 10,000 realizations, the neuron was able to fire the correct number of spikes (one spike) and the spike had an average timing difference of less than 0.03 ms with respect to the timing of the target spike, after about 8 minutes of learning (simulated time, 241 learning epochs). Learning worked even when the inputs were jittered, i.e. at each trial, input spikes were displaced around the reference timing according to a gaussian distribution. For example, in the same conditions as before but with an input jittered with a 5 ms amplitude, in more than 95% of trials, the neuron fired one spike with an average timing error of less than 2 ms, after about 8 minutes of learning (225 epochs). A 5 ms gaussian jitter amplitude corresponds to a 3.99 ms average timing displacement of the input spikes (Supplementary Material), so, in this case, the mapping also led to noise reduction, by doubling the precision of spike timing.

There are input-output mappings that are mechanistically impossible to be performed by a spiking neuron. For example, when there is no input, the neuron obviously cannot fire. When input spikes arrive uncorrelated on each of its synapses, the range of the outputs that the neuron is able to map to these inputs, by adjusting the synaptic efficacies, dramatically increases. But if the neuron has to perform several different input-output mappings with the same set of synaptic efficacies, the various mappings constrain each other through the synaptic efficacies. These constraints lead to the mechanistical impossibility that the neuron performs new input-output mappings beyond the current ones, and thus to a finite memory capacity of the neuron, which is discussed next.

Memory Capacity of the Chronotron

The chronotron is able to perform generic classification tasks, where p input patterns must be classified into c categories through hetero-association. In our simulations, equal number of patterns were randomly assigned to each category. The ability of neurons to memorize mappings corresponding to classification tasks increases with the number of synapses (number of input neurons) n. The ratio $\alpha = p/n$ (the number of input patterns memorized per synapse) represents the load imposed by the task on the neuron. A characteristic of the neuron's ability to learn is the maximum load for which the mappings are performed correctly (Gütig and Sompolinsky, 2006), which we call the capacity α_m of the neuron. We considered that the chronotron had a correct output when target spikes were reproduced with a 1 ms precision. Fig. 5 illustrates the performance of the chronotron in simulations where inputs were classified into $c = 3$ categories. For the studied setup, both I-learning and ReSuMe led to a capacity between 0.02 and 0.04, while E-learning led to a capacity $\alpha_m \approx 0.22$ patterns per synapse.

The load and the corresponding capacity has been used for characterizing neurons with binary outputs, which memorize one bit of information for every pattern. The chronotron can classify inputs in more than one category, and for c categories it memorizes $i = \log_2(c)$ bits of information for every input pattern. Therefore, a

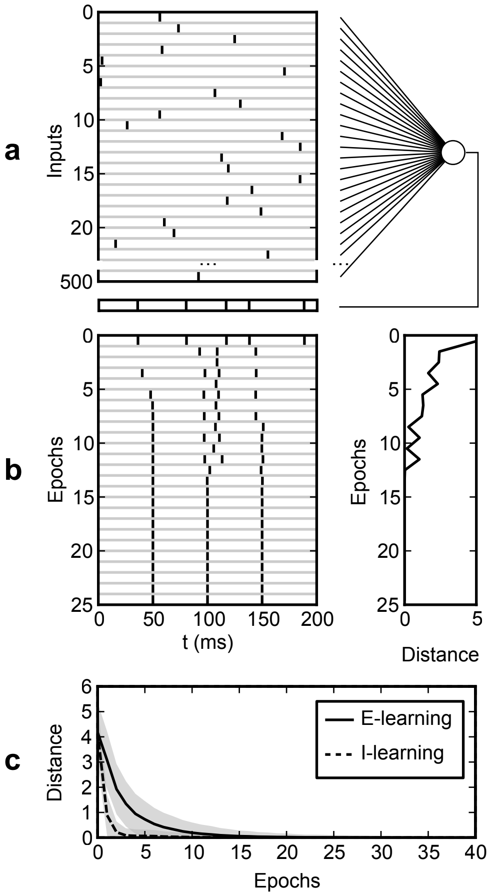

Figure 3: Learning of a mapping between one input pattern and one output spike train. The trained neuron receives inputs from 500 neurons. The spike trains received from these neurons form the input pattern. Each input spike train consists of one spike within the 200 ms of a trial, generated at a random timing having an uniform distribution within the trial. The target output spike train consists of spikes at 50, 100 and 150 ms. (a) A sample input pattern and the output spike train of the trained neuron, corresponding to this input, before learning. Only part of the 500 input spike trains are illustrated. (b) The synaptic efficacies change according to E-learning, such that the trained neuron's output reproduces the target spike train. Left: The output spike train during learning. Right: The VP distance between the actual and the target output spike train, during learning. The target output is reproduced after less than 15 epochs (presentations of the input pattern). (c) The VP distance between the actual and the target output spike train during learning, for E-learning and I-learning: averages and standard deviations over 10,000 realizations of the same experiment. Each realization uses different, random input spike trains and initial values of the synaptic efficacies.

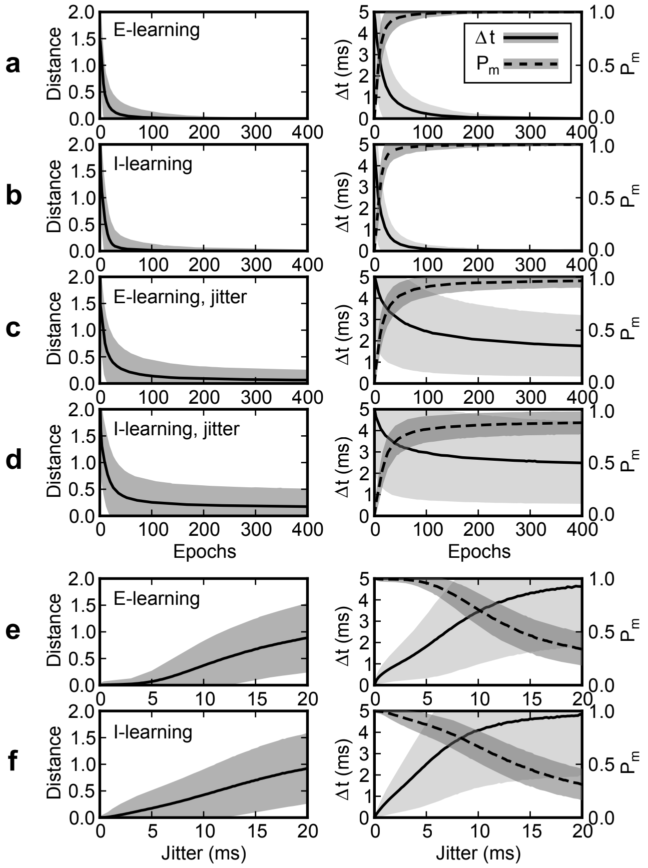

Figure 4: Learning of a mapping between 10 input patterns, with and without jitter, and one output spike train. Left: The VP distance between the actual and the target output spike train. Right: The timing difference Δt between matching spikes and the target spikes, and, respectively, the probability P_m that the fired spikes matched the target ones. The graphs represent averages and standard deviations over 10,000 realizations. (a), (c), (e): E-learning. (b), (d), (f): I-learning. (a)-(d): Evolution during learning, as a function of the learning epoch. (a), (b): No jitter. (c), (d): A gaussian jitter with an amplitude of 5 ms is added to each presentation of the input patterns. (e), (f): Values after 400 learning epochs, as a function of the amplitude of the input jitter. The inputs and the length of the trial are as in Fig. 3. The target output spike train consists of one spike at 100 ms.

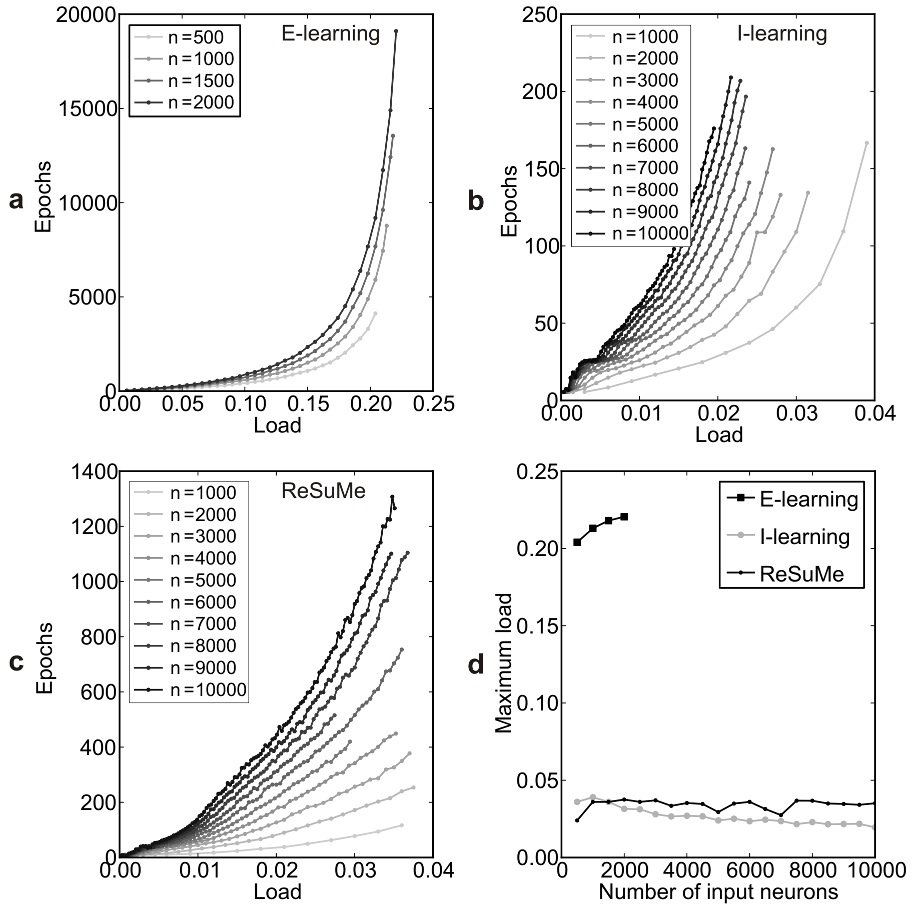

Figure 5: The performance of the chronotron learning rules for a classification problem with a phase coding of information. The input patterns are classified into 3 classes. (a)-(c) The average minimum number of epochs required for correct learning is displayed as a function of the load α, for various values of the number of synapses (input neurons) n. Note the scale differences. (a) E-Learning. (b) I-learning. (c) ReSuMe. (d) The maximum load for which correct learning can be achieved (the capacity), as a function of n. E-learning has a much better performance than I-learning or ReSuMe. For E-learning, simulations for higher n were not performed because of the high computational cost, due to the high capacity resulted through this learning rule. Averages were computed over 500 realizations with different, random initial conditions.

better measure for the chronotron's learning ability is the information load $\iota = \alpha\ i$ and the corresponding information capacity ι_m, equal to the maximum information load. The number of categories into which a chronotron classifies its inputs is limited only by the temporal precision of the output spikes. For example, if this temporal precision is 1 ms, with the setup presented here, the chronotron can encode up to about $c = 80$ categories (Supplementary Material). For phase coding of information, the simulations showed that the chronotron's capacity does not depend on the number of categories c (Supplementary Material, Fig. S2). The information capacity of the chronotron, for E-learning and the setup that we used, can be then computed as $\iota_m \approx 1.39$ bits per synapse (Supplementary Material). The information capacity of the perceptron is 2 and the one of the tempotron is around 3 (Gütig and Sompolinsky, 2006). However, if more than two input categories have to be discriminated, the chronotron has the advantage of being able to carry computations that need multiple perceptrons or tempotrons to be performed, being thus more efficient. Unlike the tempotron, the chronotron uses the same coding of information for both inputs and outputs and is therefore able to interact with other chronotrons.

Assuming that all neurons in the brain use the same encoding of information as the chronotron, for about 10^{11} neurons with about 10^4 synapses each, the total memory capacity of the brain is about 1.39 petabits. The capacities computed here are lower bounds, since it might be possible to develop learning rules which are more efficient than E-learning.

Optimal Information Representation

In our setups, information was represented in the precise timings of spikes relative to the beginning of trials of constant duration. If trials correspond to periods of a background oscillation, the timing of spikes corresponds to the phase relative to this oscillation. Simulations performed in this framework have shown that chronotrons have the best efficacy when both input and output consist of one spike per trial (period). Setups where inputs or outputs consisted of more than one spike, or where some of the inputs fired no spikes, skipping oscillation cycles, had suboptimal performance (Supplementary Material, Figs. S3–S5). Chronotrons' efficacy was not affected by initial conditions at the beginning of trials if output spikes were fired after a delay relative to the beginning of the trial of about 6 times the time constant of the membrane potential's exponential decay (Supplementary Material, Fig. S6). These results underline the role of oscillations for temporal information processing. First, they segment time into frames (periods), offering a reference for temporal encoding of information in spike phases. Second, in the parts of the cycles where neurons are globally inhibited or global excitation is low, oscillations ensure that neurons are reset such that they are able to process independently the inputs corresponding to different frames (periods). In the brain, when the spike phase encodes information relative to a background oscillation, the neurons fire no more than one spike per cycle in some, but not all, experiments (Nádasdy et al., 1999; Margrie and Schaefer, 2003; Lee et al., 2005; Jacobs et al., 2007; Rutishauser et al., 2010). Our results show that phase-of-firing coding with one spike per cycle is optimal for processing and memorization of temporally encoded information by spiking neurons. Cycle skipping is suboptimal, from an information processing perspective, but it might be motivated by energy expenditure constraints. For phase-of-firing coding, chronotrons have an optimal capacity if the oscillation period is about 8–10 times larger than the membrane time constant (Supplementary Material, Fig. S7).

Discussion

We have thus shown that, through appropriate learning methods, spiking neurons are able to process and memorize information that is encoded in the precise timing of spikes. By deriving analytically a spike-timing-based learning rule (E-learning) and then simulating its performance, we computed, at least for a particular setup, lower bounds of the memory capacity of a spiking neuron with temporal coding of information.

We have also developed a supervised learning rule, I-learning, which has a high degree of biological plausibility. It implies that synaptic changes are proportional to the corresponding synaptic currents. Postsynaptic spikes lead to synaptic depression similar to anti-Hebbian spike timing-dependent plasticity (STDP) (Bell et al., 1997; Egger et al., 1999; Han et al., 2000; Tzounopoulos et al., 2004; Fino et al., 2005; Letzkus et al., 2006; Sjöström and Häusser, 2006; Tzounopoulos et al., 2007; Fino et al., 2010), while the timings of target postsynaptic spikes trigger potentiation. The depression and potentiation should balance each other when actual spikes occur at the target timings. These target timings could be indicated by spikes coming from other, teacher neurons. The firing of these teacher neurons should lead to heterosynaptic associative changes according to the I-learning rule and should not have a significant impact on the trained neuron's potential. Such associative heterosynaptic plasticity that is similar to the potentiation component of I-learning has been observed in the brain (Humeau et al., 2003; Huang et al., 2004; Dudman et al., 2007; Izumi and Zorumski, 2008). For example, in the hippocampal CA1 pyramidal neurons, the stimulation of distal perforant path (PP) inputs induces long-term potentiation of the CA1 proximal Schaffer collateral (SC) synapses when the two inputs are paired at a precise interval (Dudman et al., 2007). This is similar to how the pairing of target spikes and of the synaptic inputs leads to potentiation in I-learning. The PP inputs are not plastic and generate much lower postsynaptic potentials than the SC ones, seeming to have an instructive role for the SC synapses, as target spikes have in I-learning, rather than an active role in driving the postsynaptic neuron. It remains to be explored whether the potentiation generated through such a mechanism can coexist with and be balanced by the anti-Hebbian STDP when the trained neuron reproduces the firing of the teacher neuron. In this case, the trained neuron's firing should then become increasingly correlated to the one of the teacher neuron, eventually mimicking its firing with a lag corresponding to the delay of the arrival of the teaching spikes. If the trained neuron learns from several teacher neurons, it should learn to fire when either one of the teacher neurons fires, acting thus as a kind of multiplexer.

If the trained neuron does not need to reproduce the entire activity of teaching neurons, but just the one during salient events, teaching could be modulated by a neuromodulator. Neuromodulation of supervised learning could be similar to the control of induction of associative plasticity in Purkinje cells through targeted modulation of instructive climbing fiber synapses (Carey and Regehr, 2009) or the neuromodulation of STDP (Seol et al., 2007; Couey et al., 2007; Pawlak and Kerr, 2008; Shen et al., 2008; Zhang et al., 2009; Pawlak et al., 2010).

Like STDP (Gerstner et al., 1996) or its neuromodulation (Florian, 2005, 2007) were predicted theoretically in advance of experimental verification, future experiments may find plasticity mechanisms similar to I-learning in the brain. For example, such mechanisms might be responsible for neural synchronization that modulates neural interactions (Womelsdorf et al., 2007), such as the synchronization of thalamic neurons needed for driving the cortex through weak synapses (Bruno and Sakmann, 2006); for encoding of information through synchronization (Singer, 1999); or for the fine temporal tuning of excitation relative to inhibition that contributes to stimulus selectivity in rat somatosensory cortex (Wilent and Contreras, 2005).

Besides the particular supervised learning rules introduced here, other learning mechanisms, such as reinforcement learning, or genetic constraints, could lead to the chronotron-like processing of temporally-coded information that has been demonstrated here.

Many applications of the presented learning rules are possible. For example, the supervised learning rules presented here could be used to train readout neurons of liquid state machines, for which perceptrons or spiking neurons with rate coding of outputs were previously used (Maass et al., 2002). Using spiking neurons with temporal coding as readouts for liquid state machines makes their information representation compatible to the one of spiking neurons in the liquid, thus allowing the outputs of the readouts to be fed back into the liquid. Such feedback significantly improves the computing power of liquid state machines (Maass et al., 2007), allowing the development of better models of information processing in the brain. Another possible application is the decoding of neural signals. Less efficient learning rules than the ones presented here have been already applied successfully, and with better results than alternative methods, to train simulated spiking neural networks to extract arm movement direction and hand orientation intent from the timing of spike trains recorded from monkeys (Fang et al., 2010). These are just a few examples of the potential uses of the learning rules presented here. These rules open the way to a plethora of future experiments that will explore how information encoded in the precise timing of spikes can be processed and memorized. This should lead to a better understanding of the information-processing features of neurons in the brain.

Funding

This work was supported by the Romanian National Authority for Scientific Research (PNCDI II, Parteneriate, grant number 11-039/2007); and by the Max Planck - Coneural Partner Group.

Acknowledgments

I thank to Raul C. Mureşan and Vlad V. Moca for comments on the manuscript.

References

Bell, C. C., Han, V. Z., Sugawara, Y. and Grant, K. (1997), ‘Synaptic plasticity in a cerebellum-like structure depends on temporal order’, *Nature* **387**(6630), 278–281.

Bohte, S. M. (2004), ‘The evidence for neural information processing with precise spike-times: A survey’, *Natural Computing* **3**(2), 195–206.

Bruno, R. M. and Sakmann, B. (2006), ‘Cortex is driven by weak but synchronously active thalamocortical synapses’, *Science* **312**, 1622–1627.

Carey, M. R. and Regehr, W. G. (2009), ‘Noradrenergic control of associative synaptic plasticity by selective modulation of instructive signals’, *Neuron* **62**, 112–122.

Couey, J. J., Meredith, R. M., Spijker, S., Poorthuis, R. B., Smit, A. B., Brussaard, A. B. and Mansvelder, H. D. (2007), ‘Distributed network actions by nicotine increase the threshold for spike-timing-dependent plasticity in prefrontal cortex’, *Neuron* **54**(1), 73–87.

D’Haene, M., Schrauwen, B., Van Campenhout, J. and Stroobandt, D. (2009), ‘Accelerating event-driven simulation of spiking neurons with multiple synaptic time constants’, *Neural Computation* **21**(4), 1068–1099.

Dudman, J. T., Tsay, D. and Siegelbaum, S. A. (2007), ‘A role for synaptic inputs at distal dendrites: Instructive signals for hippocampal long-term plasticity’, *Neuron* **56**(5), 866–879.

Egger, V., Feldmeyer, D. and Sakmann, B. (1999), ‘Coincidence detection and changes of synaptic efficacy in spiny stellate neurons in rat barrel cortex’, *Nature Neuroscience* **2**(12), 1098–1105.

Fang, H., Wang, Y. and He, J. (2010), ‘Spiking neural networks for cortical neuronal spike train decoding’, *Neural Computation* **22**(4), 1060–1085.

Fino, E., Glowinski, J. and Venance, L. (2005), ‘Bidirectional activity-dependent plasticity at corticostriatal synapses’, *The Journal of Neuroscience* **25**(49), 11279–11287.

Fino, E., Paille, V., Cui, Y., Morera-Herreras, T., Deniau, J.-M. and Venance, L. (2010), ‘Distinct coincidence detectors govern the corticostriatal spike timing-dependent plasticity’, *The Journal of Physiology* **588**(16), 3045–3062.

Florian, R. V. (2005), A reinforcement learning algorithm for spiking neural networks, *in* D. Zaharie, D. Petcu, V. Negru, T. Jebelean, G. Ciobanu, A. Cicortaş, A. Abraham and M. Paprzycki, eds, ‘Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2005)’, IEEE Computer Society, Los Alamitos, CA, USA, pp. 299–306.

Florian, R. V. (2007), ‘Reinforcement learning through modulation of spike-timing dependent plasticity’, *Neural Computation* **19**(6), 1468–1502.

Florian, R. V. (2008), Tempotron-like learning with ReSuMe, *in* V. Kurkova, R. Neruda and J. Koutnik, eds, ‘Artificial Neural Networks - Proceedings of the 18th International Conference (ICANN 2008), Prague, Czech Republic’, Vol. 5164 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 368–375.

Fries, P., Nikolić, D. and Singer, W. (2007), ‘The gamma cycle’, *Trends in Neurosciences* **30**(7), 309 – 316. July INMED/TINS special issue–Physiogenic and pathogenic oscillations: the beauty and the beast.

Gerstner, W., Kempter, R., van Hemmen, J. and Wagner, H. (1996), 'A neuronal learning rule for sub-millisecond temporal coding', *Nature* **383**, 76–78.

Gerstner, W. and Kistler, W. M. (2002), *Spiking neuron models*, Cambridge University Press, Cambridge, UK.

Gütig, R. and Sompolinsky, H. (2006), 'The tempotron: a neuron that learns spike timing-based decisions', *Nature Neuroscience* **9**(3), 420–428.

Han, V. Z., Grant, K. and Bell, C. C. (2000), 'Reversible associative depression and nonassociative potentiation at a parallel fiber synapse', *Neuron* **27**, 611–622.

Hertz, J., Krogh, A. and Palmer, R. G. (1991), *Introduction to the theory of neural computation*, Perseus Books, Cambridge, MA.

Huang, Y.-Y., Pittenger, C. and Kandel, E. R. (2004), 'A form of long-lasting, learning-related synaptic plasticity in the hippocampus induced by heterosynaptic low-frequency pairing', *Proceedings of the National Academy of Science of the USA* **101**(3), 859–864.

Humeau, Y., Shaban, H., Bissière, S. and Lüthi, A. (2003), 'Presynaptic induction of heterosynaptic associative plasticity in the mammalian brain', *Nature* **426**, 841–845.

Izumi, Y. and Zorumski, C. F. (2008), 'Direct cortical inputs erase long-term potentiation at schaffer collateral synapses', *The Journal of Neuroscience* **28**(38), 9557–9563.

Jacobs, J., Kahana, M. J., Ekstrom, A. D. and Fried, I. (2007), 'Brain oscillations control timing of single-neuron activity in humans', *Journal of Neuroscience* **27**(14), 3839–3844.

Ji, D. and Wilson, M. A. (2007), 'Coordinated memory replay in the visual cortex and hippocampus during sleep', *Nature Neuroscience* **10**(1), 100–107.

Junek, S., Kludt, E., Wolf, F. and Schild, D. (2010), 'Olfactory coding with patterns of response latencies', *Neuron* **67**, 872–884.

Lee, H., Simpson, G., Logothetis, N. and Rainer, G. (2005), 'Phase locking of single neuron activity to theta oscillations during working memory in monkey extrastriate visual cortex', *Neuron* **45**(1), 147–156.

Letzkus, J. J., Kampa, B. M. and Stuart, G. J. (2006), 'Learning rules for spike timing-dependent plasticity depend on dendritic synapse location', *Journal of Neuroscience* **26**(41), 10420–10429.

Maass, W., Joshi, P. and Sontag, E. D. (2007), 'Computational aspects of feedback in neural circuits', *PLoS Computational Biology* **3**(1), e165.

Maass, W., Natschlaeger, T. and Markram, H. (2002), 'Real-time computing without stable states: A new framework for neural computation based on perturbations', *Neural Computation* **14**, 2531–2560.

Margrie, T. W. and Schaefer, A. T. (2003), 'Theta oscillation coupled spike latencies yield computational vigour in a mammalian sensory system', *Journal of Physiology* **546**(2), 363–374.

Montemurro, M. A., Rasch, M. J., Murayama, Y., Logothetis, N. K. and Panzeri, S. (2008), ‘Phase-of-firing coding of natural visual stimuli in primary visual cortex’, *Current Biology* **18**(5), 375–380.

Nádasdy, Z., Hirase, H., Czurkó, A., Csicsvari, J. and Buzsáki, G. (1999), ‘Replay and time compression of recurring spike sequences in the hippocampus’, *Journal of Neuroscience* **19**(21), 9497–9507.

Pawlak, V. and Kerr, J. N. D. (2008), ‘Dopamine receptor activation is required for corticostriatal spike-timing-dependent plasticity’, *Journal of Neuroscience* **28**(10), 2435–2446.

Pawlak, V., Wickens, J. R., Kirkwood, A. and Kerr, J. N. D. (2010), ‘Timing is not everything: Neuromodulation opens the STDP gate’, *Frontiers in Synaptic Neuroscience* **2**, 146.

Ponulak, F. (2005), ReSuMe — new supervised learning method for the spiking neural networks, Technical report, Institute of Control and Information Engineering, Poznań University of Technology, Poland.

Ponulak, F. and Kasiński, A. (2010), ‘Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting’, *Neural Computation* **22**(2), 467–510.

Rutishauser, U., Ross, I., Mamelak, A. and Schuman, E. (2010), ‘Human memory strength is predicted by theta-frequency phase-locking of single neurons’, *Nature* **464**(7290), 903–907.

Seol, G. H., Ziburkus, J., Huang, S., Song, L., Kim, I. T., Takamiya, K., Huganir, R. L., Lee, H.-K. and Kirkwood, A. (2007), ‘Neuromodulators control the polarity of spike-timing-dependent synaptic plasticity’, *Neuron* **55**, 919–929.

Shen, W., Flajolet, M., Greengard, P. and Surmeier, D. J. (2008), ‘Dichotomous dopaminergic control of striatal synaptic plasticity’, *Science* **321**(5890), 848–851.

Siegel, M., Warden, M. and Miller, E. (2009), ‘Phase-dependent neuronal coding of objects in short-term memory’, *Proceedings of the National Academy of Sciences* **106**(50), 21341–21346.

Singer, W. (1999), ‘Neuronal synchrony: A versatile code for the definition of relations?’, *Neuron* **24**, 49–65.

Sjöström, P. J. J. and Häusser, M. (2006), ‘A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons’, *Neuron* **51**(2), 227–238.

Tiesinga, P., Fellous, J. and Sejnowski, T. (2008), ‘Regulation of spike timing in visual cortical circuits’, *Nature Reviews Neuroscience* **9**(2), 97–107.

Tzounopoulos, T., Kim, Y., Oertel, D. and Trussell, L. O. (2004), ‘Cell-specific, spike timing-dependent plasticities in the dorsal cochlear nucleus’, *Nature Neuroscience* **7**(7), 719–725.

Tzounopoulos, T., Rubio, M. E., Keen, J. E. and Trussell, L. O. (2007), 'Coactivation of pre- and postsynaptic signaling mechanisms determines cell-specific spike-timing-dependent plasticity', *Neuron* **54**(2), 291–301.

Urbanczik, R. and Senn, W. (2009), 'A gradient learning rule for the tempotron', *Neural Computation* **21**(2), 340–352.

Victor, J. D. and Purpura, K. P. (1996), 'Nature and precision of temporal coding in the visual cortex: A metric-space analysis', *Journal of Neurophysiology* **76**(6), 1310–1326.

Wilent, W. B. and Contreras, D. (2005), 'Dynamics of excitation and inhibition underlying stimulus selectivity in rat somatosensory cortex', *Nature Neuroscience* **8**(10), 1364–1370.

Womelsdorf, T., Schoffelen, J.-M., Oostenveld, R., Singer, W., Desimone, R., Engel, A. K. and Fries, P. (2007), 'Modulation of neuronal interactions through neuronal synchronization', *Science* **316**, 1609–1612.

Zhang, J. C., Lau, P. M. and Bi, G. Q. (2009), 'Gain in sensitivity and loss in temporal contrast of STDP by dopaminergic modulation at hippocampal synapses', *Proceedings of the National Academy of Science of the USA* **106**(31), 13028–13033.

The chronotron: A neuron that learns to fire temporally-precise spike patterns
Supplementary material

Răzvan V. Florian

Center for Cognitive and Neural Studies (Coneural),
Romanian Institute of Science and Technology,
Str. Cireşilor nr. 29, 400487 Cluj-Napoca, Romania
florian@coneural.org

Contents

1 Supplementary results

1.1 The error landscape

Fig. S1 (a)-(b) illustrates the discontinuities of the Victor & Purpura (VP) distance (Victor and Purpura, 1996) and, respectively, of our E distance between the actual and the target spike train, as a function of the synaptic efficacies, for a neuron with two synapses. Fig. S1 (c)-(e) illustrates how the learning rules perform a descent in the landscape defined by the VP or E distance.

1.2 Capacity does not depend on the number of categories

Fig. S2 shows that, for phase coding of information, the chronotron's capacity does not depend on the number of categories. Except the number of categories, the setup was as in Fig. 5, with $n = 500$.

1.3 Sensitivity of learning performance to various parameters of the studied setup

In Fig. S3, we explored the chronotron performance as a function of the number of output spikes per trial, for a setup where the chronotron had to fire the same output for all inputs. The setup was as in Fig. 5, with E-learning and $n = 500$, except that $c = 1$, and that the output consisted of o output spikes, placed at $k\,T/(o+1)$, for $k \in \{1,\ldots,o\}$. Best performance was achieved for a single output spike per trial.

In Fig. S4, we explored the chronotron performance as a function of the firing rate of the inputs. Here, the input spike trains were generated using a Gamma process of order 3 and time constant τ_Γ, i.e. the interspike intervals were generated randomly with a probability distribution, for an interspike interval t,

$$P(t) = \frac{t^2}{\tau_\Gamma^3}\frac{\exp(-t/\tau_\Gamma)}{2}. \tag{1}$$

This leads to input spike trains having an average firing rate of $1/\tau_\Gamma$ and average interspike interval (period) τ_Γ. The learning rate was adapted to the input firing rate, $\gamma = 5{,}000(\tau_\Gamma/T)/(n\,p)$ pC nF. Except the input and the learning rate, the setup was as in Fig. 5, with E-learning and $n = 500$. We studied the performance as a function of the normalized average period $\phi = \tau_\Gamma/T$. Best capacity was achieved for values of ϕ around 1, i.e. a single input spike per trial, for each synapse, on average, while fastest learning was achieved for ϕ around 0.5. The probability distribution of the number of input spikes per trial, for these values of ϕ, is illustrated in Fig. S11.

In Fig. S5, we explored the chronotron performance as a function of the probability P_s that input neurons skip cycles. Each input spike train consisted of either one spike generated at a random timing, distributed uniformly between 0 and T, as before, or, with a probability P_s, of no spikes. This models the cycle skipping of neurons in oscillatory networks. The learning rate was adapted as $\gamma = 5{,}000/(n\,p)/(1-P_s)$ pC nF. Except the input and the learning rate, the setup was as in Fig. 5, with E-learning and $n = 500$. Best capacity was achieved for values of P_s less or equal to 0.1, while fastest learning was achieved for no cycle skipping.

In Fig. S6, we explored the chronotron performance as a function of the timing of the output spike and of the initial state of the membrane potential. The neuron had to learn to have the same output for all inputs. The output was one spike at a given timing ψ relative to the beginning of the trial. At the beginning of each trial, the membrane potential u was either set to $0.8\,\theta$, as in the other experiments (stable initial state), or was generated randomly, with a uniform distribution, between 0 and $0.8\,\theta$ (random initial state). Except the target output, the setup was as in Fig. 5, with E-learning, $n = 500$, and $c = 1$. For this setup, the capacity and the learning time for reaching the correct output, for stable initial state, does not depend on ψ if it is larger than about 40 ms. Because of the exponential decay of the membrane potential of the chronotron with a time constant τ_m, the effect of the random initial state of the membrane potential on the chronotron's performance, as a function of the output spike timing ψ, becomes insignificant at $\psi \approx 6\,\tau_m$, similarly to $\exp(-\psi/\tau_m)$, as $\exp(-6) \approx 0.002$.

In Fig. S7, we explored the chronotron performance as a function of the trial length T. Except the trial length, the setup was as in Fig. 5, with E-learning and $n = 500$. Best performance was achieved for $T = 80 \ldots 100$ ms ($T/\tau_m = 8 \ldots 10$). Since the setup is invariant to a change of the time scale, the relevant parameters are the relative time scales T/τ_m, T/τ_s, T/τ_r, T/τ_q.

In Fig. S8, we explored the chronotron performance as a function of the reset potential u_r. Except the reset potential, the setup was as in Fig. 5, with E-learning and $n = 500$. The performance does not depend on the reset potential if it is lower than half of the firing threshold, $\theta/2 = 10$ mV.

In Fig. S9, we used parameters optimized for fast learning for a setup with a relatively low load, $p = 20$, $n = 1{,}000$ ($\alpha = 0.02$), with $c = 5$. Parameters were optimized to lead to the minimum average number of learning epochs needed for correct learning for this setup. Averages were computed over 500 realizations with random initial conditions and inputs. Inputs and outputs were phase-coded, as in Fig. 5. The parameters that resulted from the optimization were: for E-learning, $\gamma = 11{,}800/(n\ p)$ pC nF, $\gamma_r = 87.5$ ms, $\tau_q = 23.5$ ms; for I-learning, $\gamma = 30.5/p$ ms; for ReSuMe, $\gamma = 96{,}000/(n\ p)$ pC, $\tau_R = 10$ ms, $a_R = 0$. For the setup that was optimized and for the optimal parameters, ReSuMe had the fastest learning (16.752 ± 7.427 epochs), followed by I-learning (23.388 ± 6.866 epochs) and E-learning (36.480 ± 7.615 epochs). However, the advantages of the first two learning rules over E-learning disappeared for setups with higher loads or higher number of inputs than the optimized setup, when the other parameters were kept the same.

1.4 Synaptic distributions resulted from learning

Fig. S10 presents the distribution of the synaptic efficacies, before and after learning, for the experiments presented in Fig. 4.

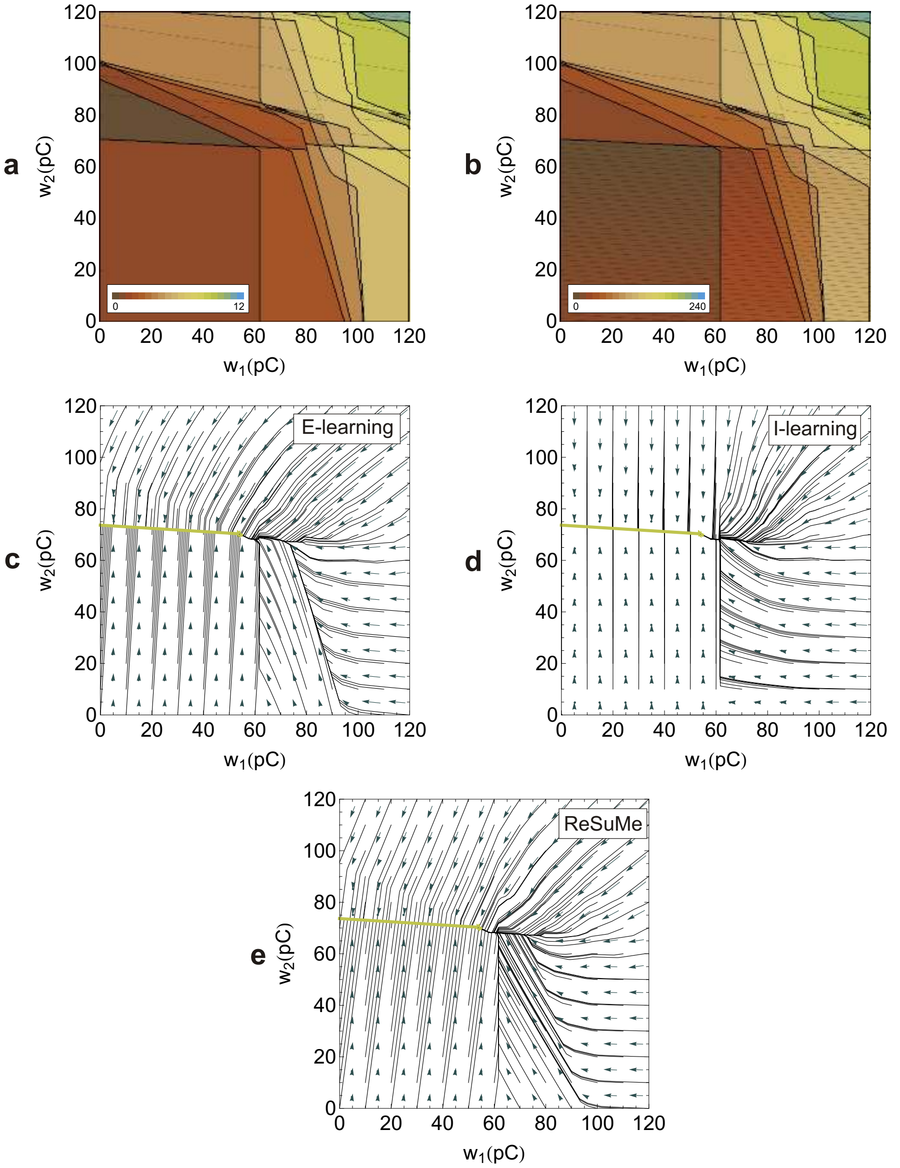

Figure S1: The error landscape for a neuron with two synapses and the descent on this landscape during learning. The neuron receives several input spikes on each synapse and has to fire one spike at a predefined target timing. (a) A contour plot of the VP distance between the actual spike train and the target spike train as a function of the values of the synaptic efficacies. The thick lines correspond to discontinuities of the distance. (b) A contour plot of the E distance. (c), (d), (e) The dynamics of the synaptic efficacies according to the learning rules. The black lines represent actual trajectories of the synaptic efficacies. The vectors represent synaptic changes. The green line corresponds to the values of the synaptic efficacies for which the output corresponds to the target spike train. (c) E-learning. (d) I-learning. (e) ReSuMe.

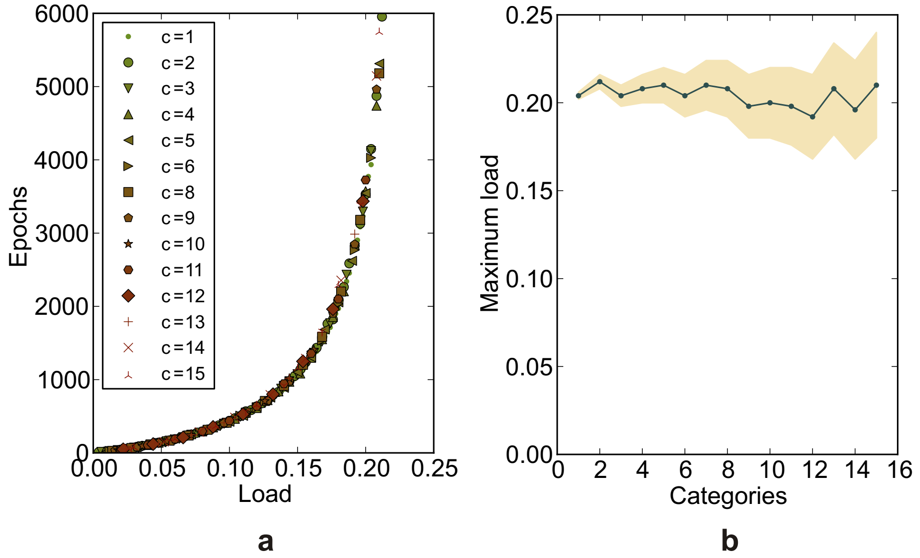

Figure S2: The dependence on the number of categories c of the performance of E-learning for a classification problem with a phase coding of information. (a) The average minimum number of epochs required for correct learning, as a function of the load α, for various numbers of categories c. Regardless of c, the points fall on the same curve. b) The maximum load for which correct learning is achieved (the capacity), as a function of the number of categories c. The shaded area represents the uncertainty due to the fact that the load can vary only discretely, in steps of c/n, for a particular c. The capacity is approximately constant for all c.

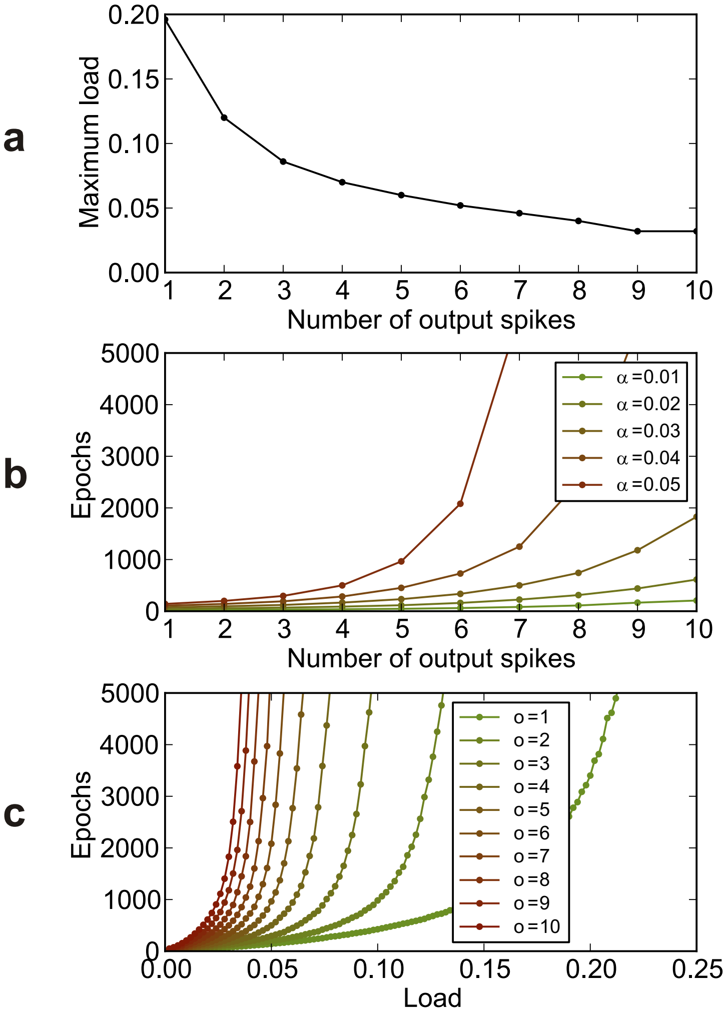

Figure S3: The dependence of chronotron performance on the number of output spikes per trial. The neuron had to learn to have the same output for all inputs, using E-learning. The output consisted of o output spikes, placed at $k\,T/(o+1)$, for $k \in \{1,\ldots,o\}$. (a) The maximum load (the capacity) as a function of the number of output spikes o. (b) The number of learning epochs required for correct learning as a function of the number of output spikes, for various loads α. (c) The number of learning epochs required for correct learning as a function of load, for various numbers of output spikes o.

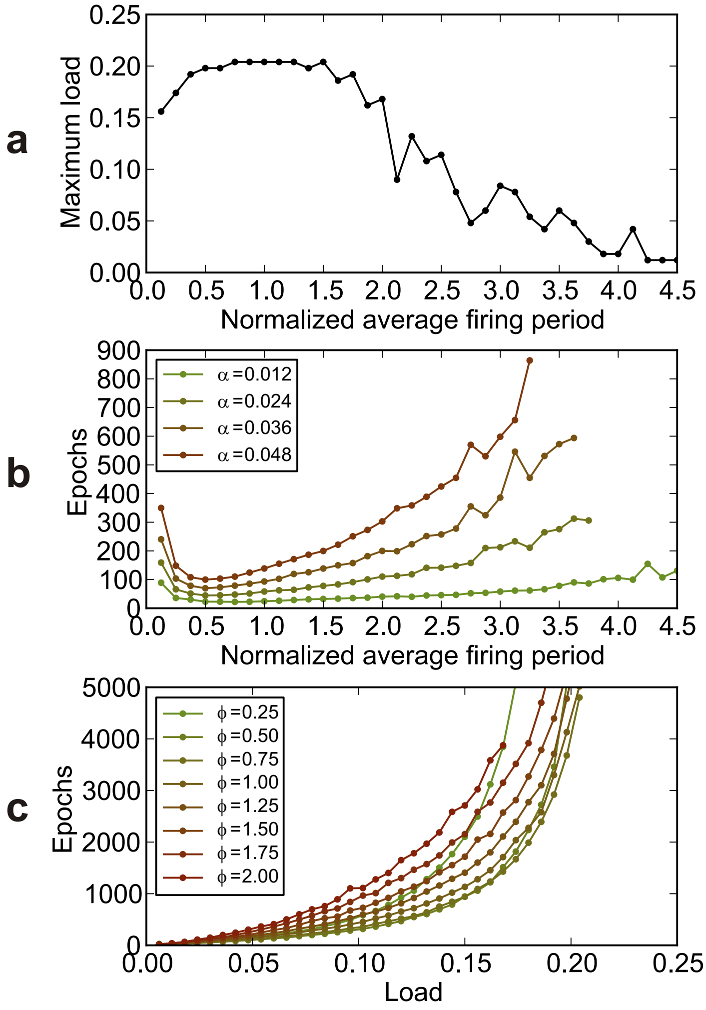

Figure S4: The dependence of chronotron performance on the firing rate of the inputs. The inputs were generated using a Gamma process having a normalized average period ϕ (see text). (a) The maximum load (the capacity) as a function of the normalized average period. (b) The number of learning epochs required for correct learning as a function of the normalized average period, for various loads α. (c) The number of learning epochs required for correct learning as a function of load, for various values of the normalized average period ϕ.

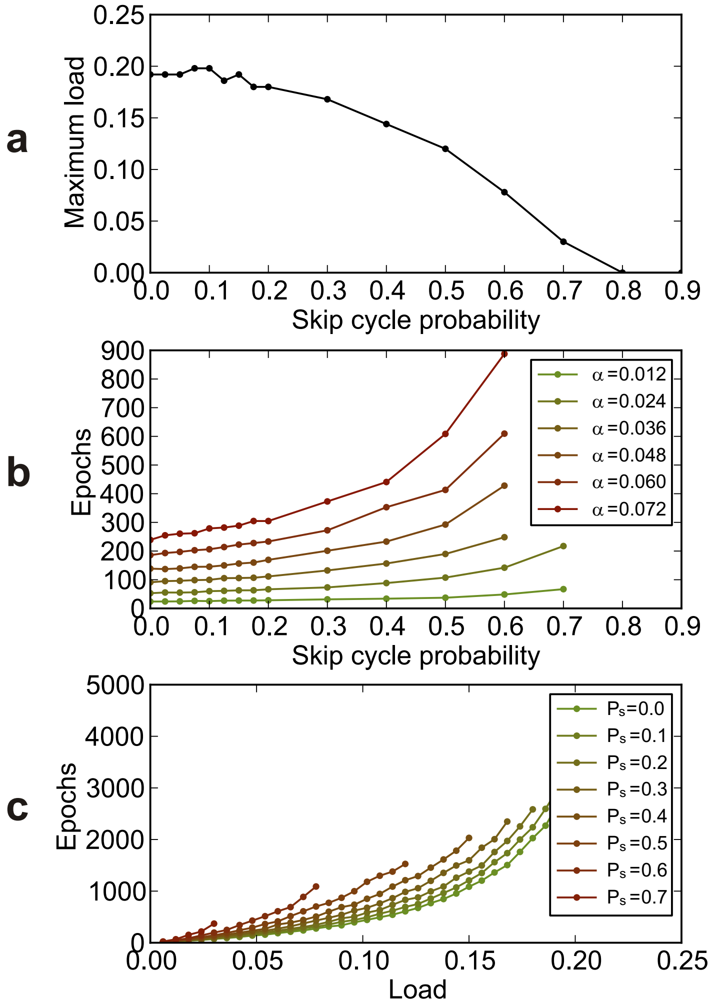

Figure S5: The dependence of chronotron performance on the probability P_s that input neurons skip cycles. Each input spike train consisted of either one spike generated at a random timing or, with a probability P_s, of no spikes. (a) The maximum load (the capacity) as a function of the skip cycles probability. (b) The number of learning epochs required for correct learning as a function of the skip cycles probability, for various loads α. (c) The number of learning epochs required for correct learning as a function of load, for various values of the skip cycles probability P_s.

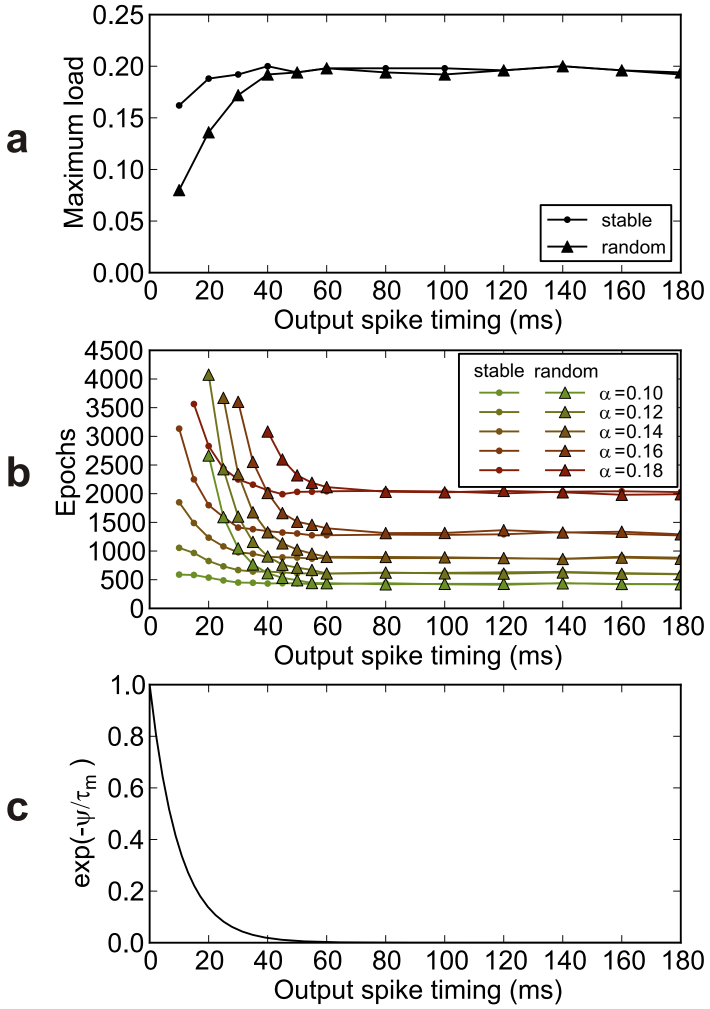

Figure S6: The dependence of chronotron performance on the timing of the output spike and on the initial state of the membrane potential. The neuron had to learn to have the same output for all inputs, using E-learning. The output was one spike at a given timing ψ. At the beginning of each trial, the membrane potential u was either set to $0.8\,\theta$, as in the other experiments (stable initial state), or was generated randomly, with a uniform distribution, between 0 and $0.8\,\theta$ (random initial state). (a) The maximum load (the capacity) as a function of the timing of the output spike. (b) The number of learning epochs required for correct learning as a function of the timing of the output spike, for various loads α. (c) $\exp(-\psi/\tau_m)$, as a reference for comparing the effect on learning of the initial conditions, as a function of the timing of the output spike ψ.

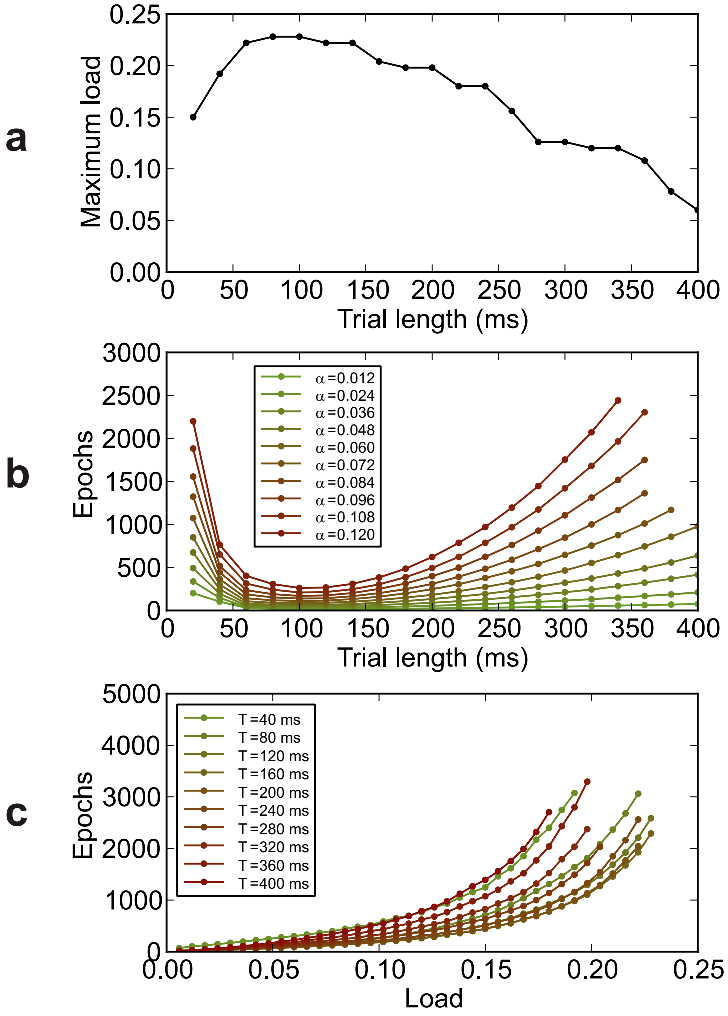

Figure S7: The dependence of chronotron performance on trial length T. (a) The maximum load (the capacity) as a function of the trial length. (b) The number of learning epochs required for correct learning as a function of the trial length, for various loads α. (c) The number of learning epochs required for correct learning as a function of load, for various values of the trial length T. Best performance was achieved for $T = 80 \ldots 100$ ms ($T/\tau_m = 8 \ldots 10$).

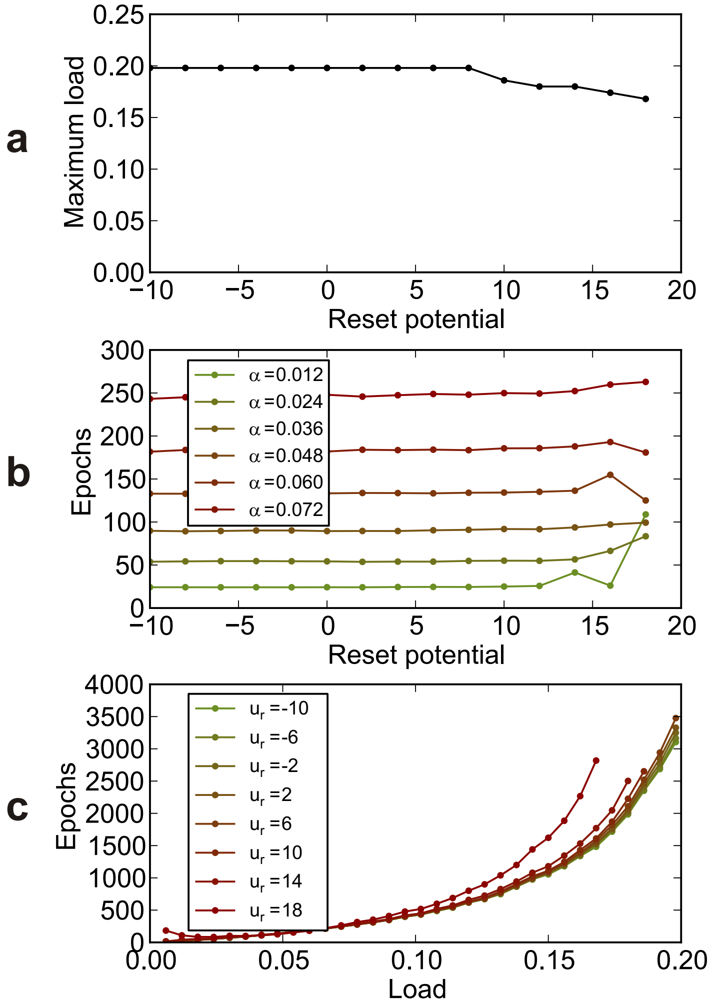

Figure S8: The dependence of chronotron performance on the reset potential u_r. (a) The maximum load (the capacity) as a function of the reset potential. (b) The number of learning epochs required for correct learning as a function of the reset potential, for various loads α. (c) The number of learning epochs required for correct learning as a function of load, for various values of the reset potential u_r. The performance does not depend on the reset potential if it is lower than half of the firing threshold, $\theta/2 = 10$ mV.

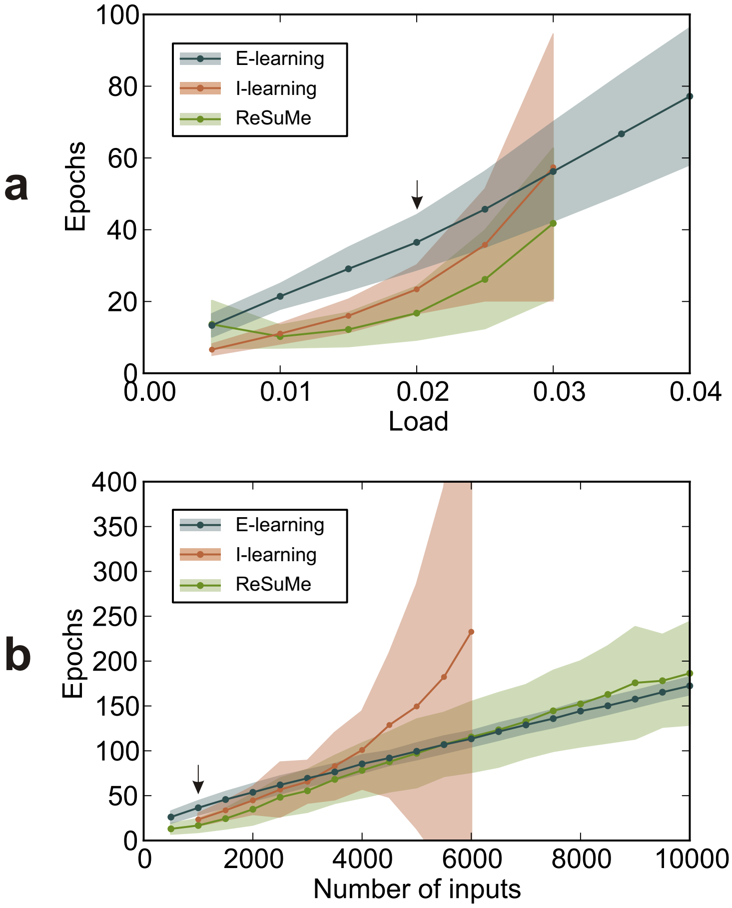

Figure S9: The performance of learning rules when their parameters were optimized for fast learning for $p = 20$, $n = 1,000$ ($\alpha = 0.02$). (a) The number of learning epochs required for correct learning as a function of the load α, for $n = 1,000$. Correct learning was not achieved for I-learning and ReSuMe for α larger than 0.03. (b) The number of learning epochs required for correct learning as a function of the number of inputs n. Correct learning was not achieved for I-learning for $n = 500$ nor n larger than 6,000. Averages and standard deviations over 500 realizations. The arrows indicate the conditions for which the parameters were optimized.

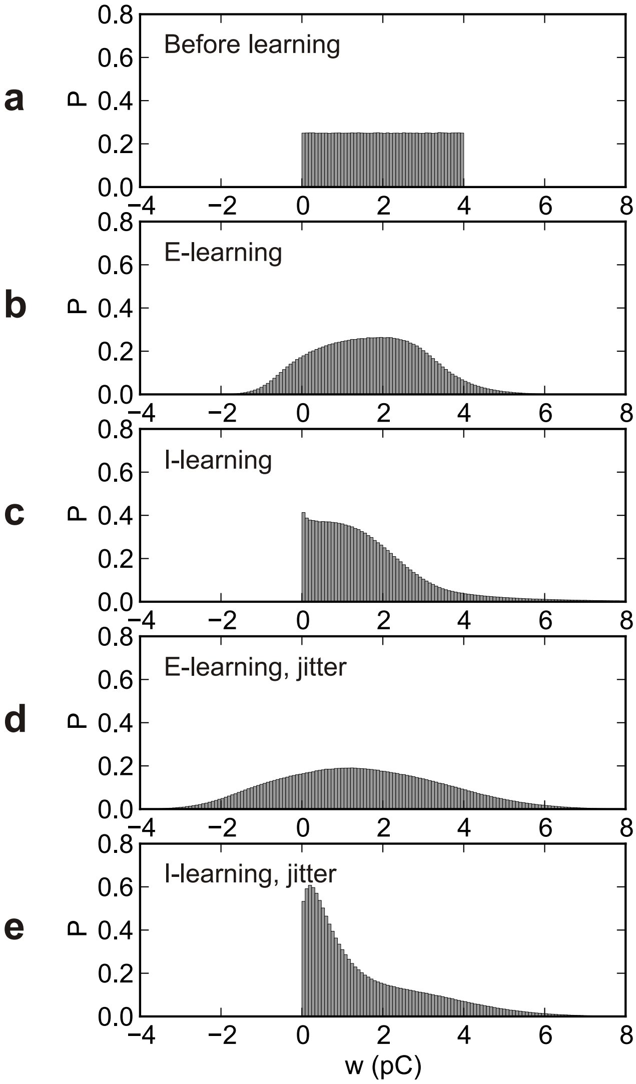

Figure S10: The distribution of the synaptic efficacies, before and after learning, for the experiments presented in Fig. 4. (a) Before learning. (b)-(e) After 400 learning epochs. (b), (d) E-learning. (c), (e) I-learning. (b), (c) No jitter. (d), (e) A gaussian jitter with an amplitude of 5 ms is applied to the inputs.

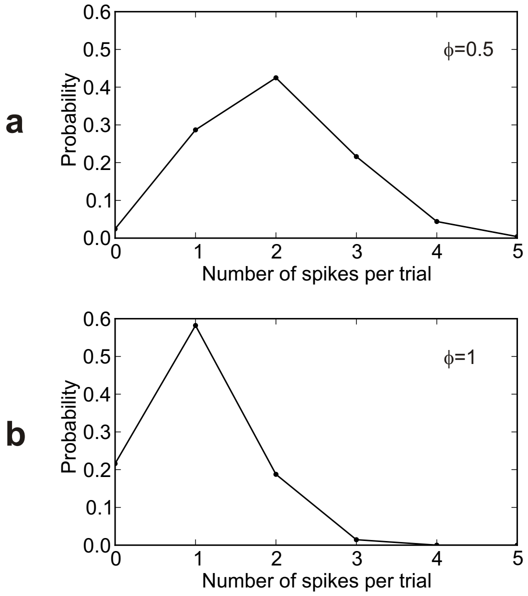

Figure S11: The distribution of the number of input spikes per trial, for inputs generated using a Gamma process, as in Fig. S4. (a) $\phi = 0.5$. (b) $\phi = 1$.

2 Detailed methods

2.1 The neural model

Our analysis uses the Spike Response Model (SRM) of spiking neurons, which reproduces with high accuracy the dynamics of the complex Hodgkin-Huxley neural model while being amenable to analytical treatment (Gerstner, 2001; Gerstner and Kistler, 2002). For this model, the dynamics of the membrane potential u of a neuron as a function of the time t is given by

$$u(t) = \eta(t, \hat{t}) + \sum_j w_j \sum_f \epsilon_j(t, \hat{t}, t_j^f), \tag{2}$$

where η is a kernel that represents the refractoriness caused by the last spike of the neuron; $\hat{t}$ is the last time the neuron fired before t; the first sum runs over all neurons j presynaptic to the considered neuron; w_j is the synaptic efficacy of the synapse from neuron j; the second sum runs over the set of the timings when spikes fired by neuron j reach the postsynaptic neuron, $t_j^f \in \mathscr{F}_j$; ϵ_j is a normalized kernel that determines the postsynaptic potential (PSP) caused by a presynaptic spike. We have chosen the reference of the membrane potential such that the resting potential of the neuron is 0. The ϵ kernel is causal, i.e. $\epsilon_j(t, \hat{t}, t_j^f) = 0$ for $t < t_j^f$, and also decays to 0 for $t \to \infty$. We denote as λ_j the total normalized PSP,

$$\lambda_j(t, \hat{t}, \mathscr{F}_j) = \sum_f \epsilon_j(t, \hat{t}, t_j^f) \tag{3}$$

and thus

$$u(t) = \eta(t, \hat{t}) + \sum_j w_j \, \lambda_j(t, \hat{t}, \mathscr{F}_j). \tag{4}$$

When the membrane potential reaches the firing threshold θ, the neuron fires a spike and the membrane potential is reset.

2.2 Graphical illustration of the chronotron problem

We consider the problem of training the plastic parameters of a spiking neuron, such that the spike train of the trained neuron is, for a given input, as close as possible to some given target spike train. Although we focus here on training the synaptic efficacies, the plastic parameters may also be synaptic delays, the firing threshold, the membrane time constant, etc.

We consider the vector $\mathbf{w}$ having as components the synaptic efficacies w_j and the vector $\boldsymbol{\lambda}(t, \hat{t}, \{\mathscr{F}_j\})$ having as components the normalized total synaptic PSPs $\lambda_j(t, \hat{t}, \mathscr{F}_j)$. Each of these vectors has a dimension equal to the number of synapses, n. The equation that defines the dynamics of the Spike Response Model, Eq. 4, can be then rewritten in vectorial form as

$$u(t) = \eta(t, \hat{t}) + \mathbf{w} \cdot \boldsymbol{\lambda}(t, \hat{t}, \{\mathscr{F}_j\}). \tag{5}$$

We consider here that synaptic changes are applied on a time scale that is much slower than the time scale of the variation of the PSPs, or that the synaptic changes are applied at the end of episodes of information processing.

The dynamics of the normalized synaptic PSPs define a trajectory of $\boldsymbol{\lambda}$ in the corresponding n-dimensional space. After each postsynaptic spike, the normalized PSPs are reset to 0 and thus this trajectory always starts from the origin of the space. The neuron fires a spike when $u(t) = \theta$, or

$$\mathbf{w} \cdot \boldsymbol{\lambda}(t, \hat{t}, \{\mathscr{F}_j\}) = \theta - \eta(t, \hat{t}). \tag{6}$$

The magnitude of the projection of the $\boldsymbol{\lambda}$ vector on the $\mathbf{w}$ vector is $(\mathbf{w} \cdot \boldsymbol{\lambda}) / |\mathbf{w}|$. Thus, the neuron fires a spike when the magnitude of the projection of $\boldsymbol{\lambda}$ on $\mathbf{w}$ reaches $\left(\theta - \eta(t, \hat{t})\right) / |\mathbf{w}|$, i.e. $\boldsymbol{\lambda}$ reaches a spike-generating hyperplane which is perpendicular on the vector $\mathbf{w}$ and at a distance $\left(\theta - \eta(t, \hat{t})\right) / |\mathbf{w}|$ of the origin. This is illustrated in Fig. 1 for a neuron with 3 synapses and in Figs. S12 and S13 for a neuron with 2 synapses. These artificially low numbers of synapses were chosen because it is difficult to visualize spaces with dimensions higher than 3.

The chronotron problem can then be understood as setting the vector $\mathbf{w}$ such that the spike-generating hyperplane that it defines is such that $\boldsymbol{\lambda}$ reaches it at the moments of the target spikes. This problem is very similar to the problem that needs to be solved in reservoir computing (Jaeger, 2001; Maass et al., 2002; Schrauwen et al., 2007), where the state of a high-dimensional dynamical system, such as our vector $\boldsymbol{\lambda}$, is processed by a (usually) linear discriminator such that the switch between output states (the crossing of the hyperplane defined by the linear discriminator) happens at desired moments of time.

2.3 Analytical derivation of the E-learning rule

For a given input, the trained neuron fires at the moments t^f, where f represents the index of the spike in the spike train. The ordered set of the spikes in the spike train fired by the neuron is $\mathscr{F} = \{t^1, t^2, ...\}$. The target spike train that the neuron should fire for that input is $\tilde{\mathscr{F}} = \{\tilde{t}^1, \tilde{t}^2, ...\}$.

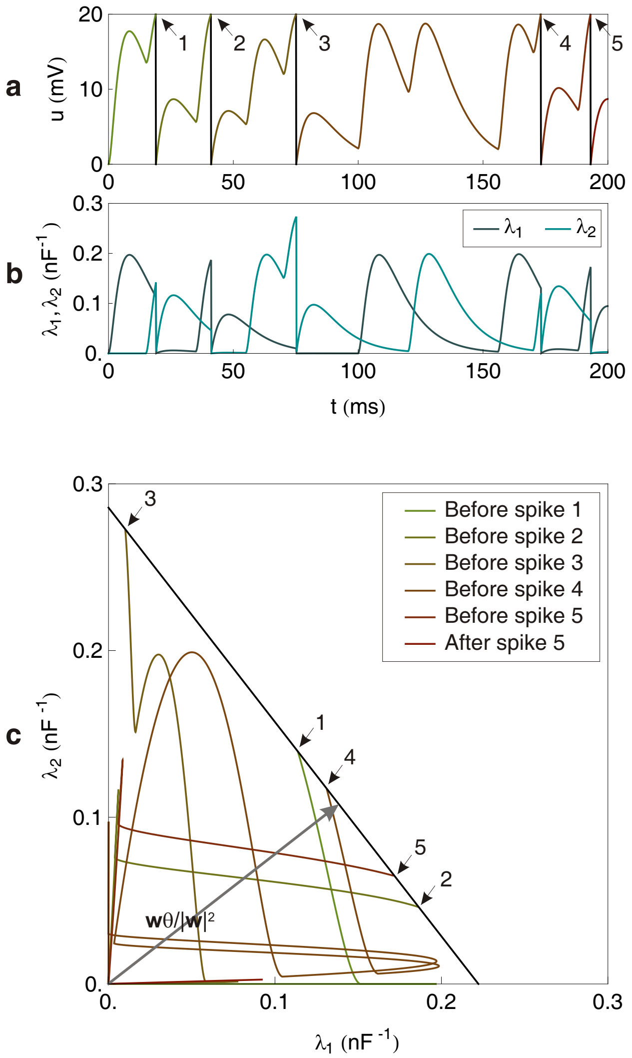

Figure S12: A graphical illustration of the chronotron problem for a neuron with $n = 2$ synapses and $\eta(t, \hat{t}) = 0$. (a) The dynamics of the membrane potential u. The numbered arrows indicate the timings when the membrane potential reaches the firing threshold and spikes are fired. (b) The dynamics of the two components of $\boldsymbol{\lambda}$. (c) The trajectory of $\boldsymbol{\lambda}$. Spikes are generated when the trajectory reaches the spike-generating hyperplane, which is here a line. The chronotron problem is solved by adjusting the location of the spike-generating hyperplane, through changes in **w**, such that the timings of the fired spikes are the target ones. The numbered arrows indicate the generation of spikes at the times when the spike-generating line is reached. The neuron's parameters are as described in section 2.7, with $\mathbf{w} = (90, 70)$ pC and several input spikes generated at random timings, the same as in Fig. S1.

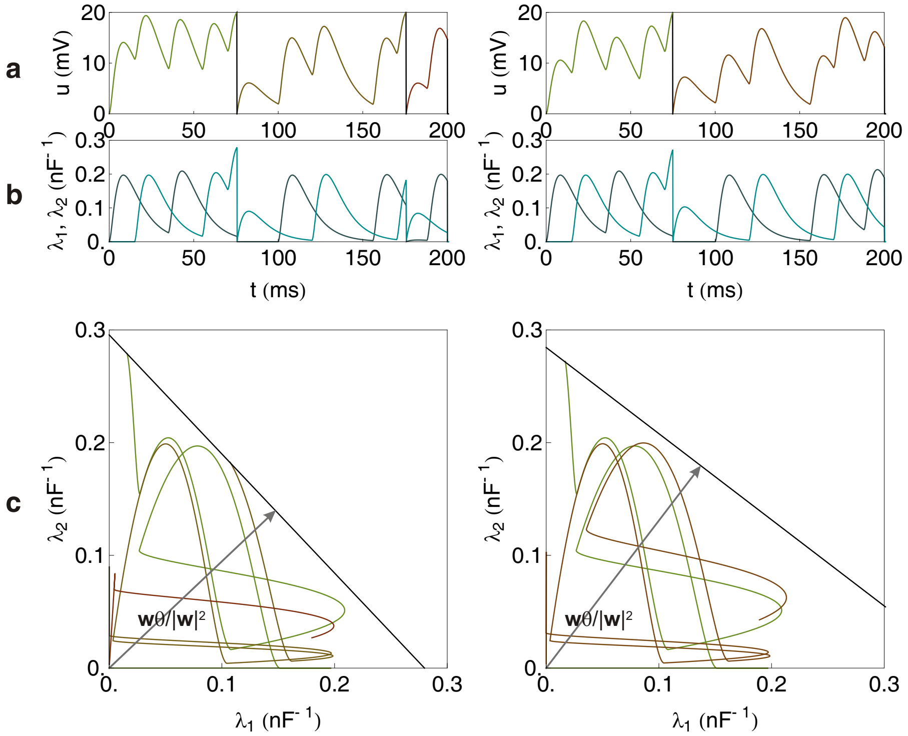

Figure S13: As in Fig. S12, but for other values of **w**, resulted through the application of E-learning, starting from the situation in Fig. S12, and having as a target the generation of one spike at 75 ms. Left: during learning. Right: after learning converged, $\mathbf{w} \approx (53.75, 70.32)$ pC.

The key to solving the chronotron problem is finding appropriate error functions that can be afterwards minimized through methods like gradient descent in the space of the plastic parameters. In order to find such an error function, we start from the Victor & Purpura (VP) family of metrics based on spike times that defines distances between pairs of spike trains (Victor and Purpura, 1996, 1997). The distance between two spike trains is defined as the minimum cost required to transform one into the other. This is the sum of the costs assigned to either insertion of spikes, removal of spikes or shifting the timing of spikes. The cost of adding or deleting a single spike is set to 1, while the cost of shifting a spike by an amount Δt is $\sigma(|\Delta t|/\tau_q)$, where σ is a positive, increasing function with $\sigma(0) = 0$, and τ_q is a positive time constant that is a parameter of the metric. The commonly used form of this function is simply $\sigma(x) = x$ (Victor and Purpura, 1996, 1997).

Because the transformation is of minimal cost, the operations that define it are severely constrained. The same spike cannot be both moved and deleted, nor inserted and moved, nor inserted and deleted. A spike can be moved in only one direction, and the trajectories of moved spikes should not intersect (Victor and Purpura, 1996). Thus, in a transformation of minimal cost of the actual spike train $\mathscr{F}$ into the target one $\tilde{\mathscr{F}}$, the operations involved are the following: removal of spikes (that are not previously moved); insertion of spikes (at their target timings, so that they are not moved after insertion); and shifting of spikes toward their target timings. The order of these operations is irrelevant.

We denote as $\mathscr{F}^*$ the subset of $\mathscr{F}$ that represents the spikes that should be eliminated; and as $\tilde{\mathscr{F}}^*$ the subset of $\tilde{\mathscr{F}}$ that represents the timings of target spikes at which new spikes should be inserted into $\mathscr{F}$. The spikes in the actual spike train that are not eliminated, $\mathscr{F} - \mathscr{F}^*$, are in a one-to-one correspondence with the spikes in the target spike train for which a correspondent is not inserted, $\tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*$, and they should be moved towards their targets. We say that the spikes in $\mathscr{F}^*$ and $\tilde{\mathscr{F}}^*$ are independent, while the spikes in $\mathscr{F} - \mathscr{F}^*$ and $\tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*$ are linked or paired to their correspondent (match). The VP distance is then

$$E_{VP} = \sum_{t^f \in \mathscr{F}^*} 1 + \sum_{\tilde{t}^f \in \tilde{\mathscr{F}}^*} 1 + \sum_{\substack{(t^f,\, \tilde{t}^g) \\ t^f \in \mathscr{F} - \mathscr{F}^* \\ \tilde{t}^g \in \tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*}} \sigma(|t^f - \tilde{t}^g|/\tau_q), \tag{7}$$

where the first sum equals the number of elements in $\mathscr{F}^*$, the second sum equals the number of elements in $\tilde{\mathscr{F}}^*$, and the last sum runs over all unique pairs of matching spikes.

The existing algorithm that computes the VP distance between two given spike trains (Victor and Purpura, 1996, 1997) can be extended in order to also compute the sets $\mathscr{F}^*$, $\tilde{\mathscr{F}}^*$ and their complements. We present this extended algorithm in Section 2.9.

We can thus determine which of the actual spikes fired by the trained neuron should be removed, which target spikes do not have a correspondent and thus new spikes should be created to match them, and which spikes should be moved and toward which of the targets, in order to transform the actual spike train into the target one with a minimal cost. The plastic parameters should then change in order to perform this transformation.

For an existing spike at t^f that should be moved towards $\tilde{t}^g$, the error that should be minimized is $\sigma(|t^f - \tilde{t}^g|/\tau_q)$. This can be differentiated piecewisely with respect to the plastic parameters, so that the changes of the parameters that lead to a decrease of the error can be computed. However, the spikes in $\mathscr{F}^*$ and $\tilde{\mathscr{F}}^*$ that are independent contribute to the distance a constant term of 1 each, and this is not differentiable with respect to the plastic parameters. In order to be able to minimize the contribution of these terms to the distance between the spike trains, we must focus more closely on the mechanisms of spike creation and removal.

The neuron fires a spike when its membrane potential u reaches the firing threshold θ; after a spike is emitted, the membrane potential is reset to u_r. If a new spike should be created at a target timing $\tilde{t}^f$, this is because the membrane potential is not high enough at that moment. In order to minimize the spike train distance by creating a new spike, we should thus minimize the error $\theta - u(\tilde{t}^f)$. This reflects the amount with which the membrane potential should increase at $\tilde{t}^f$ in order to reach the threshold and let the neuron fire at the target timing. Analogously, if an actual spike at t^f should be removed we should decrease the membrane potential at that timing and minimize $u(t^f)$. Note that we minimize the membrane potential at the *current* moments of the spikes to be removed. The membrane potential at a *generic* moment of these spikes equals the firing threshold, thus being a constant that cannot be minimized. The effect of this minimization will be, in most cases, a change of the timing of these spikes, until their elimination.

These error terms that depend on the values of the membrane potential at the timings of the spikes are piecewisely differentiable with respect to the plastic parameters. We will replace, in the error function to be minimized by changes in the plastic parameters, the constant terms corresponding to independent actual and target spikes with these new error terms. Because the new error terms are not commensurable with the original spike train distance, we scale the original terms by a constant, positive parameter γ_d. The final error function that we seek to minimize is thus

$$E = \sum_{t^f \in \mathscr{F}^*} u(t^f) + \sum_{\tilde{t}^f \in \tilde{\mathscr{F}}^*} (\theta - u(\tilde{t}^f)) + \gamma_d \sum_{\substack{(t^f,\, \tilde{t}^g) \\ t^f \in \mathscr{F} - \mathscr{F}^* \\ \tilde{t}^g \in \tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*}} \sigma(|t^f - \tilde{t}^g| / \tau_q). \tag{8}$$

The first sum is over the independent actual spikes, the second sum is over the independent target spikes, and the last sum is over unique pairs of linked spikes, consisting of one target spike and one actual spike that should be moved towards the target one.

We aim to minimize this error function by piecewise gradient descent in the space of the plastic parameters of the trained neuron. We will consider here training the efficacies w_j of the synapses afferent to the neuron, where the index j indicates the synapse. The synaptic changes that aim to minimize the error function are thus

$$\Delta w_j \sim - \frac{\partial E}{\partial w_j}. \tag{9}$$

We have

$$\frac{\partial}{\partial w_j} \sigma\left(\frac{|t^f - \tilde{t}^g|}{\tau_q} \right) = \sigma'\left(\frac{|t^f - \tilde{t}^g|}{\tau_q} \right) \frac{\mathrm{sign}(t^f - \tilde{t}^g)}{\tau_q} \frac{\partial t^f}{\partial w_j}, \tag{10}$$

where $\sigma'(x) = \mathrm{d}\sigma(x)/\mathrm{d}x$. Because of the presence of the absolute value function in the argument of σ, its derivative is discontinuous when the actual spike is at its target timing, $t^f - \tilde{t}^g \to 0$, unless we have $\sigma'(0) = 0$. We would like to fulfill this condition in order to avoid, during learning, oscillations of the emitted spikes around the target positions. Here we will use

$$\sigma(x) = \frac{1}{2}\, x^2. \tag{11}$$

Like for the commonly used form $\sigma(x) = x$, for our choice of σ the switch from considering two spikes (one from each of the two spike trains) as independent to considering them as linked is when the difference of their timings, in absolute value, is $|\Delta t^{\mathrm{switch}}| = 2\,\tau_q$. This is because a pair of independent spikes contributes to the distance with a term of 1 each, for a total of 2 (one actual spike should be removed and a matching spike for the target one should be created); and $\sigma(|\Delta t^{\mathrm{switch}}|/\tau_q) = \sigma(2) = 2$.

We have $\sigma'(x) = x$ and

$$\frac{\partial}{\partial w_j}\sigma\left(\frac{|t^f - \tilde{t}^g|}{\tau_q}\right) = \frac{t^f - \tilde{t}^g}{\tau_q^2}\,\frac{\partial t^f}{\partial w_j}. \tag{12}$$

The derivative of the firing time of the neuron with respect to a synaptic efficacy can be computed by taking into consideration that the firing time depends on the synaptic efficacies through its dependence on the dynamics of the membrane potential of the neuron. However, the membrane potential at a generic firing time is always constant, equal to the firing threshold, and thus we have (Bohte et al., 2002; Booij, 2004):

$$u(t^f) = \theta \tag{13}$$

$$\mathrm{d}u(t^f) = 0. \tag{14}$$

By expanding the last equation, we get

$$\frac{\partial u(t^f)}{\partial w_j}\mathrm{d}w_j + \frac{\partial u(t^f)}{\partial t}\,\frac{\partial t^f}{\partial w_j}\mathrm{d}w_j = 0 \tag{15}$$

and, finally,

$$\frac{\partial t^f}{\partial w_j} = -\frac{1}{\dfrac{\partial u(t^f)}{\partial t}}\,\frac{\partial u(t^f)}{\partial w_j}. \tag{16}$$

By introducing Eqs. 8, 12, and 16 into Eq. 9, we get:

$$\Delta w_j \sim \sum_{\tilde{t}^f \in \tilde{\mathcal{F}}^*} \frac{\partial u(\tilde{t}^f)}{\partial w_j} - \sum_{t^f \in \mathcal{F}^*} \frac{\partial u(t^f)}{\partial w_j} + \frac{\gamma_d}{\tau_q^2} \sum_{\substack{(t^f,\, \tilde{t}^g) \\ t^f \in \mathcal{F} - \mathcal{F}^* \\ \tilde{t}^g \in \tilde{\mathcal{F}} - \tilde{\mathcal{F}}^*}} \frac{t^f - \tilde{t}^g}{\dfrac{\partial u(t^f)}{\partial t}}\,\frac{\partial u(t^f)}{\partial w_j}. \tag{17}$$

In order to be able to compute the derivatives of the membrane potential with respect to the synaptic efficacies, we have to choose a specific neural model. As discussed above, here we use the Spike Response Model, Eq. 4. We can then compute

$$\frac{\partial u(t)}{\partial w_j} = \frac{\partial \eta(t,\hat{t})}{\partial \hat{t}} \frac{\partial \hat{t}}{\partial w_j} + \lambda_j(t,\hat{t},\mathscr{F}_j) + \sum_k w_k \frac{\partial \lambda_k(t,\hat{t},\mathscr{F}_j)}{\partial \hat{t}} \frac{\partial \hat{t}}{\partial w_j} \tag{18}$$

$$\frac{\partial u(t)}{\partial w_j} = \lambda_j(t,\hat{t},\mathscr{F}_j) + \left[\frac{\partial \eta(t,\hat{t})}{\partial \hat{t}} + \sum_k w_k \frac{\partial \lambda_k(t,\hat{t},\mathscr{F}_j)}{\partial \hat{t}} \right] \frac{\partial \hat{t}}{\partial w_j} \tag{19}$$

$$\frac{\partial u(t)}{\partial w_j} = \lambda_j(t,\hat{t},\mathscr{F}_j) + \frac{\partial u(t)}{\partial \hat{t}} \frac{\partial \hat{t}}{\partial w_j} \tag{20}$$

In order to simplify the learning rule, its presentation and its computational implementation, we neglected the last term in the last equation and we used for the simulations the approximation

$$\frac{\partial u(t)}{\partial w_j} \approx \lambda_j(t,\hat{t},\mathscr{F}_j). \tag{21}$$

Another approximation that we used was to replace the factor $1/[\partial u(t^f)/\partial t]$ in Eq. 17 with a constant. This is needed because this factor diverges numerically when a spike is fired while the membrane potential barely reaches the threshold and thus $\partial u(t^f)/\partial t$ is close to 0. This divergence reflects a discontinuity of the studied system (Booij and tat Nguyen, 2005): in this situation, an infinitesimal change of a synaptic efficacy can lead to a finite change of the error function, if this results in the removal of the considered spike. Our error function deals with spike creation or removal trough the two terms that ensure that the membrane potential is increased or, respectively, decreased at the desired timings, such that the number of spikes becomes the desired one and the actual spikes are close to the target ones. It is thus safe to enforce a hard bound for the divergent factor or, as we did here, to replace it with a constant. This constant is positive, because a spike is generated only when the membrane potential increases. We fold this constant and γ_d into a new positive constant, γ_r.

The resulting learning rule, that we call E-learning, is thus

$$\Delta w_j = \gamma \Bigg[\sum_{\tilde{t}^f \in \tilde{\mathscr{F}}^*} \lambda_j(\tilde{t}^f, \widehat{\tilde{t}^f}, \mathscr{F}_j) - \sum_{t^f \in \mathscr{F}^*} \lambda_j(t^f, \hat{t^f}, \mathscr{F}_j) \\ + \frac{\gamma_r}{\tau_q^2} \sum_{\substack{(t^f,\, \tilde{t}^g) \\ t^f \in \mathscr{F} - \mathscr{F}^* \\ \tilde{t}^g \in \tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*}} (t^f - \tilde{t}^g)\, \lambda_j(t^f, \hat{t^f}, \mathscr{F}_j) \Bigg], \tag{22}$$

where γ is the learning rate, a positive parameter.

The E-learning rule can be described more intuitively as follows. For each of the target spikes, if these target spikes are independent (do not have a corresponding actual spike close to them), each synapse j is potentiated proportionally to the normalized postsynaptic potential λ_j at the moments of these target spikes. For each of the independent actual spikes (that do not have a corresponding target spike close to them), each synapse is decreased proportionally to the normalized postsynaptic potential at the moments of these actual spikes. For each pair of matching spikes, each synapse changes proportionally to the difference between the timing of the actual spike and the timing of the target spike in the pair, and also proportionally to the normalized postsynaptic potential at the moment of the actual spike. The first two terms of the learning rule will create or remove spikes in order to match them to the target ones. The last term of the learning rule will move the actual spikes that match the target ones toward their targets. When the timing of the spikes coincide to their targets, the changes of the synaptic efficacies suggested by the learning rule become zero and thus learning stops.

The E-learning rule is appropriate for both excitatory and inhibitory synapses. If we consider that the excitatory synapses have a positive synaptic efficacy w_j and the inhibitory synapses have a negative one, the learning rule in the form presented above can be applied to both cases. Without an extra bounding of the synaptic efficacies, E-learning will transform an excitatory synapse into an inhibitory one or viceversa, as needed for minimizing the error function.

The E-learning rule can also be understood intuitively by considering snapshots of the trajectory of $\boldsymbol{\lambda}(t)$ at the timings of target and actual spikes of the trained neuron. The equation defining the E-learning rule, Eq. 22, can be written in vectorial form as

$$\begin{aligned}\Delta\mathbf{w} = \gamma \Bigg[&\sum_{\tilde{t}^f \in \tilde{\mathscr{F}}^*} \boldsymbol{\lambda}(\tilde{t}^f, \widehat{\tilde{t}^f}, \{\mathscr{F}_j\}) - \sum_{t^f \in \mathscr{F}^*} \boldsymbol{\lambda}(t^f, \hat{t^f}, \{\mathscr{F}_j\}) \\ &+ \frac{\gamma_r}{\tau_q^2} \sum_{\substack{(t^f,\, \tilde{t}^g) \\ t^f \in \mathscr{F} - \mathscr{F}^* \\ \tilde{t}^g \in \tilde{\mathscr{F}} - \tilde{\mathscr{F}}^*}} (t^f - \tilde{t}^g)\, \boldsymbol{\lambda}(t^f, \hat{t^f}, \{\mathscr{F}_j\}) \Bigg].\end{aligned} \tag{23}$$

At the timings $\tilde{t}^f$ of independent target spikes, the spike-generating hyperplane must be brought closer to the $\boldsymbol{\lambda}(\tilde{t}^f)$ vector and the $\mathbf{w} \cdot \boldsymbol{\lambda}(\tilde{t}^f)$ product must be increased. This can be done best, for a given perturbation of $\mathbf{w}$, by increasing just the component of $\mathbf{w}$ that is parallel to the $\boldsymbol{\lambda}(\tilde{t}^f)$, which would lead $\mathbf{w}$ to turn towards $\boldsymbol{\lambda}(\tilde{t}^f)$. This leads to setting $\Delta\mathbf{w} \sim \boldsymbol{\lambda}(\tilde{t}^f)$, hence the first term of Eq. 23. At the timings t^f of independent actual spikes, $\boldsymbol{\lambda}(t^f)$ reaches the spike-generating hyperplane which must be then moved away from $\boldsymbol{\lambda}(t^f)$ and thus it leads to $\Delta\mathbf{w} \sim -\boldsymbol{\lambda}(t^f)$, hence the second term of Eq. 23.

When an actual spike at t^f is followed closely by a matching target spike at $\tilde{t}^g$, bringing the spike-generating hyperplane closer to $\boldsymbol{\lambda}(\tilde{t}^g)$ is deleterious since it does not take into account that $\boldsymbol{\lambda}$ has just been reset to 0 because of the recent actual spike. In this case, what should be done is just delaying the actual spike. This could be done by moving the spike-generating hyperplane away from $\boldsymbol{\lambda}(t^f)$, proportionally to $t^f - \tilde{t}^g$. When a target spike at $\tilde{t}^g$ is followed closely by a matching actual spike at t^f, bringing the spike-generating hyperplane closer to $\boldsymbol{\lambda}(\tilde{t}^g)$, in the same way as in the case of an independent target spike described above, would bring the timing of the actual spike closer to the target one, but in an imprecise fashion. We would like that the convergence of the actual spike towards the target one to be smooth. The third term of Eq. 23, $\Delta\mathbf{w} \sim (t^f - \tilde{t}^g)\boldsymbol{\lambda}(t^f)$, takes care of the last two situations.

2.4 The I-learning rule

The form of the synaptic changes indicated by the previously described E-learning rule depends on whether spikes are independent or not, being different in the two cases. While, as the simulations have shown, this learning rule is very efficient, the biological plausibility of this switch of the form of the synaptic changes is debatable. For this reason, we sought a more biologically plausible supervised learning rule.

One may consider the limit $\tau_q \to 0$ (all spikes are independent; $\mathscr{F} = \mathscr{F}^*$ and $\tilde{\mathscr{F}} = \tilde{\mathscr{F}}^*$) and thus consider just the first two terms of the E-learning rule, which depend on the normalized postsynaptic potentials. The switch of the form of the synaptic changes as a function of the pairing status of the spikes is then removed. However, due to the spike generation mechanism, the postsynaptic potentials suffer a discontinuity after each actual spike, being reset to zero. If there is no distinct treatment of pairs of close actual and target spikes, this leads to a discontinuity of the synaptic changes when an actual spike oscillates around a target one. Learning does not converge to a stable firing of the actual spikes at the target timings. Moreover, it is not clear whether the normalized postsynaptic potential (i.e., the postsynaptic potential with the synaptic efficacy factored out) is a quantity available to the synapse.

For these reasons, we heuristically defined a new learning rule. As before, we wanted the synaptic changes to depend on a quantity that reflects the contribution of each synapse to the membrane potential, a quantity that would be correlated to λ_j, which is used by the analytically-derived E-learning rule. As in E-learning with $\tau_q \to 0$, the synaptic changes for excitatory synapses should be determined by synaptic increases proportional to the value of the considered quantity at the timing of the target spikes and by synaptic decreases proportional to the value of that quantity at the timing of the actual spikes. In this case, when the actual spikes coincide with the target ones, the terms cancel out, resulting in the convergence of the learning rule. Another condition was that the sum of the terms corresponding to a pair of close actual and target spikes converges continuously to zero when the actual spikes moves towards the target one. We also wanted that the quantity used by the learning rule to be locally available to the synapse, thus ensuring its biological plausibility. We thus used the synaptic current, I_j, as this quantity. The resulting learning rule, that we call I-learning, is thus:

$$\Delta w_j = \gamma \,\mathrm{sign}(w_j) \left[\sum_{\tilde{t}^f \in \tilde{\mathscr{F}}} I_j(\tilde{t}^f) - \sum_{t^f \in \mathscr{F}} I_j(t^f) \right], \tag{24}$$

with γ being the learning rate, a positive constant. Although we did not make it explicit in the notation, the synaptic currents I_j on each synapse obviously depend on the parts of the presynaptic spike trains $\mathscr{F}_j$ coming through that synapse previous to the moment at which I_j is evaluated. The $\mathrm{sign}(w_j)$ in the learning rule (i.e., ± 1 as a function of whether the synapse is excitatory or inhibitory) reflects that the sign of the synaptic changes depends on the sign of the synaptic efficacy. For excitatory synapses, both w_j and I_j are positive, while for negative synapses both are negative.

Because I_j is proportional to w_j, the I-learning rule does not allow an excitatory synapse to become inhibitory or viceversa, for small γ, thus corresponding to the biological reality.

This learning rule is quite similar to the existing ReSuMe learning rule (Ponulak, 2005; Ponulak and Kasiński, 2006; Ponulak, 2008; Ponulak and Kasiński, 2010). As in ReSuMe, actual and target postsynaptic spikes lead to synaptic changes of opposite signs, such that when the actual spike train corresponds to the target one the terms cancel out and synapses become stable. In contrast to ReSuMe, where synaptic changes depend exponentially on pairs of pre- and postsynaptic spikes, as in some models of spike-timing-dependent plasticity, here we consider that synaptic changes depend on the value of the synaptic current.

E-learning is an episodic learning rule, because one needs the actual spikes fired during the entire trial under study in order to compute which spikes are independent and which are linked, although one can imagine approximate algorithms for matching the spike trains, which would also work online. However, I-learning can be applied for both episodic learning and online learning, only the latter case being biologically plausible.

2.5 ReSuMe

We have also performed simulations using the ReSuMe learning rule (Ponulak, 2005; Ponulak and Kasiński, 2006; Ponulak, 2008; Ponulak and Kasiński, 2010), in order to compare it to the new learning rules introduced here. We used the following form of ReSuMe:

$$\Delta w_j = \gamma \left\{ \sum_{\tilde{t}^f \in \tilde{\mathscr{F}}} \left[a_R + \sum_{t_j^g < \tilde{t}^f} \exp\left(-\frac{\tilde{t}^f - t_j^g}{\tau_R} \right) \right] - \sum_{t^f \in \mathscr{F}} \left[a_R + \sum_{t_j^g < t^f} \exp\left(-\frac{t^f - t_j^g}{\tau_R} \right) \right] \right\}, \tag{25}$$

where γ is the learning rate, a_R is a non-Hebbian term, and τ_R is a time constant, all being positive parameters of the learning rule.

2.6 Details of the neural model

In order to be able to test the learning rules in a computer simulation, we must define the forms of the ϵ and η kernels of the Spike Response Model. We define them to correspond to the classical leaky integrate-and-fire neural model, which is a particular case of the Spike Response Model (Gerstner and Kistler, 2002). A further choice must be made for the form of the synaptic currents. We modeled the kernel α that reflects the form of the synaptic current generated by the arrival of a presynaptic spike through the synapse j at the timing t_j^f as a difference of two exponentials (double-exponential current):

$$\alpha(t, t_j^f) = \frac{1}{\tau_s - \tau_r} \left[\exp\left(-\frac{t - t_j^f}{\tau_s} \right) - \exp\left(-\frac{t - t_j^f}{\tau_r} \right) \right], \tag{26}$$

for $t \geq t_j^f$, where τ_s and τ_r are positive parameters (time constants). The α kernel is illustrated in Fig. S14 (a).

The synaptic current contributed by one spike at t_j^f is the product of the synaptic efficacy w_j and of the normalized α kernel:

$$I_j^f(t, t_j^f) = w_j \, \alpha(t, t_j^f). \tag{27}$$

The α kernel is normalized:

$$\int_{t_j^f}^{\infty} \alpha(t, t_j^f) \, \mathrm{d}t = 1, \tag{28}$$

and thus the synaptic efficacy w_j represents the total charge transmitted to the postsynaptic neuron as a consequence of one presynaptic spike.

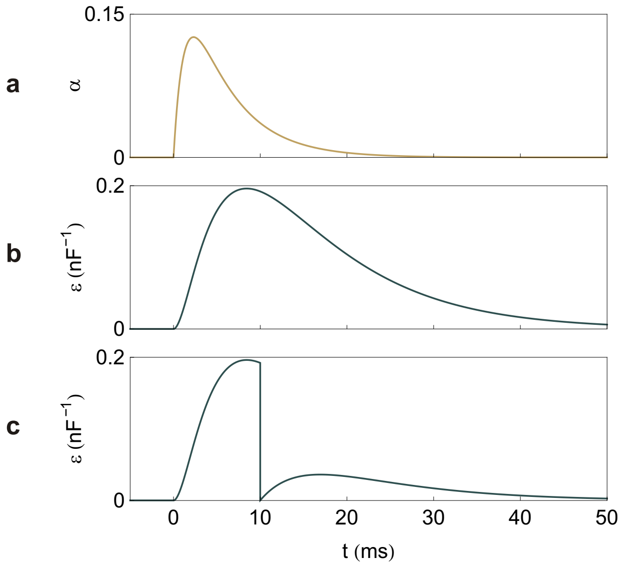

Figure S14: (a) The α kernel. (b), (c) The ϵ kernel. In (b) there is no postsynaptic spike. In (c), a postsynaptic spike is fired at $t = 10$ ms. A presynaptic spike is received at $t = 0$. The parameters are as described in Section 2.7.

The synaptic current generated through the synapse j by one or more presynaptic spikes is

$$I_j(t) = \sum_{\substack{t_j^f \in \mathscr{F}_j \\ t_j^f \le t}} I_j^f(t, t_j^f), \tag{29}$$

and the total synaptic current received by the neuron from all synapses is

$$I(t) = \sum_j I_j(t). \tag{30}$$

The dynamics of the membrane potential u of the leaky integrate-and-fire neuron is defined by

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = -\frac{u(t)}{\tau_m} + \frac{I(t)}{C}, \tag{31}$$

where τ_m is the time constant of the neuron's leakage and C is the capacity of the neuron's membrane (we use here a scale for the membrane potential where the resting potential is 0). When the membrane potential reaches the threshold θ, the neuron fires a spike and the membrane potential is reset to the reset potential u_r.

By integrating the last equation between the moment $\hat{t}$ of the last emitted spike before t, and, respectively, t, we get

$$u(t) = u_r \exp\left(-\frac{t-\hat{t}}{\tau_m}\right) + \frac{1}{C}\int_{\hat{t}}^{t} I(s) \exp\left(-\frac{t-s}{\tau_m}\right) \mathrm{d}s. \tag{32}$$

By expanding $I(s)$ into its components generated by each presynaptic spike, we get

$$u(t) = u_r \exp\left(-\frac{t-\hat{t}}{\tau_m}\right) + \frac{1}{C}\sum_j w_j \sum_{t_j^f \le t} \int_{\hat{t}}^{t} \alpha_j(s, t_j^f) \exp\left(-\frac{t-s}{\tau_m}\right) \mathrm{d}s. \tag{33}$$

We define:

$$\eta(t,\hat{t}) = u_r \, \exp\left(-\frac{t-\hat{t}}{\tau_m}\right) \tag{34}$$

$$\epsilon_j(t,\hat{t},t_j^f) = \frac{1}{C}\int_{\hat{t}}^{t} \alpha_j(s,t_j^f)\, \exp\left(-\frac{t-s}{\tau_m}\right)\, \mathrm{d}s \tag{35}$$

We can then express the dynamics of the integrate-and-fire neuron in the form of the Spike Response Model, Eq. 2:

$$u(t) = \eta(t,\hat{t}) + \sum_j w_j \sum_f \epsilon_j(t,\hat{t},t_j^f). \tag{36}$$

After performing the integration in Eq. 35 by taking into account the form of the α kernel given by Eq. 26, we get:

$$\begin{aligned}
\epsilon_j(t,\hat{t},t_j^f) = {} & \frac{\tau_m}{C\,(\tau_s-\tau_r)} \\
& \times \begin{cases}
\frac{\tau_s}{\tau_m-\tau_s}\left[\exp\left(-\frac{t-t_j^f}{\tau_m}\right)-\exp\left(-\frac{t-t_j^f}{\tau_s}\right)\right] \\
\quad -\frac{\tau_r}{\tau_m-\tau_r}\left[\exp\left(-\frac{t-t_j^f}{\tau_m}\right)-\exp\left(-\frac{t-t_j^f}{\tau_r}\right)\right], & \text{if} \quad t \geq t_j^f > \hat{t}, \\
\frac{\tau_s}{\tau_m-\tau_s}\left[\exp\left(-\frac{t-\hat{t}}{\tau_m}\right)\exp\left(-\frac{\hat{t}-t_j^f}{\tau_s}\right)-\exp\left(-\frac{t-t_j^f}{\tau_s}\right)\right] \\
\quad -\frac{\tau_r}{\tau_m-\tau_r}\left[\exp\left(-\frac{t-\hat{t}}{\tau_m}\right)\exp\left(-\frac{\hat{t}-t_j^f}{\tau_r}\right)-\exp\left(-\frac{t-t_j^f}{\tau_r}\right)\right], & \text{if} \quad t \geq \hat{t} \geq t_j^f, \\
0, & \text{otherwise.}
\end{cases}
\end{aligned} \tag{37}$$

The form of ϵ is illustrated in Fig. S14 (b)-(c).

2.7 Computer simulations

The learning rules were tested and explored in computer simulations. We trained an integrate-and-fire neuron with double-exponential synaptic currents, as described in Section 2.6. The neuron had a membrane time constant $\tau_m = 10$ ms, a capacity $C = 2.5$ nF, a firing threshold $\theta = 20$ mV, and the resting potential was 0. Except in Fig. S8, the neuron had a reset potential equal to the resting potential, $u_r = 0$. The time constants that define the dynamics of the synaptic currents were $\tau_s = 5$ ms and $\tau_r = 1.25$ ms.

Since we were interested in the coding of information in the precise timing of the spikes, we used an event-driven simulation (D'Haene et al., 2009) where the timing of input spikes were represented with machine precision and the timing of the trained neuron's spikes were computed with a precision of 10^{-5} ms.

The neuron received inputs through n synapses. In Figs. 3, 4, and S2–S8 we used $n = 500$, while in Figs. 5 and S9 n was variable, but at least 500. At the beginning of learning experiments, synaptic efficacies were generated randomly, with an uniform distribution between 0 and w_m. In Figs. 3 and 4 we used $w_m = 2{,}000/n$ pC, while in Figs. 5 and S2–S9 we used $w_m = 1{,}000/n$ pC.

The neuron was trained to learn p input patterns by firing a pre-determined output spike train for each of the inputs. Except in Fig. S7, the length T of the input patterns was of 200 ms. An input pattern consisted of the ensemble of the n input spike trains coming through the n synapses during the interval T. During learning, the input patterns were presented sequentially, in batches consisting of the p patterns, and the synapses were modified at the end of the batches according to the learning rules. A presentation of one input pattern and the simulation of the output of the trained neuron corresponding to this input is called a trial. Each batch of presentations of the p patterns (trials) is called an epoch. Except in Fig. S6, at the beginning of each trial, the membrane potential of the neuron was reset to 0.8 of the value of the firing threshold θ; this value was used in order to allow the neuron to fire even at moments close to the beginning of the trial.

The learning rates that we used were, for E-learning: $\gamma = 1{,}250/(n\,p)$ pC nF in Fig. 3, $\gamma = 2{,}500/(n\,p)$ pC nF in Fig. 4, and $\gamma = 5{,}000/(n\,p)$ pC nF in Figs. 5, S2, S3, S6, and S7. We also used $\gamma_r = 15$ ms. For I-learning, the learning rates were $\gamma = 5/p$ ms in Figs. 3 and 4, and $\gamma = 20/p$ ms in Fig. 5. These values were close to the optimal ones. The inverse proportionality to p reflects the accumulation of the synaptic changes during the presentation of the p patterns. The inverse proportionality with n for E-learning but not for I-learning reflects that the average value of the synaptic efficacies scales inversely proportional to the number of synapses, for about the same behavior of the neuron. In I-learning the changes of synaptic efficacies are proportional to the synaptic current, which is already scaled inversely proportional to the number of synapses as it is proportional to the synaptic efficacy, and thus no scaling with n is needed for the learning rate. We also used $\tau_q = 10$ ms.

The I-learning rule implies that changes of synaptic efficacies are proportional to the synaptic current and thus to the values of the synaptic efficacies. Thus, if the initial synaptic efficacies are all positive, they cannot become negative if the learning rates are sufficiently small. The application in batches of synaptic changes or rounding errors may, however, allow a sign change of the synaptic efficacies in a computer simulation. In our simulations with I-learning we enforced that synaptic efficacies stayed positive, by using a hard bound. For E-learning, we allowed the synapses to switch sign, according to the changes suggested by the learning rule.

In Figs. 3–5, S2, S3, and S6–S9, the input spike trains consisted, for each of the n synapses, of one spike generated at a random timing, distributed uniformly between 0 and T (phase coding of information).

In Fig. 4, we considered that the actual spike matched the target one if there was exactly one actual spike and its timing was within τ_q of the target timing. The probability P_m that the fired spikes matched the target ones was the number of patterns within a trial for which the actual spikes matched the target one, divided by the total number of patterns, $p = 10$.

In the experiments presented in Figs. 4, 5 and S2–S9, we trained the neuron to perform classifications by setting its target output to be the same for several different, randomly generated, inputs. The number c of the different outputs was the number of categories into which the neuron classified the p input patterns. We assigned equal number of patterns into each category, such that p was an integer multiple of c. We considered that an input-output association was learned correctly by the trained neuron if the number of the actual output spikes was the one in the target spike train and each of the output spikes was fired within less than 1 ms of the target timing. We considered that the chronotron was able to learn correctly a particular setup if, during each of 500 realizations of the setup with different, random, initial conditions, all input-output mappings were learned correctly in no more than 10,000 epochs.

The output used a phase-coded representation of the information. Except in Figs. S3 and S6, the target spike train for each category $k \in \{1, \ldots, c\}$ consisted of one spike at $k\, T/(c+1)$.

For each realization of the experiments, both the input patterns and the initial synaptic efficacies were generated randomly. In Figs. 5, S2 and S4, for various values of the number of inputs n we increased the load $\alpha = p/n$ until the chronotron was not able to learn correctly all the 500 random realizations of the setup. The capacity for a particular setup was the maximum load for which the chronotron was able to learn correctly that setup, lower than the first load for which the chronotron was not able to learn correctly the setup. In Figs. S3 and S5–S8, the capacity was the maximum load for which chronotron was able to learn correctly a particular setup, lower than the first load for which the chronotron was not able to learn correctly more than 1% of the 500 random realizations of the setup. For each setup and load, we recorded the minimum number of epochs e after which the chronotron was able to learn correctly the setup.

In Figs. 5, S4, S5, S7, and S8, the experiments were performed with $c = 3$ categories, in Figs. S3 and S6 with $c = 1$, in Fig. S9 with $c = 5$, and in Fig. S2 the number of categories varied.

For E-learning, simulations for n higher than 2,000 were not performed in Fig. 5 because of the high computational cost, due to the high capacity resulted through this learning rule. For example, the simulations required for obtaining the results presented for $n = 2{,}000$ took about 13 days on a computer with 8 Xeon cores running in parallel at 2.33 GHz.

For the simulations using ReSuMe in Fig. 5, we used $\gamma = 75{,}000/(n\, p)$ pC and $\tau_R = 20$ ms. These were optimal parameters, that led to the lowest occurrence of cases where correct learning was not achieved for $p = 21$, $n = 500$ ($\alpha = 0.042$), $c = 3$, from a scan of the γ, τ_R parameter space with a resolution of $2{,}500/(n\, p)$ pC and, respectively, 2 ms. We also used $a_R = 0$. We verified that, for $n = 500$, the capacity did not increase if we used nonzero a_R, for various values spanning several orders of magnitude.

In Fig. 1, we used $\mathbf{w} = (20, 70, 55)$ pC, $\mathscr{F}_1$ ={0, 235, 468, 550, 649, 734, 826, 962} ms, $\mathscr{F}_2$ ={30, 177, 285, 396, 782, 922} ms, and $\mathscr{F}_3$ ={107, 452, 586, 945} ms.

In Figs. S1, S12, and S13 we used $\mathscr{F}_1$ ={0, 35, 100, 156, 188} ms and $\mathscr{F}_2$ ={15, 55, 70, 120, 170} ms. In Figs. S1 and S13, the target spike train was $\tilde{\mathscr{F}}$ = {75} ms.

2.8 The information capacity of the chronotron

The load α of a neuronal classifier is the number of patterns it memorizes per each synapse of the neuron. If the neuron has n synapses and memorizes p patterns, the load is

$$\alpha = \frac{p}{n}. \tag{38}$$

We define the information load ι of a neuronal classifier as the quantity of information it can store for each of the patterns that it memorizes, per each synapse of the neuron. We assume that the patterns are classified into c categories, and the same number of patterns is assigned to each category. The neuron stores then $i = \log_2(c)$ bits of information for each pattern, and the information load is

$$\iota = \frac{p\, i}{n} = \alpha\, i. \tag{39}$$

The information capacity ι_m of a neuronal classifier is the maximum information load it can carry. It depends on both the maximum load α_m it can carry as well as the maximum quantity of information it can store for each pattern, if they are independent.

For the perceptron and the tempotron, which can classify patterns in just $c = 2$ categories, we have $i = 1$ and thus the information load equals the (pattern) load and the information capacity equals the (pattern) capacity.

For chronotrons with phase coding of their outputs, the information capacity depends on the temporal precision of the output spikes and on the duration of the interval in which output spikes can be fired with no loss of capacity. We consider that T_0 is the time interval, at the beginning of each trial, where, if target spikes are located, learning capacity is reduced (see Fig. S6). A chronotron having a precision of the output spike of $\pm\delta t$, can encode, for a trial duration T, at most $c_m = (T - T_0)/(2\,\delta t)$ categories. The information capacity of the chronotron is then

$$\iota_m = \alpha_m \log_2(c_m). \tag{40}$$

The capacity obtained in our simulations for E-learning was $\alpha_m \approx 0.22$. For $T = 200$ ms, $\delta t = 1$ ms, $T_0 \approx 40$ ms, we get the maximum number of categories $c_m \approx 80$, the corresponding information memorized per pattern $i_m = \log_2(c_m) \approx 6.32$ bits per pattern, and the information capacity $\iota_m \approx 1.39$ bits per input synapse.

Barak and Tsodyks (2006) have developed a learning rule that allows an integrate-and-fire neuron with exponential currents to recognize input patterns from a given set, by increasing its firing rate for learned patterns in comparison to the one for background inputs. The maximum number of patterns that this rule can learn is $p_m = n\,\tau'_s/T$, where τ'_s is the decay time constant of the exponential neurons. Thus, the capacity of this rule is $\alpha_m = p_m/n = \tau'_s/T$. If we extrapolate this result to neurons with double-exponential currents by assuming that the same relationship applies if we consider the largest time constant of the double-exponential current, τ_s in our case, instead of τ'_s, then the capacity of a neuron for recognizing patterns would be, for our setup, $\alpha_m = \tau_s/T = 0.025$. It can be seen then that the capacity that we obtained in simulations through E-learning, about 0.22, for having a particular, precisely-timed spike output pattern for each input, is about an order of magnitude larger than the capacity computed for just the recognition of patterns.

2.9 The algorithm for computing the sets of spikes to be removed, inserted or moved

Victor and Purpura (1996, 1997) presented an algorithm for computing the distance between spike trains that they defined, but not one for indicating the pairs of matching spikes (consisting of one spike from each spike train) and the sets of independent spikes that the distance implies. This information represents the structure of the pair of spike trains, as defined by the metric. Here we extend the Victor & Purpura algorithm with the capacity of computing this structure.

When the two spike trains that are compared consist of one that is fixed (the target one) and one that is modifiable (the actual one), as in our supervised learning problem, the set $\tilde{\mathscr{F}}^*$ of independent spikes in the target spike train corresponds to timings when new spikes should be created in the actual spike train; the set $\mathscr{F}^*$ of independent spikes in the actual spike train represents the spikes that have to be removed; and pairs of matching spikes define the set of actual spikes that have to move and their targets.

The original algorithm (Victor and Purpura, 1996, 1997) computes the distance between spike trains inductively, as follows. Let $D_{i,\,j}$ be the distance between the spike trains composed of the first i spikes of $\mathscr{F}$, $\mathscr{F}^i = \{t^1, t^2, \ldots t^i\}$, and, respectively, the first j spikes of $\tilde{\mathscr{F}}$, $\tilde{\mathscr{F}}^j = \{\tilde{t}^1, \tilde{t}^2, \ldots \tilde{t}^j\}$. $D_{i,\,j}$ is computed as:

$$D_{i,\,j} = \min\Big\{D_{i-1,\,j} + 1,\; D_{i,\,j-1} + 1,\; D_{i-1,\,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q)\Big\}. \tag{41}$$

The elements from which the induction starts are $D_{i,\,0} = i$ and $D_{0,\,j} = j$, because it is considered that $\mathscr{F}^0 = \tilde{\mathscr{F}}^0 = \emptyset$ (the distance between a train of i spikes and a train of no spikes is i because all spikes must be removed or correspondents for all must be inserted for a cost of 1 each). If there are n spikes in $\mathscr{F}$ and $\tilde{n}$ spikes in $\tilde{\mathscr{F}}$, the algorithm needs to use a $n+1$ by $\tilde{n}+1$ matrix that stores the $D_{i,\,j}$ values for the various i and j. The actual distance between the full spike trains is $D_{n,\,\tilde{n}}$, the element at the bottom right of the matrix. Because the computation of each $D_{i,\,j}$ element requires all the three values placed above, left and above left in the matrix, all the elements in the matrix have to be computed inductively.

The choice of the minimum of the three values performed at the computation of each element $D_{i,\,j}$ of the matrix (except the ones in the left and top edges of the matrix, $i = 0$ or $j = 0$) reflects an optimal choice of the status of the last spikes in the partial spike trains corresponding to the considered element. The optimal status of the last spikes depends on the structure of the pair of partial spike trains that precedes them. If the minimum is $D_{i-1,\,j}+1$, then the spike at t^i has a contribution of 1 to the distance and it is thus independent of any spike in the reciprocal spike train $\tilde{\mathscr{F}}^j$; the spike at $\tilde{t}^j$ may or may not be independent, as a function of the structure of the $(\mathscr{F}^{i-1}, \tilde{\mathscr{F}}^j)$ pair of spike trains. If the minimum is $D_{i,\,j-1} + 1$, then the spike at $\tilde{t}^j$ is independent of any spike in the reciprocal spike train $\mathscr{F}^i$; again, the spike at t^i may or may not be independent, as a function of the structure of the $(\mathscr{F}^i, \tilde{\mathscr{F}}^{j-1})$. If the minimum is $D_{i-1,\,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q)$ then the actual spike at t^i is linked to the target one at $\tilde{t}^j$ and will have to move towards it.

If more than one of the three values have the minimum value, then, at least theoretically, they might represent different, alternative choices of the optimal structure of the $(\mathscr{F}^i, \tilde{\mathscr{F}}^j)$ pair of spike trains. We will consider here that a pair of spikes $(t^i, \tilde{t}^j)$ is linked if and only if $D_{i-1,\,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q)$ is a strict minimum, i.e. it is the only one of the three choices that corresponds to the minimum value of $D_{i,\,j}$. If it is equal to another minimum, the link has just been broken and we will consider the alternative structure. If $D_{i-1,\,j} + 1$ and $D_{i,\,j-1} + 1$ are equal minima, they might correspond to different structures, as a function of the structures of $(\mathscr{F}^{i-1}, \tilde{\mathscr{F}}^j)$ and $(\mathscr{F}^i, \tilde{\mathscr{F}}^{j-1})$. However, if these structures involve pairs of linked spikes, it is extremely improbable that the equality $D_{i-1,\,j} = D_{i,\,j-1}$ will hold exactly, especially in a numerical computer simulation. The equality can hold with a non-vanishing probability when all spikes in $(\mathscr{F}^{i-1}, \tilde{\mathscr{F}}^j)$ and $(\mathscr{F}^i, \tilde{\mathscr{F}}^{j-1})$ are independent, in which case the two alternative structures for $(\mathscr{F}^i, \tilde{\mathscr{F}}^j)$ are actually identical, since they both consider that all the spikes are independent. Even in the improbable case that the equality holds when links do exist, for our purpose of supervised learning is is sufficient to consider only one of the alternatives, as long as we are consistent in the choice of this alternative.

It can be shown that, if $D_{i-1,\,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q)$ is a strict minimum value for computing $D_{i,\,j}$, then $\sigma(|t^i - \tilde{t}^j|/\tau_q) < 2$ and thus the two spikes are linked (not independent), as follows. The addition of a spike at t^i to the pair of spike trains $\mathscr{F}^{i-1}$ and $\tilde{\mathscr{F}}^{j-1}$ can increase the distance with at most 1, because in the worst case

```
Input: The pair of spike trains $\mathscr{F}$, $\tilde{\mathscr{F}}$; the parameter $\tau_q$; the function $\sigma$
Output: The distance between the spike trains and the structure of the spike
        trains corresponding to this distance
$D_{0,0} := 0$;
$S_{0,0} := \emptyset$;
$\tilde{S}_{0,0} := \emptyset$;
Set the left edge of the matrix
for $i := 1 \ldots n$ do
    $D_{i,0} := i$;
    $S_{i,0} := S_{i-1,0} \cup (i, \circ)$;
    $\tilde{S}_{i,0} := \emptyset$;
Set the top edge of the matrix
for $j := 1 \ldots \tilde{n}$ do
    $D_{0,j} := j$;
    $S_{0,j} := \emptyset$;
    $\tilde{S}_{0,j} := \tilde{S}_{0,j-1} \cup (j, \circ)$;
Perform the inductive computation
for $i := 1 \ldots n$ do
    for $j := 1 \ldots \tilde{n}$ do
        Compute $(D_{i,j}, (S_{i,j}, \tilde{S}_{i,j}))$
        $\varsigma := D_{i-1,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q)$;
        if $D_{i-1,j} \leq D_{i,j-1} \wedge D_{i-1,j} + 1 \leq \varsigma$ then
            Spike $j$ is independent
            $D_{i,j} := D_{i-1,j} + 1$;
            $S_{i,j} := S_{i-1,j} \cup (i, \circ)$;
            $\tilde{S}_{i,j} := \tilde{S}_{i-1,j}$;
        else if $D_{i,j-1} + 1 \leq \varsigma$ then
            Spike $i$ is independent
            $D_{i,j} := D_{i,j-1} + 1$;
            $S_{i,j} := S_{i,j-1}$;
            $\tilde{S}_{i,j} := \tilde{S}_{i,j-1} \cup (j, \circ)$;
        else
            Spikes $i$ and $j$ are linked
            $D_{i,j} := \varsigma$;
            $S_{i,j} := S_{i-1,j-1} \cup (i, j)$;
            $\tilde{S}_{i,j} := \tilde{S}_{i-1,j-1} \cup (j, i)$;
return $(D_{n,\tilde{n}}, (S_{n,\tilde{n}}, \tilde{S}_{n,\tilde{n}}))$;
```

Algorithm 1: The algorithm for computing the distance between two spike trains and the structure of these spike trains corresponding to the distance. The text in italics represents comments.

the spike will be removed for a cost of 1. We thus have

$$D_{i,\,j-1} \leq D_{i-1,\,j-1} + 1 \tag{42}$$

$$D_{i,\,j-1} + 1 \leq D_{i-1,\,j-1} + 2. \tag{43}$$

But if $D_{i-1,\,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q)$ is a strict minimum, then

$$D_{i-1,\,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q) < D_{i,\,j-1} + 1 \tag{44}$$

and from the last two equations we get

$$D_{i-1,\,j-1} + \sigma(|t^i - \tilde{t}^j|/\tau_q) < D_{i-1,\,j-1} + 2 \tag{45}$$

$$\sigma(|t^i - \tilde{t}^j|/\tau_q) < 2, \tag{46}$$

which was to be demonstrated.

The algorithm for computing the structure of the pair of spike trains $(\mathscr{F}, \tilde{\mathscr{F}})$ has to compute the structure inductively, along with the computation of the distance between the spike trains. We will thus have to store the structure of all pairs of partial spike trains $(\mathscr{F}^i, \tilde{\mathscr{F}}^j)$ for $i = 0 \ldots n$ and $j = 0 \ldots \tilde{n}$. This structure is defined by indicating for each spike whether it is independent or not; if it is linked (not independent), it will also have to indicate the index of the spike in the other train to which it is linked. The structure of $(\mathscr{F}^i, \tilde{\mathscr{F}}^j)$ is formed by the pair $(S_{i,\,j}, \tilde{S}_{i,\,j})$ where the first element is the structure information for $\mathscr{F}^i$ when used for computed $D_{i,\,j}$ and the second element is the structure information for $\tilde{\mathscr{F}}^j$ when used for computed $D_{i,\,j}$. More precisely, $S_{i,\,j}$ is a list in which each element indicates whether the corresponding spike k, with $k = 1 \ldots i$, is independent, which we denote through an element $(k, \circ)$; or whether the spike is linked to a spike l in the other spike train, which we denote through an element (k, l). The list $\tilde{S}_{i,\,j}$ has an analogous meaning for the spikes in the target spike train $\tilde{\mathscr{F}}^j$.

Algorithm 1 lists the entire procedure of computing the structure of the pair of spike trains along with the distance between them.

2.10 Average displacement for gaussian jitter

In Fig. 4, input spikes were displaced randomly around the reference timing according to a gaussian distribution with an amplitude ς. The probability density of a (positive or negative) displacement δt is

$$P(\delta t) = \frac{1}{\sqrt{2\,\pi\varsigma^2}} \exp\left(-\frac{\delta t^2}{2\,\varsigma^2}\right). \tag{47}$$

The probability density of a given displacement, in absolute value, is

$$P(|\delta t|) = 2\,P(\delta t). \tag{48}$$

The average displacement (in absolute value) is then

$$\overline{|\delta t|} = \int_0^\infty P(|x|)\, x \,\mathrm{d}x \tag{49}$$

$$= \int_0^\infty \sqrt{\frac{2}{\pi\varsigma^2}} \exp\left(-\frac{x^2}{2\,\varsigma^2}\right) x \,\mathrm{d}x \tag{50}$$

$$= \sqrt{\frac{2}{\pi}}\varsigma. \tag{51}$$

For $\varsigma = 5$ ms, we get $\overline{|\delta t|} \simeq 3.99$ ms.

3 Comparison to other results

The first supervised learning method for spiking neurons was SpikeProp (Bohte et al., 2000, 2002), a method inspired by the backpropagation algorithm used for training classical neural networks. SpikeProp works by minimizing the difference between the timing of an output spike and the desired timing. The first versions of the learning method required a feedforward network and that each neuron in the network fires only once during a trial. Later versions (Xin and Embrechts, 2001; Moore, 2002; Schrauwen et al., 2004; Booij and tat Nguyen, 2005; Tiňo and Mills, 2005, 2006; McKennoch et al., 2006) extended the method for including a momentum term; adapting the synaptic delays, time constants and neurons' thresholds during learning; for networks where the input (but not the output) neurons fire more than once per trial; for recurrent networks; and for improving learning speed under certain assumptions. However, the method is designed for adjusting just the timing of a single (first) spike per output neuron and assumes that the synapses are such that each output neuron fires at least one spike for the given inputs. The method is not suitable for adjusting the number of output spikes nor for training a network to fire given output spike patterns that extend in time.

Carnell and Richardson (2005) devised a method for modifying the synaptic weights such as the weighted sum of the presynaptic spike trains (in an algebraic representation) converge to a desired one. If the neuron model is such that the firing of the postsynaptic neuron is close to this weighted sum, then the method allows the supervised learning of a target output spike train. The method is quite original and general, but ignores the details of the dynamics of the postsynaptic potential and of the neuronal membrane.

Pfister et al. (2006) have derived supervised learning rules for probabilistic neurons. The learning method is based on gradient ascent in the space of synaptic efficacies, which maximizes the likelihood of having a trained neuron firing at the desired moments. Because of the probabilistic framework, the learning rules do not involve the actual timing of the output spikes, but the probability of having a particular output spike train given a particular input spike train. Calculating such a probability while taking into account the reset of the membrane potential after the spikes of the output neuron is computationally challenging and not biologically plausible.

Legenstein and colleagues (Legenstein et al., 2005; Legenstein and Maass, 2006) have studied a supervised, biologically-inspired learning method for spiking neurons that works by clamping neurons to the desired output and applying spike timing–dependent plasticity (STDP) to the afferent synapses of the trained neurons. Under certain conditions, after learning, the neurons yield the desired output even after the teaching signal is removed. The efficacy of this learning method has been proved analytically only for Poisson input spike trains, and there are worst case scenarios where the method fails, but simulations have shown that the method is effective in more general conditions. The method works only when synapses have hard bounds, by driving synaptic efficacies toward these bounds. Thus, the output patterns that this method can learn are restricted to those that can be generated by synapses that have either minimum (zero) or maximum efficacy. A similar rule is effective for supervised learning of patterns by networks (Gerstner et al., 1993), but not by single neurons.

The tempotron (Gütig and Sompolinsky, 2006) implements supervised learning for a particular task where an output neuron either fires one spike or does not fire, when presented with an input spike train. The approach assumes that after the neuron emits a spike in response to a input pattern all other incoming spikes have no effect on the neuron (are shunted), which is artificial. The method requires information that is nonlocal in time, needing to monitor the maximum of the output, and information that is not available to the neuron, such as the maximum of the membrane potential that would have been reached if the neuron would have not fired. The timing of the output spike cannot be controlled with this method. The tempotron has a binary response and thus its output cannot distinguish between more than two input categories. All these constraints undermine its biological plausibility and its applicability to more general problems. We have shown that the tempotron is equivalent to a particularization of the ReSuMe learning rule (Florian, 2008). A learning rule by Urbanczik and Senn (2009) improves the original tempotron learning rule but is still focused on the artificial tempotron setup.

Barak and Tsodyks (2006) have developed learning rules that increase the variance of the input current evoked by a set of learned patterns relative to that obtained from random background patterns. The trained neuron then has a larger firing rate when presented with one of the learned patterns, as compared to when presented with a typical background pattern. The learning rules are quite complex, with low biological plausibility. They allow a neuron to recognize input patterns, but the timing of the output spikes is not controlled by these rules. Other complex setups for recognizing spike patterns were also developed (Hopfield and Brody, 2001; Gers et al., 2002; Jin, 2004, 2008).

A few other supervised learning methods for supervised neural networks also exist but work only for some specific cases, such as population-temporal coding (Schrauwen and Van Campenhout, 2006), theta neurons (Voegtlin, 2007; McKennoch et al., 2009), neurons with very large membrane decay time constants and constant interspike intervals for the inputs (Kaiser and Feldbusch, 2007), networks with time to first spike coding for classification through plasticity of synaptic delays (Paugam-Moisy et al., 2007), neurons having two presynaptic and one postsynaptic spikes per learning cycle (Ruf and Schmitt, 1997), specific configurations, composed of several modules, of the trained network (Amin and Fujii, 2004).

ReSuMe (Ponulak, 2005, 2006*a*,*b*; Kasiński and Ponulak, 2005; Ponulak and Kasiński, 2006; Ponulak, 2008; Ponulak and Kasiński, 2010) is a general supervised learning method for spiking neurons that allows learning of arbitrary output spike trains. It is the only existing learning rule that is comparable to the ones introduced here. However, this learning rule has been conjectured without an analytical justification, by analogy to the Widrow-Hoff rule for analog neurons. To date, it has been shown analytically that ReSuMe will converge to an optimal solution only for the case of one input spike and one target output spike (Ponulak, 2006*a*). Simulations have shown that not all the terms of the conjectured learning rule are needed for learning (Ponulak, 2008). We have shown here (Fig. 5) that ReSuMe is less efficient than E-learning, leading to a lower capacity.

For a review of supervised learning methods for spiking neural networks, see Kasiński and Ponulak (2006).

References

Amin, H. H. and Fujii, R. H. (2004), Input arrival-time-dependent decoding scheme for a spiking neural network, *in* 'ESANN 2004 proceedings — European Symposium on Artificial Neural Networks', Bruges, Belgium, pp. 355–360.

Barak, O. and Tsodyks, M. (2006), 'Recognition by variance: learning rules for spatiotemporal patterns', *Neural Computation* **18**, 2343–2358.

Bohte, S. M., Poutré, H. L. and Kok, J. N. (2000), SpikeProp: Error-backpropagation for networks of spiking neurons, *in* M. Verleysen, ed., 'Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2000)', D-Facto Press, pp. 419–425.

Bohte, S. M., Poutré, H. L. and Kok, J. N. (2002), 'SpikeProp: Error-backpropagation for networks of spiking neurons', *Neurocomputing* **48**, 17–37.

Booij, O. (2004), Temporal pattern classification using spiking neural networks, Master's thesis, Informatics Institute, Faculty of Science, University of Amsterdam, Netherlands.

Booij, O. and tat Nguyen, H. (2005), 'A gradient descent rule for spiking neurons emitting multiple spikes', *Information Processing Letters* **95**(6), 552–558.

Carnell, A. and Richardson, D. (2005), Linear algebra for time series of spikes, *in* 'Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2005)'.

D'Haene, M., Schrauwen, B., Van Campenhout, J. and Stroobandt, D. (2009), 'Accelerating event-driven simulation of spiking neurons with multiple synaptic time constants', *Neural Computation* **21**(4), 1068–1099.

Florian, R. V. (2008), Tempotron-like learning with ReSuMe, *in* V. Kurkova, R. Neruda and J. Koutnik, eds, 'Artificial Neural Networks - Proceedings of the 18th International Conference (ICANN 2008), Prague, Czech Republic', Vol. 5164 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 368–375.

Gers, F. A., Schraudolph, N. N. and Schmidthuber, J. (2002), 'Learning precise timing with LSTM recurrent networks', *Journal of Machine Learning Research* **3**, 115–143.

Gerstner, W. (2001), A framework for spiking neuron models: The spike response model, *in* F. Moss and S. Gielen, eds, 'The Handbook of Biological Physics', Vol. 4, Elsevier Science, chapter 12, pp. 469–516.

Gerstner, W. and Kistler, W. M. (2002), *Spiking neuron models*, Cambridge University Press, Cambridge, UK.

Gerstner, W., Ritz, R. and van Hemmen, J. L. (1993), 'Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns', *Biological Cybernetics* **69**, 503–515.

Gütig, R. and Sompolinsky, H. (2006), 'The tempotron: a neuron that learns spike timing-based decisions', *Nature Neuroscience* **9**(3), 420–428.

Hopfield, J. J. and Brody, C. D. (2001), 'What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration', *Proceedings of the National Academy of Science of the USA* **90**(3), 1282–1287.

Jaeger, H. (2001), 'The echo state approach to analysing and training recurrent neural networks', *GMD Report 148, German National Research Center for Information Technology* .

Jin, D. Z. (2004), 'Spiking neural network for recognizing spatiotemporal sequences of spikes', *Physical Review E* **69**, 021905–1–021905–13.

Jin, D. Z. (2008), 'Decoding spatiotemporal spike sequences via the finite state automata dynamics of spiking neural networks', *New Journal of Physics* **10**, 015010.

Kaiser, F. and Feldbusch, F. (2007), Building a bridge between spiking and artificial neural networks, *in* 'Artificial Neural Networks – ICANN 2007', Vol. 4668 of *Lecture Notes in Computer Science*, Springer, Berlin / Heidelberg, pp. 380–389.

Kasiński, A. and Ponulak, F. (2005), Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks, *in* W. Duch, J. Kacprzyk, E. Oja and S. Zadrozny, eds, 'Proceedings of the 15th International Conference on Artificial Neural Networks: Biological Inspirations', Vol. 3696 of *Lecture Notes in Computer Science*, Springer, pp. 145–152.

Kasiński, A. and Ponulak, F. (2006), 'Comparison of supervised learning methods for spike time coding in spiking neural networks', *International Journal of Applied Mathematics and Computer Science* **16**(1), 101–113.

Legenstein, R. A. and Maass, W. (2006), A criterion for the convergence of learning with spike timing dependent plasticity, *in* L. K. Saul, Y. Weiss and L. Bottou, eds, 'Advances in Neural Information Processing Systems', Vol. 17, MIT Press, pp. 865–872.

Legenstein, R., Naeger, C. and Maass, W. (2005), 'What can a neuron learn with spike-timing-dependent plasticity?', *Neural Computation* **17**, 2337–2382.

Maass, W., Natschlaeger, T. and Markram, H. (2002), 'Real-time computing without stable states: A new framework for neural computation based on perturbations', *Neural Computation* **14**, 2531–2560.

McKennoch, S., Liu, D. and Bushnell, L. (2006), Fast modifications of the SpikeProp algorithm, *in* 'International Joint Conference on Neural Networks, 2006. IJCNN '06.', pp. 3970–3977.

McKennoch, S., Voegtlin, T. and Bushnell, L. G. (2009), 'Spike-timing error backprop in theta neuron networks', *Neural Computation* **21**(1), 9–45.

Moore, S. C. (2002), Back-propagation in spiking neural networks, Master's thesis, University of Bath, UK.

Paugam-Moisy, H., Martinez, R. and Bengio, S. (2007), A supervised learning approach based on STDP and polychronization in spiking neuron networks, *in* 'European Symposium on Artificial Neural Networks (ESANN 2007)'.

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 Computation* **18**(6), 1318–1348.

Ponulak, F. (2005), ReSuMe — new supervised learning method for the spiking neural networks, Technical report, Institute of Control and Information Engineering, Poznań University of Technology, Poland.

Ponulak, F. (2006*a*), ReSuMe — proof of convergence, Technical report, Institute of Control and Information Engineering, Poznań University of Technology, Poland.

Ponulak, F. (2006*b*), Supervised learning in spiking neural networks with ReSuMe method, PhD thesis, Poznań University of Technology, Poland.

Ponulak, F. (2008), 'Analysis of ReSuMe learning process for spiking neural networks', *International Journal of Applied Mathematics and Computer Science* **18**(2), 117–127.

Ponulak, F. and Kasiński, A. (2006), Generalization properties of SNN trained with ReSuMe, *in* 'Proceedings of the European Symposium on Artificial Neural Networks, ESANN'2006, Bruges, Belgium'.

Ponulak, F. and Kasiński, A. (2010), 'Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting', *Neural Computation* **22**(2), 467–510.

Ruf, B. and Schmitt, M. (1997), 'Learning temporally encoded patterns in networks of spiking neurons', *Neural Processing Letters* **5**(1), 9–18.

Schrauwen, B., and Van Campenhout, J. (2004), Extending SpikeProp, *in* 'Proceedings of the 2004 IEEE International Joint Conference on Neural Networks', Vol. 1, p. 475.

Schrauwen, B. and Van Campenhout, J. (2006), Backpropagation for population-temporal coded spiking neural networks, *in* 'Proceedings of the 2006 International Joint Conference on Neural Networks', pp. 3463–3470.

Schrauwen, B., Verstraeten, D. and Van Campenhout, J. (2007), An overview of reservoir computing: theory, applications and implementations, *in* 'Proceedings of the 15th European Symposium on Artificial Neural Networks', pp. 471–482.

Tiňo, P. and Mills, A. J. S. (2005), Learning beyond finite memory in recurrent networks of spiking neurons, *in* L. Wang, K. Chen and Y. Ong, eds, 'Advances in Natural Computation - ICNC 2005', Vol. 3611 of *Lecture Notes in Computer Science*, Springer, Berlin/Heidelberg, pp. 666–675.

Tiňo, P. and Mills, A. J. S. (2006), 'Learning beyond finite memory in recurrent networks of spiking neurons', *Neural Computation* **18**(3), 591–613.

Urbanczik, R. and Senn, W. (2009), 'A gradient learning rule for the tempotron', *Neural Computation* **21**(2), 340–352.

Victor, J. D. and Purpura, K. P. (1996), 'Nature and precision of temporal coding in the visual cortex: A metric-space analysis', *Journal of Neurophysiology* **76**(6), 1310–1326.

Victor, J. D. and Purpura, K. P. (1997), 'Metric-space analysis of spike trains: theory, algorithms and applications', *Network: Computation in Neural Systems* **8**, 127–164.

Voegtlin, T. (2007), Temporal coding using the response properties of spiking neurons, *in* B. Schölkopf, J. Platt and T. Hoffman, eds, ‘Advances in Neural Information Processing Systems 19’, MIT Press, Cambridge, MA, pp. 1457–1464.

Xin, J. and Embrechts, M. J. (2001), Supervised learning with spiking neural networks, *in* ‘Proceedings of the International Joint Conference on Neural Networks (IJCNN 2001)’, Vol. 3, pp. 1772–1777.