LETTER Communicated by Mark van Rossum

Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting

Filip Ponulak
ponulak@bcf.uni-freiburg.de; Filip.Ponulak@put.poznan.pl
Institute of Control and Information Engineering, Poznań University of Technology, Poznań 60-965, Poland, and Bernstein Center for Computational Neuroscience, Albert-Ludwigs University Freiburg, Freiburg 79-104, Germany

Andrzej Kasiński
Andrzej.Kasinski@put.poznan.pl
Institute of Control and Information Engineering, Poznań University of Technology, Poznań 60-965, Poland

Learning from instructions or demonstrations is a fundamental property of our brain necessary to acquire new knowledge and develop novel skills or behavioral patterns. This type of learning is thought to be involved in most of our daily routines. Although the concept of instruction-based learning has been studied for several decades, the exact neural mechanisms implementing this process remain unrevealed. One of the central questions in this regard is, How do neurons learn to reproduce template signals (instructions) encoded in precisely timed sequences of spikes?

Here we present a model of supervised learning for biologically plausible neurons that addresses this question. In a set of experiments, we demonstrate that our approach enables us to train spiking neurons to reproduce arbitrary template spike patterns in response to given synaptic stimuli even in the presence of various sources of noise.

We show that the learning rule can also be used for decision-making tasks. Neurons can be trained to classify categories of input signals based on only a temporal configuration of spikes. The decision is communicated by emitting precisely timed spike trains associated with given input categories. Trained neurons can perform the classification task correctly even if stimuli and corresponding decision times are temporally separated and the relevant information is consequently highly overlapped by the ongoing neural activity.

Finally, we demonstrate that neurons can be trained to reproduce sequences of spikes with a controllable time shift with respect to target templates. A reproduced signal can follow or even precede the targets. This surprising result points out that spiking neurons can potentially be applied to forecast the behavior (firing times) of other reference neurons or networks.

1 Introduction

Supervised learning was proposed as a successful concept of information processing in neural network already in the early years of the theory of neural computation (Rosenblatt, 1958; Widrow & Hoff, 1960; Widrow, 1962; Werbos, 1974). Recently, there has been an increasing body of evidence that instruction-based learning is also exploited by the brain (Knudsen, 1994). The most documented evidence for this type of learning in the central nervous system comes from studies on the cerebellum and the cerebellar cortex, and thus refers mostly to motor control and motor learning (Thach, 1996; Ito, 2000a; Montgomery, Carton, & Bodznick, 2002). In particular, supervised learning is believed to be utilized by the neural motor centers to form the internal representation of the body and the environment (Shidara, Kawano, Gomi, & Kawato, 1993; Kawato & Gomi, 1992a, 1992b; Miall & Wolpert, 1996; Wolpert, Miall, & Kawato, 1998) or for behavioral simulations and the encapsulation of learned skills (Doya, 1999). Learning from instructions is also supposed to control the representation of information in the sensory networks (Gaze, Keating, Szekely, & Beazley, 1970; Udin, 1985; Knudsen, 1991). It is likely that supervised learning also contributes to the establishment of networks that support certain cognitive skills, such as pattern recognition or language acquisition, although there is no strong experimental confirmation of the proposition (Knudsen, 1994; Thach, 1996; Ito, 2000b, 2008).

The instruction signals studied so far are believed to have a form of activity templates to be reproduced (Udin & Keating, 1981; Miall & Wolpert, 1996) or error signals to be minimized (Georgopoulos, 1986, Kawato & Gomi, 1992a; Montgomery et al., 2002). There is evidence that these signals are provided to the learning modules by sensory feedback (Carey, Medina, & Lisberger, 2005; Knudsen, 2002) or by other supervisory neural structures in the brain (Ito, 2000a; Doya, 1999). But how are the instructions exploited by the learning neural circuits? What is the exact neural representation of the instructive signals? And, finally, how do the biological neurons learn to generate the desired output given these instructions? Despite the extensive exploration of these issues, the exact mechanisms of supervised learning in the biological neurons remain unknown (for discussion, we refer to Knudsen, 1994; Lisberger, 1995; Ito, 2000a).

Whereas there is a well-documented and richly represented group of learning models for rate-based neurons (Kroese & van der Smagt, 1996; Rojas, 1996), spike-based coding schemes are still highly uncovered by the existing approaches. Only recently have several concepts been proposed to explain supervised learning in biologically realistic neuron models operating on the precise timing of particular action potentials (Kasiński & Ponulak, 2006). Most of these concepts, however, are physiologically implausible, since their applicability is either limited to the selected, analytically tractable spiking neuron models (Bohte, Kok, & Poutré, 2002; Booij

& Nguyen, 2005; Tiňo & Mills, 2005; Xin & Embrechts, 2001) or they make use of other mechanisms unavailable in the biological networks (Carnell & Richardson, 2005; Schrauwen & Campenhout, 2006).

Another limitation of many learning algorithms is that they offer a solution to specific information coding schemes only, such as single spike time coding (Bohte et al., 2002; Gütig & Sompolinsky, 2006, Belatreche, Maguire, McGinnity, & Wu, 2003), and thus they significantly reduce the richness of the possible neural information representations (Borst & Theunissen, 1999).

In this letter, we consider an alternative approach called ReSuMe (or remote supervised method) (Ponulak, 2005, 2006b). The algorithm is one of only a few existing supervised learning models able to code neural information in precisely timed spike trains. The method employs well-recognized physiological phenomena—it is based on the interaction between two spike-timing-dependent plasticity (STDP) processes (Kistler, 2002).

ReSuMe represents a spiking analogy to the classical Widrow-Hoff algorithm proposed for rate-based neuron models (Widrow & Hoff, 1960; Widrow, 1962). Similar to the Widrow-Hoff rule, ReSuMe minimizes the error between the target and output signals (here, the target and postsynaptic spike trains, respectively) without the need for an explicit gradient calculation. In this manner, it bypasses one of the fundamental limitations of the traditional gradient-based optimization methods in the domain of spiking neural networks (Bohte et al., 2002).

It has been demonstrated that ReSuMe enables effective learning of complex temporal and spatio temporal firing patterns with an arbitrary high accuracy (Kasiński & Ponulak, 2005; Ponulak, 2008). In Ponulak (2006a) we present a formal proof for convergence of the ReSuMe process at least for the case with input, output, and target patterns consisting of single spikes. The generalization property of spiking neurons trained with ReSuMe was discussed in Ponulak and Kasiński (2006a). In the same paper, it was demonstrated that spiking neurons trained with ReSuMe can learn any arbitrary transformation of input to output signals, limited only by the properties of the neuron and network dynamics (see also Ponulak, 2006b, for details).

Due to the ability of the method to operate online and its fast convergence, the method is suitable for real-life applications. This was confirmed in our initial simulation studies, which demonstrate that spiking neural networks (SNN) trained with ReSuMe become efficient neurocontrollers for movement generation and control (Ponulak & Kasiński, 2006b; Ponulak, Belter, & Kasiński, 2006; Ponulak, Belter, & Rotter, 2008; Belter, Ponulak, & Rotter, 2008).

In this letter, we provide a comprehensive overview of the properties of ReSuMe in the tasks often considered fundamental in neural computation. First, we consider a task of sequence learning. We demonstrate the ability of spiking neurons trained with ReSuMe to learn and reproduce complex patterns of spikes. We show that unlike many other supervised learning rules proposed for SNN, ReSuMe is not limited to the particular neuron

models and can even control the number of spikes in a burst of intrinsically bursting neuron models.

Next, we investigate the precision and reliability of the neural responses to highly noisy stimuli. We demonstrate that the appropriate learning procedure can substantially increase this reliability. As a result of our study, we identify some mechanisms by which the neurons can compensate for noise and uncertainty of the environment. We hypothesize that the identified mechanisms can potentially be exploited also by the biological neurons.

In another experiment, we explore the suitability of ReSuMe for classification tasks. We analyze the ability of spiking neurons trained with ReSuMe to discriminate among different temporal sequences of spikes. This is a task similar to the one discussed in Maass, Natschlaeger, and Markram (2002) and Gütig and Sompolinsky (2006). However, in contrast to those results, here we demonstrate that spiking neurons can communicate the classification decisions by the precisely timed spike sequences associated with the particular input categories.

Finally, we demonstrate that with a specific setting of the learning rule parameters, spiking neurons can be trained to reproduce target sequences of spikes with a controllable time lag, such that the reproduced signal follows or even precedes the target one. This observation has important consequences for possible applications of ReSuMe in forecasting tasks, where SNN-based adaptive models could predict the behavior of the reference neurons or networks.

Preliminary versions of some topics considered in this letter (derivation of the learning algorithm, independence of the algorithm of the neuron model, and spike shifting) have already been described in a doctoral thesis (Ponulak, 2006b) and presented at a conference (Kasiński & Ponulak, 2005). These results are reorganized and extended here and are provided for the sake of completeness and readability.

2 Learning Algorithm

The basic principle of supervised learning in artificial neural networks is that the adjustable parameters $\mathbf{w}$ associated with a given neuron o are modified to minimize the error between the target (y_d) and the actual neuron output (y_o). It is usually assumed that $\mathbf{w} = [w_{o1}, \ldots, w_{on}]^T$ refers to the efficacies of synaptic inputs $i = 1, \ldots, n$ converging onto the neuron.

In the case of SNN, where the signals y_d, y_o and the inputs $\mathbf{x} = [x_1, \ldots, x_n]^T$ are encoded by the timing of sequences of spikes, it is not so clear what exactly the representation of error signal should be and how this error should determine weight modifications $\Delta\mathbf{w}$.

In the following, we present an approach used in ReSuMe to address this problem. As a starting point, we consider the Widrow-Hoff rule defined for the ith synaptic input to neuron o as

$$\Delta w_{oi} = \alpha x_i (y_d - y_o), \tag{2.1}$$

where α is the learning rate; without loss of generality we assume here that $\alpha = 1$.

Let us now simply rewrite this rule as

$$\Delta w_{oi} = x_i y_d - x_i y_o. \tag{2.2}$$

By referring formula 2.2 to the observations of Hebb (1949) we can consider the right-hand side of equation 2.2 as a compound of the two Hebbian processes: the first one triggered by the correlation of the desired signal y_d with input x_i and the second one arising from the correlation between the actual output y_o and input x_i. Due to the minus sign in equation 2.2 the second term can be considered an anti-Hebbian process.

As in the Widrow-Hoff rule, we assume here that the instructive signal modulates the plasticity at the synaptic inputs to the trained neuron, but it has only a marginal or no direct effect on the postsynaptic somatic membrane potential.

Since we are interested in a learning rule that operates on the precise timing of spikes we rewrite equation 2.2 in the context of the spike-timing-based plasticity. We assume that the presynaptic, postsynaptic, and target signals x_i, y_o, y_d, are represented by spike trains, which we denote here by $S_i(t)$, $S_o(t)$, $S_d(t)$, respectively. Following Gerstner and Kistler (2002), we define a spike train as a sequence of impulses triggered by the particular neuron at its firing times: $S(t) = \sum_f \delta(t - t^f)$, where $f = 1, 2, \ldots$ is the label of the spike and $\delta(n)$ is a Dirac function with $\delta(t) = 0$ for $t \neq 0$ and $\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 1$.

The first term $(x_i y_d)$ on the right-hand side of equation 2.2 can be interpreted as an STDP process in which synaptic plasticity is triggered by the temporal correlation of the presynaptic spike train $S_i(t)$ and the instructive (target) signal $S_d(t)$. Let us denote this process by $X_{di}(t)$. By analogy, the second factor $(-x_i y_o)$ can be considered an anti-STDP process (Roberts & Bell, 2002; Kistler, 2002) defined over the pre- and postsynaptic spike trains. We denote it by $X_{oi}(t)$.

According to these considerations, we can reformulate equation 2.2 for the spiking neurons and for the continuous timescale as

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{oi}(t) = X_{di}(t) + X_{oi}(t). \tag{2.3}$$

In order to describe $X_{di}(t)$ and $X_{oi}(t)$ in a mathematical framework, we adopt a model of spike-timing-based synaptic plasticity proposed by Kistler and van Hemmen (2000) (see also Gerstner & Kistler, 2002). According to this model we can define $X_{di}(t)$ as

$$\begin{aligned} X_{di}(t) = a + S_i(t) \left[a_i + \int_0^{\infty} a_{id}(s)\, S_d(t-s)\, \mathrm{d}s \right] \\ + S_d(t) \left[a_d + \int_0^{\infty} a_{di}(s)\, S_i(t-s)\, \mathrm{d}s \right]. \end{aligned} \tag{2.4}$$

In equation 2.4 it is assumed that apart from the activity-independent weight decay ($a < 0$), the changes in the synaptic coupling $w_{oi}(t)$, resulting from the $X_{di}(t)$ process, can be triggered by either $S_i(t)$ or $S_d(t)$ even in the absence of spikes in the latter signal (cf. Roberts & Bell, 2002, or Bi, 2002). This relationship is described by the non-Hebbian terms a_i and a_d, respectively. We do not make any prediction about the values of a_d and a_i at the moment.

The weight changes can also be triggered by temporal correlation between the spikes in $S_i(t)$ and $S_d(t)$. These changes are attributed to the integral kernels $a_{id}(s)$ and $a_{di}(s)$, with s being the delay between the presynaptic and target firings ($s = t_i^f - t_d^f$). The kernels $a_{id}(s)$, $a_{di}(s)$ define the shape of a learning window (Gerstner & Kistler, 2002):

$$W_{di}(s) = \begin{cases} a_{di}(-s) = +A_{di} \cdot \exp(s/\tau_{di}) & \text{if } s \leq 0, \\ a_{id}(s) = -A_{id} \cdot \exp(-s/\tau_{id}) & \text{if } s > 0. \end{cases} \tag{2.5}$$

Parameters $A_{di}, A_{id} > 0$ are the amplitudes, and $\tau_{di}, \tau_{id} > 0$ are the time constants of the learning process. The selection of a shape of $W_{di}(s)$ given by equation 2.5 was inspired by data recorded in neurophysiological experiments (Bi & Poo, 1998).

Analogious to equation 2.4, the anti-STDP process $X_{oi}(t)$ can be described by the following formula:

$$\begin{aligned} X_{oi}(t) = b + S_i(t) \left[b_i + \int_0^\infty b_{io}(s)\, S_o(t-s)\, \mathrm{d}s \right] \\ + S_o(t) \left[b_o + \int_0^\infty b_{oi}(s)\, S_i(t-s)\, \mathrm{d}s \right]. \end{aligned} \tag{2.6}$$

Here s denotes a delay between the pre- and postsynaptic firing times, $s = (t_i^f - t_o^f)$. The learning window $W_{oi}(s)$ for $X_{oi}(t)$ is defined as

$$W_{oi}(s) = \begin{cases} b_{oi}(-s) = -A_{oi} \cdot \exp(s/\tau_{oi}) & \text{if } s \leq 0, \\ b_{io}(s) = +A_{io} \cdot \exp(-s/\tau_{io}) & \text{if } s > 0. \end{cases} \tag{2.7}$$

Let us now come back to the learning rule given by equation 2.3. We recall that this rule is supposed to modify the strength of the synaptic inputs such that the output spike trains $S_o(t)$ approximate the target patterns $S_d(t)$. One of the necessary conditions for this learning process to be stable is that the synaptic strengths are no more modified (i.e., $\mathrm{d}w_{oi}/\mathrm{d}t = X_{di}(t) + X_{oi}(t) = 0$) if $S_o(t) = S_d(t)$. In other words, $X_{di}(t)$ and $X_{oi}(t)$ must compensate each other whenever the trained neuron fires at exactly the desired times. This is satisfied when the parameters of $X_{di}(t)$ and $X_{oi}(t)$ meet the following

conditions:

$$\begin{cases} -b = a, \quad -b_i = a_i, \quad -b_o = a_d, \\ A_{oi} = A_{di}, \quad A_{io} = A_{id}, \\ \tau_{oi} = \tau_{di}, \quad \tau_{io} = \tau_{id}. \end{cases} \tag{2.8}$$

In section 3.3 we consider the case when not all of the conditions given by 2.8 are satisfied. We demonstrate some interesting phenomena arising from this fact. For now, however, let us assume equation 2.8. By substituting equations 2.4 and 2.6 into equation 2.3 and taking 2.8, we can rewrite the learning formula as

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{oi}(t) = S_i(t) \int_0^\infty a_{id}(s) \left[S_d(t-s) - S_o(t-s) \right] \mathrm{d}s + \left[S_d(t) - S_o(t) \right] \left[a_d + \int_0^\infty a_{di}(s)\, S_i(t-s)\, \mathrm{d}s \right]. \tag{2.9}$$

Equation 2.9 provides a mathematical description of the derived learning algorithm for sequence learning in spiking neurons.

In systematic studies on this rule, we found that it can lead to successful learning, but the kernel $a_{id}(s)$ contributes nothing to this success. This can be explained by noting that $a_{id}(s)$ is related to the anticausal order of spikes, so it does not modify inputs that contribute to the neuron state briefly before the target or output firing time, but rather those inputs that fire afterward. In fact, we observed that selecting $a_{id}(s) = 0$ for all $s \in \mathbb{R}$ guarantees better learning results and faster learning convergence (Ponulak, 2008).

Therefore, it seems reasonable to consider a modified learning rule by discounting the $a_{id}(s)$-related term. In this case, equation 2.9 reduces to the following formula:

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{oi}(t) \;=\; \left[S_d(t) - S_o(t) \right] \left[a_d + \int_0^\infty a_{di}(s)\, S_i(t-s)\, \mathrm{d}s \right]. \tag{2.10}$$

Equation 2.10 expresses the ReSuMe learning rule. This rule is illustrated in Figure 1.

It can be easily demonstrated that the role of the noncorrelative factor a_d in equation 2.10 is to adjust the average strength of the synaptic inputs so to impose on a neuron a desired level of activity, such that the actual mean firing rate of $S_o(t)$ approaches the mean firing rate of signal $S_d(t)$. Thus, the task of setting up the precise timing of spikes is attributed mainly to the Hebbian term $a_{di}(s)$. Results of our analysis performed in Ponulak (2008) show that training can be successfully performed even if $a_d = 0$. On the other hand, incorporation of this non-Hebbian term in ReSuMe speeds up the learning process significantly.

It appears that with the appropriate parameter settings, the ReSuMe rule given by equation 2.10 can be applied to both excitatory and inhibitory

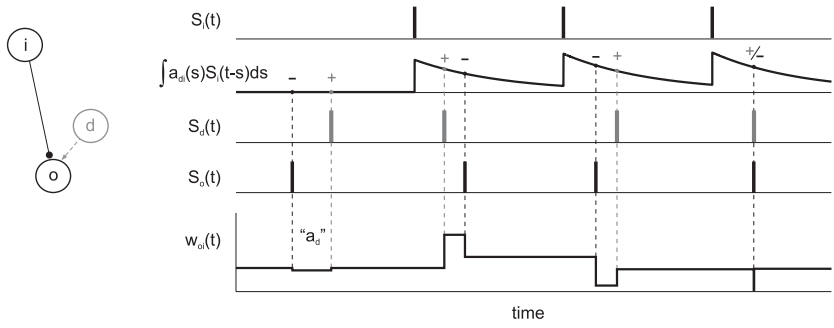

Figure 1: Illustration of the ReSuMe learning rule. For any excitatory synaptic connection from neuron i to neuron o, a synaptic strength w_{oi} is potentiated whenever a target spike ($S_d(t)$) is observed and depressed whenever the trained neuron fires ($S_o(t)$). Inhibitory synapses are driven by the opposite rule. The amount of the resulting weight change is defined by the nonassociative term a_d and by the kernel $\int a_{di}(s)\, S_i(t-s)\, \mathrm{d}s$. Potentiation and depression compensate each other whenever the trained neuron fires exactly at the target time.

synapses. Let us analyze both cases. For excitatory synapses, we require that the synapses contributing to the firing of a neuron at the desired times ($S_d(t)$) be potentiated, while other groups of excitatory synapses that evoke the neuron firing at undesired times (represented here by $S_o(t)$) be depressed. This is satisfied by equation 2.10 if $a_d > 0$ and $a_{di}(s) \geq 0$ for every s.

In contrast, the inhibitory synapses that prevent firing at the desired times are supposed to be weakened, and the inhibitory synapses that suppress the extra firing should be strengthened. Again, this can be implemented by equation 2.10 if $a_d < 0$ and $a_{di}(s) \leq 0$ for every s (i.e., according to equation 2.5, we require now that $A_{di} \leq 0$).

Alternatively, one could also think about another model, where a type of synapse is determined by the sign of its weight w (with positive values of w corresponding to excitatory synapses and negative values for inhibitory synapses). In this case, a synaptic strength is supposed to be given by $|w|$. Such a model, although far from biological realism, has proved to be a useful computational approach in artificial neural networks (Kroese & van der Smagt, 1996). In such a case, the learning rule given by equations 2.10 with parameters $a_d > 0$ and $a_{di}(s) \geq 0$ for every s can be applied directly to both types of synapses, since the opposite effect of ReSuMe on the excitatory versus inhibitory synapses is guaranteed by the sign of the variable w. Indeed, a positive weight change $\Delta w > 0$ resulting from ReSuMe will strengthen the excitatory synapses ($\Delta w, w > 0 \Rightarrow |w + \Delta w| > |w|$), and at the same time, it will reduce the strength of the inhibitory connections ($0 < \Delta w < |w|$, $w < 0 \Rightarrow |w + \Delta w| < |w|$). Consequently, a negative weight change will weaken excitatory connections while strengthening the inhibitory synapses, as required.

It is worth noting that the weight modifications induced by the learning rule can lead to changes of a sign of w and thus to transformation of excitatory to inhibitory synapses and vice versa. We observe that this property significantly improves the learning performance and the network capacity as compared to the case where the synaptic type is always preserved (Ponulak, 2006b).

We note also that equations 2.9 and 2.10 provide an alternative interpretation of the ReSuMe rule. Originally we assumed that the derived supervised learning rule can be considered a combination of two processes, X_{di} and X_{oi}, as described by equation 2.3. From equations 2.9 and 2.10, we see that the rule can also be interpreted as an STDP-like process relating the presynaptic spike trains with the error signal, where the error is represented simply as $[S_d(t) - S_o(t)]$. This conclusion is not surprising. Indeed we could intuitively expect such a learning rule by analogy to the common interpretation of the Widrow-Hoff rule as a Hebbian process defined over the input and error signals. On the other hand, it seems unlikely that any biological neuron could communicate the error signal of the form $[S_d(t) - S_o(t)]$. Alternatively, one could think about two distinct sources of $S_d(t)$ and $S_o(t)$ provided separately to the learning synaptic sites. This conclusion leads us, however, back to the initial assumption made in ReSuMe, according to which the target and output signals are processed individually.

Let us finally consider the issue of the learning process convergence for equation 2.10. Denote by D the whole time domain in which modifications of $w_{oi}(t)$ are observed. According to the assumptions made so far, we see that $[a_d + \int_0^\infty a_{di}(s)\, S_i(t-s)\, \mathrm{d}s] \neq 0$ for all $t \in D$. Therefore, the synaptic efficacy $w_{oi}(t)$ remains unmodified for the entire time domain D, that is, $\forall_{t \in D}\ \mathrm{d}w_{oi}(t)/\mathrm{d}t = 0$ if and only if $\forall_{t \in D}\ [S_d(t) - S_o(t)] = 0$. It means that equation 2.10 reaches a fixed point if the postsynaptic spike train $S_o(t)$ equals the target signal $S_d(t)$. It can be shown that under certain conditions, this fixed point is a global, positive attractor in a weight space (cf. Ponulak, 2006a).

3 Results

3.1 Learning Sequences of Spikes. In this section we present a set of experiments demonstrating that spiking neurons trained according to the ReSuMe algorithm are capable of learning and precisely reproducing arbitrary target sequences of spikes. We illustrate and analyze this process in section 3.1.1. Next, we show that ReSuMe is, to a great extent, independent of the neuron models and can be effectively applied to train various models of spiking neurons. In section 3.1.3, we focus on the noise issue and demonstrate that the precise spike timing paradigm can be considered a reliable mechanism for neural information coding even in the stochastic, noisy networks. Finally, we discuss how to extend the learning capabilities of the trained units by incorporating the ReSuMe method in a specific neural network architecture.

3.1.1 Analysis of the Learning Process. ReSuMe is a temporally local algorithm. The term *local* refers to the fact that at every time instance, the algorithm updates synaptic weights optimally for the nearest target firing times only (this is due to the exponentially decaying learning window). The global structure of an objective function, that is, the whole target firing pattern, is only to a limited extent relevant for the weight update at the given time. Despite this fact, ReSuMe proves to deal well with learning complex sequences of spikes. Here we illustrate and discuss this issue.

We first consider a simple case where a standard leaky integrate-and-fire (LIF) neuron with 400 synaptic inputs is trained on a random target firing pattern of the length 100 ms (see the appendix for details on the models and parameters used). The target is generated according to a homogeneous Poisson process with rate $r = 100$ Hz (hereafter we refer to the resulting spike trains as rHz Poisson spike trains). In this experiment, we assume that each synaptic input is allowed to fire only once during the single presentation of the target signal (this assumption is made here for illustrative reasons, which are explained below), and the particular input spikes are distributed uniformly throughout the 100 ms time interval. Synaptic strengths are initialized randomly according to a gaussian distribution. Initially all synaptic inputs are assumed excitatory. The learning is performed according to the ReSuMe rule given by equation 2.10. In this and in the following experiments, we assume that the learning rule is allowed to change the sign of the synaptic weights and that the negative weight values indicate inhibitory connections (see section 2).

Results of the experiment are presented in Figure 2a. The upper plot illustrates the evolution of the firing patterns generated by the neuron in the consecutive learning epochs. We observe that initially, the output pattern (open circles) differs from the target one (indicated by gray vertical bars) in both the mean firing rate and the firing times. However, after a few learning epochs, the extra spikes disappear, and the remaining spikes are gradually shifted toward the target times. In order to quantitatively evaluate the learning performance, we use a correlation-based measure C, which expresses the distance between the target and output trains (C is assumed 0 for uncorrelated spike trains and 1 for the perfectly matched firing patterns; see the appendix for details). The index C plotted as a function of the learning epochs m is shown in the bottom plot in Figure 2a. After 40 learning epochs, $C(m)$ approaches the value of 0.97, which corresponds to the case where all target firing times are closely matched by the trained neuron and only small fluctuations of the output spikes around the target times occur. These fluctuations vanish as we continue with learning.

It is instructive to see how the evolution of the neuron activity in the observed learning process is related to the adaptation of the particular synaptic inputs. In order to address this question, we compare the input synaptic weights before and after the training. The weights are shown in

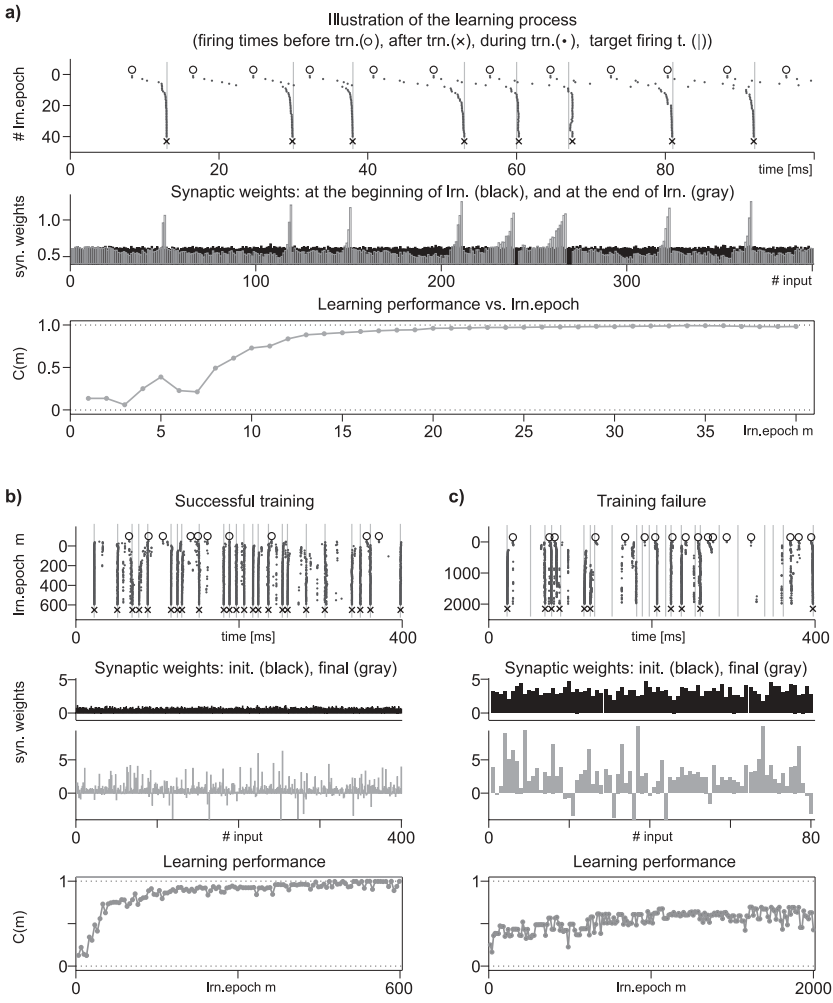

Figure 2: Illustration of the temporal sequence learning with ReSuMe. Three cases are considered. (a) A single LIF neuron with 400 synaptic inputs is trained on a 100 Hz Poisson target spike train of the length 100 ms. Every synaptic input is assumed to fire only once during the simulation time. The input spikes are uniformly distributed throughout the simulation time. The neuron successfully learns the task within around 20 learning epochs. (b) The same neuron is trained on another 100 Hz Poisson spike train of length 400 ms. This time, synaptic inputs are generated by a homogeneous 5 Hz Poisson process. The learning is slower but again successful—the output becomes highly correlated with the target pattern after around 200 epochs. (c) The learning performance drops significantly if the number of inputs is reduced to 80.

the middle graph in Figure 2a (black and gray bars for the initial and final values, respectively). The particular inputs are sorted chronologically according to their firing times in order to demonstrate how their temporal relationship with the target firing pattern determines the weight changes.

Since each input has been assumed in this experiment to fire only once, its strength is modified almost exclusively (if we ignore the contribution of the term a_d) by the local processes, that is, by the target or output spikes that occur in the temporal vicinity of the input spike. As expected, only inputs that fire shortly before the target firing times are strongly potentiated. We observe that the total amount of potentiation is an exponential function of the distance between the input spike and the following nearest target firing time. This reflects the exponential shape of the learning window used in ReSuMe. The remaining inputs—all inputs that do not fall into the learning windows associated with the particular target firing times—are slightly depressed. This can be explained by noting that initially, the neuron fired with a higher rate than desired and a global depression (due to the term a_d) was applied to adjust this parameter.

The resulting configuration of the synaptic weights observed in the experiment ensures that the neuron is strongly depolarized soon before the target spikes, while at other times, it remains hyperpolarized. This increases the likelihood of the neuron's firing at the desired times only.

In the scenario considered, where the inputs are supposed to fire once, each synapse is subject to a single local learning process; each synapse is optimized for just one target spike. Such a case is both simple for learning and convenient for analysis. Under certain conditions, the learning convergence can be guaranteed here (Ponulak, 2006a). Moreover, in this scenario, the local optimal solution is as good as the best possible solution obtained in the global optimization. On the other hand, the scenario where each input is allowed to fire only once has certain disadvantages, including: inefficient use of the available resources (relatively many synaptic inputs are required for the successful training) and a limited generalization capability. These two parameters are significantly improved in a scenario where the particular inputs transmit multiple spikes during the presentation of the target pattern. The price paid for it, however, is that the convergence cannot in general be guaranteed any more.[1]

We illustrate the case with multiple spikes per input in the next experiment. We again use the same LIF neuron model with 400 inputs as in the first experiment. This time, however, the neuron is trained to reproduce a 100 Hz Poisson spike train of length 400 ms and the input patterns are generated by a homogeneous Poisson process at rate 5 Hz. Since most of the inputs contribute now to multiple local learning processes, the training

[1]Conditions for the learning convergence and the analysis of the storage capacity of spiking neurons trained with ReSuMe are described elsewhere (Ponulak, 2009).

has to be performed more carefully. This is taken into account by choosing a relatively low learning rate (here selected to be 10 times lower than in the first experiment). In this way, we ensure that each local learning process is able to perform a local optimization while maintaining the interference with other local learning processes.

The results obtained here (see Figure 2b) demonstrate that ReSuMe deals with this task quite well. The performance index C increases from the initial value $C(0) = 0.18$ to around 0.9 within the first 150 epochs (see Figure 2b, bottom graph). As the training continues, C eventually reaches 0.95 after the next 450 epochs and fluctuates around this value in the subsequent learning epochs (the fluctuations reflect the single extra or missing spikes occasionally observed in the output signal; cf. Figure 2b, upper graph, black dots).

In a second variant of this experiment, we investigate how the learning performance is affected if we significantly reduce the number of inputs—here reduced by a factor of 5. The typical results obtained in this case are presented in Figure 2c. We observe that the neuron fails to learn the task: the performance measure reaches a value of only around 0.55 after 2000 learning epochs, and no further improvement is observed if we continue with training. The reason is that a reduced number of inputs implies fewer adjustable parameters as well as fewer postsynaptic potentials to trigger spikes at certain target times. Note, however, that unlike in the global learning methods, here the whole learning process is not affected; around half of the target spikes are precisely reproduced at the neuron output. This means that the learning algorithm makes efficient use of the available resources even if these are very limited.

Let us finally consider synaptic weight changes in relation to the learning process (see Figures 2b and 2c, middle graphs; note that this time, the inputs are not sorted chronologically). In contrast to the first experiment, the analysis of weights is more difficult here. This is due to the (usually conflicting) influence of many different local learning processes on each single synapse. Generally we can state only that the more correlated the given input with the target pattern is, the more potentiated it is expected to become. From the plots of the synaptic weights in Figures 2b and 2c, we also see that certain synaptic inputs have been converted from excitatory to inhibitory synapses (initially positive synaptic weights have become negative). Our observations indicate that this is often the case when fast, strong hyperpolarization is necessary to suppress spurious neuron firing at undesired times or to delay depolarization peaks.

3.1.2 Learning with Various Neuron Models. Many supervised learning methods that have been introduced for SNN are restricted to work only with analytically tractable neuron models, such as with a spike response model (Gerstner & Kistler, 2002). This constraint applies to all methods that explicitly refer to the specific neuron dynamics in the learning rules (Bohte et al., 2002; Booij & Nguyen, 2005; Tiňo & Mills, 2005; Xin &

Embrechts, 2001). In contrast, ReSuMe updates synaptic weights based on the correlation between the spike times only. This suggests that the method should work with various neuron models. Indeed, our analysis performed in Ponulak (2006a) and Kasiński and Ponulak (2005) confirms that ReSuMe can be applied to a broad class of spiking neurons. Here we present an experiment that complements this analysis. We not only demonstrate that the algorithm is independent of the neuron models, but also show that ReSuMe is able to control the intrinsic properties of neurons such as their bursting and nonbursting behaviour.

In the experiment, we compare results of learning for the LIF, Hodgkin-Huxley (HH), and Izhikevich (IM) neuron models (Gerstner & Kistler, 2002; Hodgkin & Huxley, 1952; Izhikevich, 2003). These three models are selected as representative for the different types of neuron dynamics (see the appendix for details on the models). The general scheme of the experiment is similar to that in section 3.1.1. The neurons are trained to reproduce target sequences of spikes in response to the given set of presynaptic spike patterns (see Figure 3). Training is performed according to the learning rule given by equation 2.10. We use a homogeneous 20 Hz Poisson process to generate the input and target patterns. In order to compare the results obtained for the different neuron models, we assume the same number of synaptic inputs (here $n = 500$), the same initial gaussian distribution of synaptic weights, and the same target and input patterns for all three neurons.

The typical results of training are shown in Figure 3c. We present the traces of the membrane potential and the firing times of the particular neurons after 30 learning epochs. We observe that the spike trains generated by all three neurons are almost indistinguishable from the target pattern (see Figure 3c, upper graph). The maximal absolute shift errors (see the appendix) between the corresponding spikes in target and output patterns are 0.5 ms, 4 ms, and 1 ms for LIF, HH, and IM, respectively. In all three cases, an average value of the absolute shifts error does not exceed 0.43 ms. In many applications, such errors are negligible as compared to 20 ms of the minimal interspike interval in the target signal.

To make the presented results statistically more reliable, we repeated the training procedure for 40 sets of input target patterns. The average dynamics of the learning process is illustrated in Figure 3b. We again use the measure $C(m)$ to express the distance between the target and output train for every learning epoch m. The round markers in the graph indicate the mean values and the vertical bars represent a standard deviation of $C(m)$ calculated over all training patterns. Analysis of this graph confirms that the learning process converges quickly for all neuron models, and after 30 learning epochs, the generated sequences of spikes become highly correlated with the target pattern: $C(m \geq 30) > 0.9$ for all three neuron models. Such high values of $C(m)$ indicate that on average, all target spikes are reproduced by the particular neuron models, and the actual firing times differ only slightly from the target ones, as illustrated in the examples in Figure 3c.

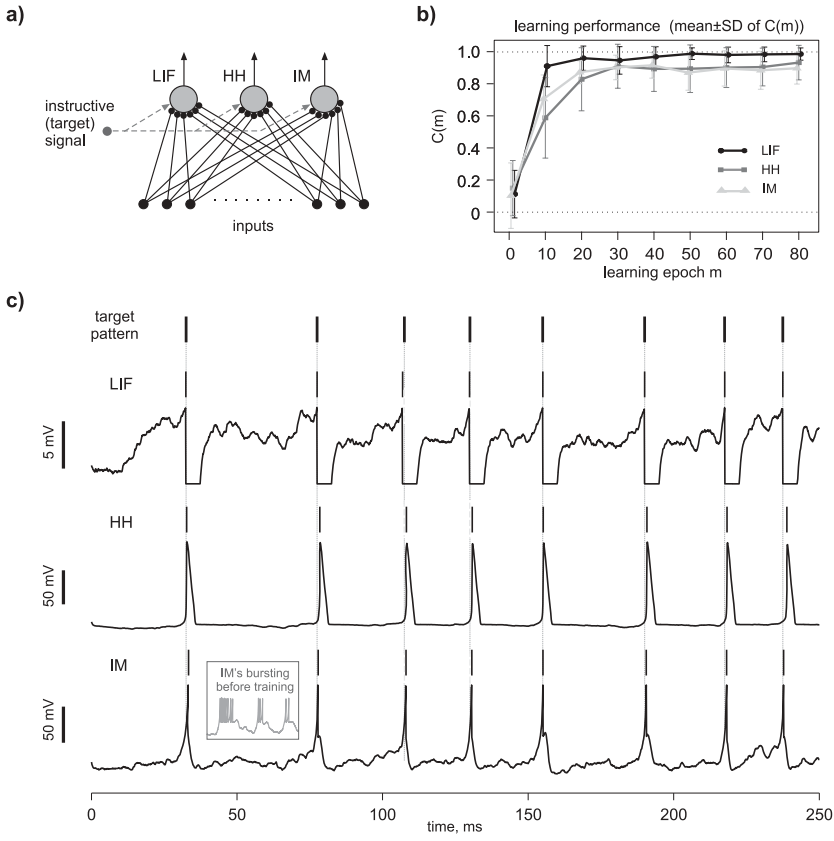

Figure 3: Spiking neurons trained with ReSuMe are able to precisely reproduce arbitrary sequences of spikes. Here illustrated with LIF, Hodgkin-Huxley (HH), and Izhikevich (IM) models. (a) All three neurons, driven by the common set of presynaptic stimuli, are trained on the common target signals. (b) Learning performance measure $C(m)$ calculated after every learning epoch m is averaged over 40 pairs of input-target (Poissonian) firing patterns. The dynamics of the learning process indicates fast convergence of ReSuMe. (c) Typical results of training for LIF, HH, and IM models, respectively. Generated spike trains and membrane potential traces are observed after 30 learning epochs. The resulting firing patterns closely resemble the target one. Note that IM is a bursting neuron (see the inset in the bottom panel). ReSuMe changes its behavior to regular spiking by suppressing all but the first spike in the particular bursts.

We note also that the fastest convergence is achieved for the LIF neuron. The possible source of the slower convergence in case of the HH and IM neuron models is their complex dynamics. For instance, the Izhikevich neuron model used in our experiments exhibits the bursting properties

with a strong tendency to fire several spikes in each burst (see Izhikevich, 2003). In this case, the ReSuMe algorithm not only has to set the initial regions of the potential bursts at the target firing times, but also has to suppress all but the first spike in each burst.

3.1.3 Learning in the Presence of Noise and Uncertainty. In the experiments considered in the previous sections, we assumed deterministic models of synapses and neurons as well as noise-free conditions for learning and pattern retrieval. Under these assumptions, the trained spiking neurons can reliably reproduce target sequences of spikes whenever the corresponding stimuli are presented. The reliability of the neural responses can, however, be significantly disturbed by noise. Such a disrupting influence of noise on the timing accuracy and reliability has been considered in many studies (Rieke, Warland, de Ruyter van Steveninck, & Bialek, 1997; Schneidman, 2001; van Rossum, O'Brien, & Smith, 2003). On the other hand, several experimental results provide evidence that the nervous system can employ some strategies to deal with noise to produce accurate and reliable responses (Mainen & Sejnowski, 1995; Shmiel et al., 2005; Montemurro et al., 2007; Tiesinga, Fellous, & Sejnowski, 2008).

Here we consider a similar issue in the context of ReSuMe training. We show that spiking neurons can precisely and reliably reproduce target sequences of spikes even under highly noisy conditions. We illustrate this property in the following computational experiment. Consider a single LIF neuron with multiple synaptic inputs ($n = 400$). The neuron is trained on a set of 20 pairs of input and target spike patterns. All patterns are generated independently according to a 20 Hz Poisson process. After training, the reliability of the target recall is tested against background noise simulated by a gaussian white noise current I_{ns} injected to the neuron. The mean value of I_{ns} is assumed zero, but its variance σ_I is systematically increased in the range of $[0,\ 30]$ nA.

For each value σ_I, a measure $C(\sigma_I)$ of a distance between the target and the observed output train is calculated. The experiment is performed according to two scenarios: the neuron is trained under the noiseless conditions, or the neuron is trained in the presence of noise of the same characteristics as the one used in a testing phase but with the variance of up to 20 nA.

The plots of $C(\sigma_I)$ obtained in both cases are depicted in Figure 4a (in black for the noisy training and in gray for the deterministic training). The round markers indicate the mean values, and the vertical bars represent the standard deviations of C over all training patterns.

In the first scenario (after deterministic training), we observe that the correlation $C(\sigma_I)$ drops almost linearly with the variance of noise. In contrast, the neuron is significantly less sensitive to noise if the noisy training is performed before and the amplitude of noise used during testing is not higher than the one used at training. In this second case, the correlation $C(\sigma_I)$ takes

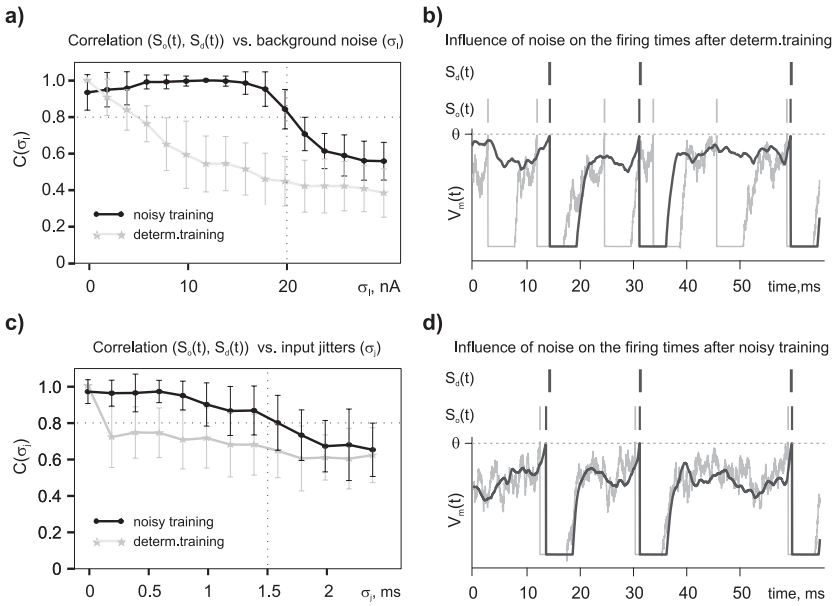

Figure 4: Spiking neurons can be trained to reliably reproduce target sequences of spikes even under noisy conditions. Precision and reliability of neural responses are investigated against (a) background white noise and (c) input spike jitters. Reliable responses to the given noise type are observed only if the same noise source was used also during training (noisy training); otherwise, (deterministic training) neural responses are affected by noise of even relatively low amplitude. The influence of background noise on the neuron's firing patterns is illustrated for (b) deterministic training and (d) noisy training. One of the strategies employed by the neurons trained under noisy conditions to improve neural reliability against the background noise is to move an operating point of the membrane potential farther away from the threshold value (θ) and to significantly strengthen only these excitatory inputs that contribute to the neuron firing at the desired times.

values between 0.9 and 1 for all σ_I in the range $[0,\ 20]$ nA and drops only for the noise variance beyond this range. At the same time, we observe that the variance of $C(\sigma_I)$ is in general much lower for the second scenario than the first. These results confirm that the noisy training enables the neuron to precisely and more reliably reproduce target firing patterns even for the relatively high level of noise. A similar beneficial effect of noisy training on the neural computation is already known from artificial nonspiking neural networks, where it has been shown that random noise added to a training set can improve not only training quality and fault tolerance but also the generalization capabilities of neural networks (Hertz, Krogh, & Palmer, 1991).

It is tempting to ask what neural mechanisms are employed here that make a neuron robust to noise. Before we answer this question, we note a rather obvious fact: the probability that the fluctuating current would trigger an extra spike in a neuron increases if the membrane potential gets closer to the threshold value. Therefore, to avoid spurious firing, the operating point of the neuron should be kept far away from the threshold value at all times when firing is undesired. On the other hand, to make sure that the neuron will fire around any given target time t_d, the synaptic excitatory inputs active just before t_d should be strong enough to absorb the possible hyperpolarizing influence of noise.

To observe whether this is indeed the case in our experiment, let us trace the membrane potential trajectories and the firing times of the considered neuron in both scenarios.

In Figure 4b, we illustrate the first case: the behavior of a neuron trained under deterministic conditions. We see that the neural response that resembles the target signal very precisely in the absence of noise (in black) changes dramatically after noise is added to the neuron (gray trace). When the training procedure is repeated, however, this time in the presence of noise (see Figure 4d), the trained firing pattern becomes robust to noise, so the particular firing times recorded in the noiseless test (in black) are shifted only slightly after the noise is added (gray). Comparison of the membrane potential trajectories observed after the noisy and deterministic training reveals that in the first case (see Figure 4d, black line), the operating point of the neuron is on average much lower than in the case of deterministic training (see Figure 4a) and the membrane potential gets close to the threshold only shortly before the firing times. This observation confirms our theoretical prediction.

It is worth emphasizing that this simple mechanism is able to cope with noise of relatively high amplitude. For example, in the experiment, $I_{ns}(t)$ evokes fluctuations of the membrane potential with amplitudes of up to 20% of the whole V_m range.

Another experiment was carried out to test the reliability of the trained neuron against unreliable stimuli. In this experiment, we considered 20 Hz Poisson spike templates for input and target signals and investigated neural responses to stimuli that varied from trial to trial. Variability of the input patterns was simulated by randomly shifting ("jittering") the firing times of the particular spikes. The jitter intervals were randomly drawn from the gaussian distribution with mean 0 and variance $\sigma_j \in [0,\ 2.5]$ ms. In addition, some spikes were randomly cancelled (with probability 0.1) or added (at the times generated by a 2Hz homogeneous Poisson process). Under these conditions, we investigated the correlation C between the target and output spike trains in two cases: after the training was performed with the varying inputs (with variance up to 1.5 ms) or when the inputs were fully reliable during the training.

The resulting plots of $C(\sigma_j)$ are presented in Figure 4c. We again observe that in the case of deterministic training (results drawn in gray), the neural

responses at the testing phase are highly sensitive to the input variability even for small values of σ_j. In contrast, the firing patterns trained under noisy conditions (drawn in black) remain highly correlated with the target as long as the stimulus variability does not exceed the one used during the training.

Again, we are interested in the neural mechanisms employed here to increase the reliability of neural responses. For this reason, we analyzed jointly the V_m traces and the synaptic weight distributions in the trained neuron. We found that the average amplitude of the excitatory postsynaptic potentials (EPSPs) contributing to the spike generation was much lower in the case of noisy training as compared to deterministic training. Consequently, many more EPSPs were required to evoke spikes in the trained neuron, and the significance of the individual EPSPs was reduced in favor of whole groups of EPSPs. Since the jitters in the input signals are generated according to a gaussian distribution with zero mean, the integration over many input impulses, which takes place in the trained neuron, results in statistically lower jitters of the output spikes as compared to the individual input spikes. This effect could be considered as one of the possible mechanisms to reduce spike timing variability propagated along neural networks and to increase the reliability of the temporal patterns generated in the networks.

It should be noted that in the experiments considered, the neuron trained under noisy conditions demonstrates high robustness to noise only in response to the stimuli used during the training. At the same time, the neuron responds highly unreliably to other stimuli. Our findings are consistent with the experimental results, which indicate that the same neuron may have very accurate spike timing in response to one stimulus and unreliable spike timing for another one (Schneidman, 2001).

The mechanisms improving neural reliability against noise, identified in our experiments, are simple but effective. These mechanisms seem to be not necessarily attributed to the particular properties of the ReSuMe learning model, and thus they are likely to arise also while using learning rules possibly employed by the biological neural networks.

3.1.4 Long-Term Temporal Integration. We demonstrated that single neurons are able to learn relatively complex sequences of spikes. Yet computational and memory capacities of single neurons have obvious limitations that are imposed mostly by a limited number of optimization parameters, short integration time constants, and the fixed dynamics of the neurons. These limitations constrain the repertoire of behaviors that can be reproduced by the single neurons.

In this section we demonstrate that by using a network of spiking neurons, we can significantly increase the learning capabilities of a neural system.

In our approach, we exploit the concept of reservoir computing (Jaeger, 2001; Maass et al., 2002; Jaeger, Maass, & Principe, 2007). The typical

reservoir network consists of a randomly generated, large recurrent neural structure (a reservoir or a liquid), and a set of output units (readouts). The recurrent network acts as a filter projecting the inputs into a higher-dimensional vector of spike trains—a so-called reservoir state. The spike trains of all network neurons are integrated in the output units, so each component of the reservoir state reflects the impact that particular neurons may have on the readouts. The main advantage of this concept is that it makes use of the computational capabilities of recurrent networks while it significantly facilitates their training, since the desired network output is obtained by training connections terminating at the readouts only. In this case, there is no need for retrograde communication of error, and local learning rules, such as ReSuMe, can be successfully applied here.

In the experiment, we consider a network with a single input, a single output, and a reservoir structure consisting of 800 neurons. All neurons are modeled with the LIF units (see the appendix for details on the network parameters).

In order to better illustrate the computational advantage of the neural networks over single neurons, let us assume that the input signal is encoded by the timing of a single spike only. In contrast, the output patterns consist of multiple spikes, and each target output pattern has its individual temporal structure. The network is trained to reproduce the target pattern i whenever the input spike occurs at a corresponding time $t = t_i^{in}$, with $i = 1, 2$. The input and target patterns are illustrated in Figure 5a. The corresponding reservoir activity is presented in Figure 5b (in gray for pattern 1 and in black for pattern 2). We assume that the reservoir is deterministic and reliable, and is not subjected to noise. Thus, in the particular runs, the network is supposed to exhibit the same spontaneous activity if initialized with the same state (note the same firing patterns of the reservoir before the first stimuli onset, i.e., within the time range $t \in [0, t_1^{in}]$; see Figure 5b).[2] After the stimuli onset, the network state starts to evolve along the individual trajectories corresponding to the particular input signals. In the raster plot in Figure 5b, this is manifested by the discrepancy between the response of the particular neurons to input patterns 1 or 2. We observe that the firing times of the corresponding spikes in response to patterns 1 or 2 often differ only by submilliseconds, but this is sufficient to change the whole state of the network observed by the readout neuron, so the readout can learn to generate clearly different output targets.

[2]Nonidentical initial conditions, unreliable input signals, or a small amount of noise added to the reservoir will in fact result in deviations of the network behavior from trial to trial, which can disrupt the learning process. This effect can, to some extent, be compensated by the appropriate training of the output neurons, so the readouts become able to absorb some fluctuations of the reservoir trajectory and still produce the desired output signal. We consider such a case for noisy input signals in section 3.2.

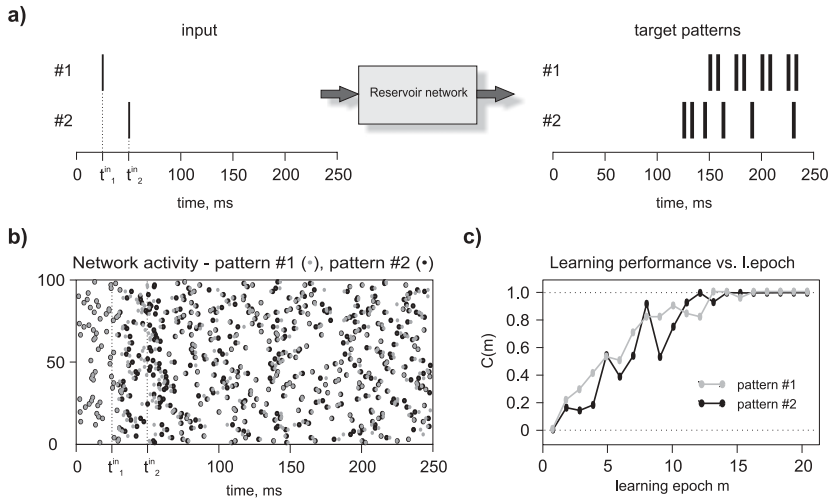

Figure 5: The repertoire of input-to-output transformations reproducible by spiking neurons can be significantly extended by employing the concept of reservoir networks. (a) In the example, a reservoir network is trained to reproduce target sequences of spikes in response to the input signal encoded in the firing time of a single spike only. The input and target output patterns are separated in time by hundreds of milliseconds (which extends beyond the time constants of the used neuron models by the factor of 20); moreover, the later input spike is supposed to trigger earlier responses. (b) The reservoir projects the input patterns into the high-dimensional network state, defined by the activity of particular neurons belonging to the network (here, drawn as spike raster plots of 100 reservoir neurons). Each trajectory of the network state is assumed to uniquely represent the particular input signal. (c) The network can simultaneously learn both pairs of input-target signals within several learning epochs.

The progress of learning is illustrated in Figure 5c. We observe that for both patterns, the metric C reaches the value of one after 16 learning epochs and remains there as we continue with the training.

Several aspects of the experiment make this task impossible with a single neuron. The first issue concerns the infeasibility of training a single neuron to generate any arbitrary predefined sequence of impulses in response to a single presynaptic spike only. In the experiment, however, the task is even more difficult since there are several different target patterns to be generated in response to the single spike fired at the different time instances.

Another factor limiting the computational capability of a single neuron is its relatively short integration time constant. As a result, the relationship between the stimulus and the corresponding neural response is defined

on a short timescale. In particular, the maximum delay between the offset of the stimulus and the response onset in the neuron models used in our simulations is limited to several milliseconds. In contrast, the reservoir, as a recurrent network, can extend the signal integration timescale to tens or hundreds of milliseconds as illustrated in Figure 5a (see also Maass & Markram, 2003).

We note also that the training pairs presented in Figure 5a are defined in such a way that the late presynaptic spike is supposed to trigger earlier response (compare pattern 2 with 1). This is yet another task that can hardly be handled by most of the single neuron models.

The experiment considered here illustrates some advantages of using networks of spiking neurons over single neurons. More examples of computing with the reservoirs in the context of spike sequence learning are presented in Kasiński and Ponulak (2005). For the systematic quantification of the memory capacity and the learning capabilities of the reservoir networks and the ReSuMe method, we refer to Ponulak (2006b).

3.2 Classification. In the past decade, spiking neural networks have been successfully applied to various classification tasks (Bohte et al., 2002; Ghosh-Dastidar & Adeli, 2007; Gütig & Sompolinsky, 2006; Hopfield & Brody, 2001; Maass et al., 2002; Maass, Natschlaeger, & Markram, 2004; Muresan, 2002).

In most cases, however, the neural classifiers communicated decisions by firing single spikes only (Bohte et al., 2002; Ghosh-Dastidar & Adeli, 2007; Gütig & Sompolinsky, 2006) or by using analog or binary (but not spiking) signal representation (Maass et al., 2004; Muresan, 2002; Hopfield & Brody, 2001). None of the used SNN-based classifiers considered so far demonstrated the ability to represent the classified categories by the associated sequences of precisely timed spikes. Here we show for the first time that this is possible with ReSuMe.

We illustrate this ability in a classification task proposed by Maass and colleagues (Maass et al., 2002; Natschlaeger, Markram, & Maass, 2003), where a spiking neural network is supposed to categorize the particular jittered segments of the input spike trains. This specific task is selected to demonstrate the classification abilities of spiking neurons trained with ReSuMe since it covers two interesting issues related to the properties of biological neural networks. First, it tests the ability of the neural classifiers to make the decision about input categories based on only the timing of spikes in the input patterns, and at the same time it explores the robustness of the classifier to the variability of spike timing (for a similar analysis in biological systems, see Montemurro et al., 2007; Furukawa & Middlebrooks, 2002). Second, it also tests the temporal integration ability of the neural circuits required for the discrimination of inputs on the timescale of hundreds of milliseconds, that is, on the timescale suitable for many cognitive and sensory-motor processes (Mauk & Buonomano, 2004).

In our experiment, we assume that the input is provided to the network through a single neuron, and all information of the input is stored only in the temporal configuration of spikes. In addition, we assume that each input pattern presented to the network consists of four segments (S1–S4) of length 250 ms. For the particular segments, two individual Poisson spike templates (T1,T2) are selected. The actual input spike trains of the length 1000 ms used for training and testing are generated by choosing for each segment one of the two associated templates (independently for each segment) and then generating their noisy version by moving each spike by an interval drawn from a gaussian distribution with mean 0 and variance varying in the particular experiment in the range of 1 to 4 ms (see Figure 6a; we refer also to Maass et al., 2002, for details).

The neural network used in our experiment consists of a single input: a deterministic, noise-free reservoir with 800 LIF neurons and 4 LIF readouts (see the appendix for the details on the reservoir structure).

The task of a readout i ($i = 1, 2, 3, 4$) is to classify the ith segment of the input signal by generating a firing pattern associated with a class of the template, from which the given segment was drawn. To make the task even more difficult, the target patterns for all readouts are defined in such a way that the target spikes occur not earlier than at 800 ms after the input onset. Thus, for example, the readout associated with segment S1 has to make a decision about a category of the first segment at least 500 ms after the end of the segment. In this case, the trace left by S1 in the firing activity of the recurrent circuit is subsequently overwritten by the next segments of the input spike train, and the readout has to extract from the composite signal only the information relevant to the corresponding segment. In contrast, the readout associated with the last segment of the input signal has to make a classification decision based on only partial knowledge about the segment, since the readout is expected to start firing before the end of the input pattern. In some cases, the temporal relationship between the last input segment and the corresponding target pattern is such that the readout can observe only a several-millisecond fragment of the segment before it has to make the decision.

In the experiment, the readouts were trained on a set of 100 input patterns and tested on 200 other presynaptic spike trains. The typical spike time histograms (STH) of the target and the corresponding output patterns recorded from the readout S1 in the testing phase are presented in Figure 6b (in black and gray, respectively). Note the jitters of the spike times in the target patterns introduced here to increase the robustness of the readouts to the fluctuations of the input signal (see section 3.1.3). We observe that the STH for output overlaps with the target to a high degree. This demonstrates that the target patterns are precisely and reliably reproduced by the readout. The results are presented for the input and target jitters with a variance of 2 ms.

In order to evaluate system performance in the classification task, we need to define the classification criteria. In our experiment, we assumed

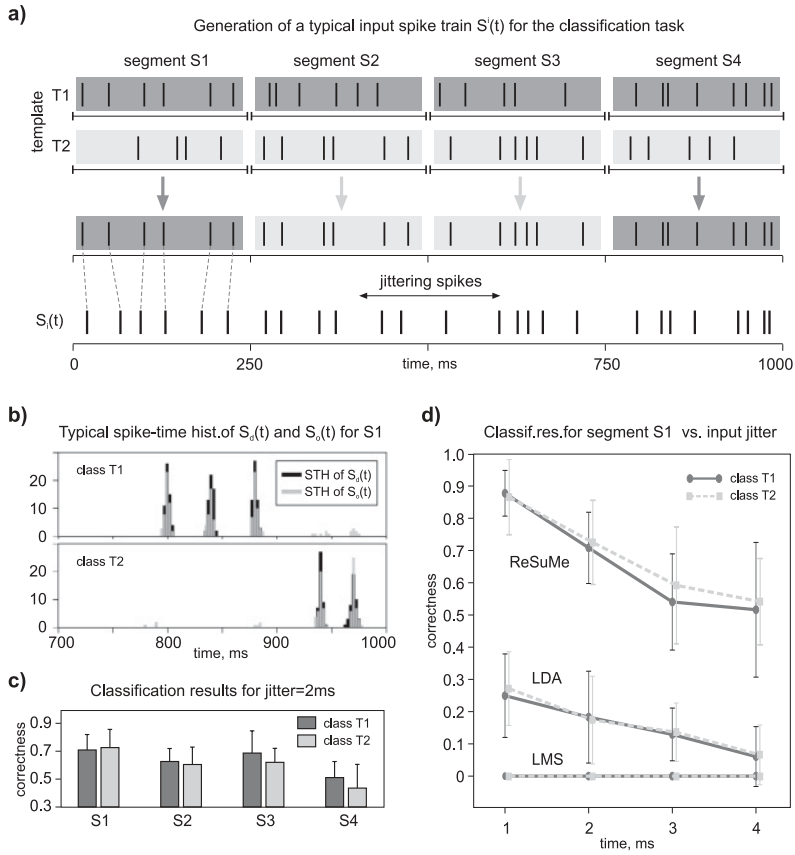

Figure 6: Classification task. The network is trained to classify jittered segments of input spike trains. (a) Input patterns of the length of 1000 ms are assumed to consist of four segments (S1–S4), each randomly drawn from one of two templates (T1,T2). Before an input pattern is presented to the classifier, all of its spikes are shifted in time by the amount drawn individually for each spike from a gaussian distribution with mean zero and a given variance. (b) The network is assumed to communicate a decision about the input classes T1,T2, by generating a predefined sequence of spikes corresponding to the particular input category. Spike time histograms (STH) of the network target ($S_d(t)$, in black) and output patterns ($S_o(t)$, in gray) calculated for 200 trials are presented. The output patterns highly resemble the targets. Note the variance of the target firing times, introduced to improve the robustness of neural responses to noise. (c) Average classification results (over 200 trials) obtained for the input jitter with variance 2 ms. (d) Performance of ReSuMe is compared to the Fisher linear discriminant (LDA) and the least mean square (LMS) algorithms. ReSuMe significantly outperforms the remaining two algorithms in the considered classification task. These results are presented for segment S1 and for the input jitters with variance 1–4 ms.

that the output pattern represents the given class T1 or T2 if all spikes of the corresponding target pattern are reproduced by the readout with a precision of 20 ms. We allow for one missing or one extra spike. Otherwise the output spike train is considered incorrect, and the classification result is not accepted. Note that in this scenario, the task is much more difficult than the binary classification considered in Maass et al. (2002), where the classifiers were supposed to output a single value (0 or 1) after the whole input pattern was presented to the network. Moreover, with the criteria proposed here, our model can output three decisions: class T1, class T2, or not assigned. This classification scheme is specifically suitable for the systems where it is important to communicate the classifier's inability to recognize the input category (e.g., due to high costs of error) rather than making unreliable decisions. In this case, the system should reject a pattern whenever the classification cannot be achieved with enough confidence (see, e.g., Bottou & Vapnik, 1992, for discussion). This approach is reasonable also if the input sets are not closed within the prespecified bounded regions of the feature space, but can potentially expand to its other areas (as it is the case in our task, where the introduced noise can potentially modify the template pattern in the unconstrained manner, locating the resulting spike train arbitrarily far away from the class prototype).

In Figure 6c we present the results of our experiment obtained on the validation set for all four readouts and the input jitter variance 2 ms (the results are averaged over 200 trials). Around 74% of the input patterns belonging to class T1 or class T2 are classified correctly for segment S1. Statistically similar results are obtained in the case of the segments S2 and S3. For the reasons explained above, the correctness of classification observed for the segment S4 is much lower. Still, however, 51% of the inputs of class T1, and correspondingly 45% of class T2, are classified correctly.

We note that the classification difficulty increases dramatically with the level of input jitters (see Figure 6d). On average, 90% of all input segments are classified correctly by readout S1 if the jitter variance equals 1 ms. On the contrary, the average correctness drops to around 55% as the jitter variance increases to 4 ms.

The performance of the ReSuMe method in this classification task was compared with the results obtained with the least mean square (LMS) method used in the original experiment by Maass et al. (2002). We also tested the classification ability of the Fisher linear discriminant analysis (LDA) approach, the method traditionally used in the classification tasks (Duda, Hart, & Stork, 2000; see the appendix for the details on both algorithms).

Since LMS and LDA are not suitable training methods for spiking neurons, we used the simple perceptrons as readouts instead of the LIF units. In order to take advantage of the temporal integration and memory capability of the reservoir in these memoryless units, the perceptrons were driven with the reservoir activity transformed into the continuous state by filtering the outputs from every reservoir unit with a low-pass filter (Maass et al., 2002)

(this approach is in fact equivalent to modeling the readouts by nonresetting leaky integrators with threshold). The original target spiking patterns used in our system were approximated by their binary representation; that is, we constructed the binary target vectors T by binning the spike timing of the target patterns into the bins of size t_s, with t_s being the sampling time of the discretized time domain. If there is a spike in the nth bin centered at $t = n \cdot t_s$, then $T(n)$ equals one, otherwise zero. The task of the perceptron readouts was to output at every simulation time step $n \cdot t_s$ the value equal to $T(n)$.

The classification results obtained with LDA and LMS for the sampling time $t_s = 1$ ms are presented in Figure 6d. In order to make the results statistically more reliable, we calculated the average classification correctness over eight experiments, where in each experiment, another randomly connected reservoir was constructed and the new input templates were generated. We observe that LMS was unable to reproduce the target patterns in any single trial (see Figure 6d). The correctness of LDA was at the level of 25% to 30% for the input jitter with a variance of 1 ms. The performance of LDA dropped to 10% as the jitter variance increased to 4 ms. Note that both LDA and LMS are the global optimization methods, which are believed to be able to find better solutions than the local methods, like ReSuMe. Moreover, the results presented for LMS and LDA are obtained for the sampling time t_s 10 times greater than the one used in the simulations with the LIF readout and the ReSuMe method,[3] which should make the classification task for LMS and LDA potentially easier (there are fewer samples where the classifier has to make the decisions). Despite these facts, ReSuMe proves to substantially outperform both LMS and LDA in the classification task. We note that for $t_s = 0.1$ ms, that is, for t_s equal to the simulation time step used in the case of ReSuMe, not only LMS but also LDA fails to reproduce the target patterns in any trial.

The observed superiority of ReSuMe over LSM and LDA seems surprising at first glance. But it can be rationally explained if we notice the following facts. First, we note that the complexity of the classification task for LSM and LDA depends on the relationship between the number of decisions that have to be made, the dimensionality of the feature space, and the number of available tunable parameters. In the experiment, this relationship is disadvantageous since there are 1000 simulation time steps, and at each time step, a single decision has to be made; at the same time there are 800 synaptic inputs (which define the dimensionality of the feature space) and 800 adaptive parameters, that is, synaptic weights. In other words there are 1000 points in 800-dimensional space, which are to be classified into two categories. In this case, the likelihood that the subsets of points representing the distinct two classes are linearly nonseparable is relatively high and

[3]The ReSuMe method and the LIF unit operate in the continuous time; we use a discretized time only for the computer simulations of both dynamic processes

increases with the number of points to be classified. Thus, the dependence of the classification complexity on the number of simulation time steps explains why both LSM and LDA fail to reproduce any target pattern if this parameter is increased.

Contrary to this case, temporal resolution is irrelevant for ReSuMe. What determines the complexity of learning with our algorithm is rather a number of spikes in a target pattern. Generally the more spikes there are, the more difficult the learning task becomes. On the contrary, the learning task is relatively easy when using a sparse firing code, exactly as in the experiment.

Another important property of ReSuMe that contributes to the success of ReSuMe in the task at hand is that the different groups of synapses can be optimized individually for the particular target firing times (as discussed in section 3.1.1). This property arises as a consequence of the temporal locality of our algorithm and constitutes yet another advantage of ReSuMe over LMS and LDA.

In another study, the classification capability of ReSuMe has been examined in comparison with Tempotron, a supervised learning method dedicated specifically to classifying spatiotemporal patterns of spikes (Gütig & Sompolinsky, 2006). The task of a neuron trained by either ReSuMe or Tempotron was to separate input spike patterns into two categories by emitting or nonemitting a spike at its output. The results of the comparison, presented in Florian (2008), demonstrate equally high performance of both methods in the task considered here. On the other hand, ReSuMe was shown to have some advantages over the Tempotron in terms of the richer repertoire of the coding schemes of the classification results. In fact, a thorough analysis of both algorithms reveals that Tempotron can be considered a particular implementation of the ReSuMe learning rule for the classification tasks (see Florian, 2008, for details).

3.3 Spike Shifting. In the experiments considered so far, training was performed according to equation 2.10 with the assumption that all conditions defined by equation 2.8 are fulfilled. In this section, we demonstrate that by weakening some of these conditions, we obtain new, interesting properties of the learning method.

Consider the ReSuMe algorithm given by equations 2.3 to 2.7. Let us again assume that $-b = a$, $-b_i = a_i$, $-b_o = a_d \geq 0$. But this time let the parameters of the learning windows be set individually for $X_{di}(t)$ and $X_{oi}(t)$. Without loss of generality, we may write $A_{di} = k_A \cdot A_{oi} > 0$ and $\tau_{di} = k_\tau \cdot \tau_{oi} > 0$, where $k_A, k_\tau \in \mathbb{R}^+$. And again we consider a case where $A_{id} = A_{io} = 0$, $\tau_{id} = \tau_{io} = 0$.

Now we investigate the influence of the different values of k_A and k_τ on the learning results. We begin with the theoretical analysis. For simplicity, we assume that the signals $S_i(t)$, $S_o(t)$, and $S_d(t)$ contain single spikes only and that the particular firings occur at times t_i, t_o, and t_d, respectively.

According to the assumptions made above, we can write the modified learning rule as

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{oi}(t) = S_d(t) \left[a_d + \int_0^\infty k_A A_{oi} \exp\left(-\frac{s}{k_\tau \tau_{oi}}\right) S_i(t-s)\, \mathrm{d}s \right] - S_o(t) \left[a_d + \int_0^\infty A_{oi} \exp\left(-\frac{s}{\tau_{oi}}\right) S_i(t-s)\, \mathrm{d}s \right], \quad (3.1)$$

with $S_i = \delta(t - t_i)$, $S_d = \delta(t - t_d)$, and $S_o = \delta(t - t_o)$.

We compute the total weight change Δw at time $t \to \infty$ (to ensure that $t > t_i, t_d, t_o$) by integrating equation 3.1 over a time interval $[0, +\infty)$. After performing necessary transformations, we obtain

$$\Delta w = k_A A_{oi} \exp\left(\frac{-t_d + t_i}{k_\tau \tau_{oi}}\right) + A_{oi} \exp\left(\frac{-t_o + t_i}{\tau_{oi}}\right). \quad (3.2)$$

Now we search for the steady state of the learning process defined by equation 3.1. By following a procedure similar to that in Ponulak (2006a), it can be shown that in the scenario, the stable fixed point of the learning process exists, and it is reached if and only if $\Delta w = 0$. According to equation 3.2, condition $\Delta w = 0$ is equivalent to

$$0 = k_A \exp\left(\frac{-t_d + t_i}{k_\tau \tau_{oi}}\right) + \exp\left(\frac{-t_o + t_i}{\tau_{oi}}\right). \quad (3.3)$$

To see the outcomes of this remark on the learning results, we examine equation 3.3 in two situations:

- Assume $k_A = 1$. Then, after substituting this to equation 3.3 and after some elementary transformations, we get

$$(t_o - t_i) = k_\tau (t_d - t_i). \quad (3.4)$$

- Assume $k_\tau = 1$. Applying this condition to equation 3.3 yields

$$(t_d - t_o) = \tau_{oi} \ln(k_A). \quad (3.5)$$

First, we examine the consequences of learning with $k_A = 1$ and $k_\tau \neq 1$. From equation 3.4, it is clear that in the steady state of the learning process, the time lag $(t_o - t_i)$ between the presynaptic spike and the resulting postsynaptic spike is proportional to the delay $(t_d - t_i)$ with the proportion quotient k_τ (see Figure 7a). For the given $k_\tau \neq 1$, the time delay $|(t_d - t_o)|$ between the desired and actual spike times increases as the neuron is to fire farther from the stimulus given at t_i. Only for the case $k_\tau = 1$ does the

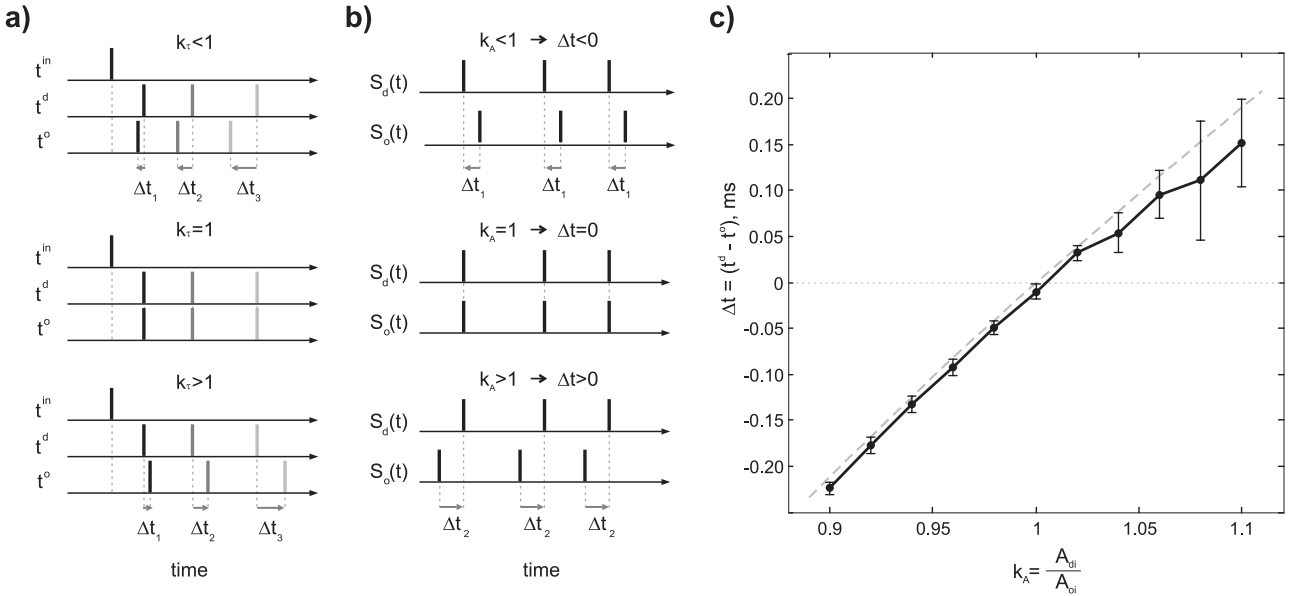

Figure 7: Spike shifting. (a) Illustration of the effect of parameter k_τ on the learning results. The output spikes are shifted in time with respect to the target ones, according to the relationship $(t_o - t_i)/(t_d - t_i) = k_\tau$. (b) Illustration of the effect of parameter k_A on the learning results. The output spikes are shifted in time with respect to the target ones by Δt, according to the relationship $\Delta t = (t_d - t_o) = \tau_+ ln(k_A)$. (c) Spike shift Δt of signals $S_o(t)$ with respect to the corresponding $S_d(t)$ as a function of k_A. Experimental results (solid line) with median (bold dots) and standard deviation (vertical bars). Theoretically predicted results (dashed line) obtained from equation 3.5.

delay $|(t_d - t_o)|$ equal zero (see Figure 7a). This conclusion, made here for single spikes, does not generally hold in a scenario with signals consisting of multiple spikes. In cases where a neuron is excited with many presynaptic spikes before it fires, the time of the resulting postsynaptic spike is a function of the neuron's excitation over a longer range of its history. In such a case, the contribution of the particular presynaptic spikes to the final time lag $(t_d - t_o)$ may not be easily distinguished, and formula 3.4 no longer holds.

Next, we consider learning with $k_\tau = 1$ and $k_A \neq 1$. In this case, the steady state of the learning process is represented by equation 3.5. From this equation, we see that t_o equals t_d if and only if $k_A = 1$, that is, if the amplitudes A_{di} and A_{oi} of the learning windows are equal. For $k_A \neq 1$, there is a fixed time lag $\Delta t = (t_d - t_o)$ between the desired and the actual spike times. The lag Δt is a function of the time constant τ_{oi} and the ratio k_A (see Figure 7b). This theoretical analysis is performed here for the scenario with the signals $S_i(t)$, $S_d(t)$, and $S_o(t)$ consisting of single spikes only. However, our computational experiments demonstrate that the conclusion drawn here from equation 3.5 holds even for sequences of spikes. To illustrate this compliance, let us consider an experiment with a single LIF neuron trained to reproduce 20 different sequences of spikes in response to the corresponding input spike patterns, all generated by a 20 Hz Poisson process. The training was performed according to equation 3.1 with $k_\tau = 1$. In the particular realizations of the experiment, the parameter k_A was varied systematically in the range $[0.9, 1.1]$. For each value of k_A, the median and the standard deviation of the time lag Δt were calculated.

After training, we observed that the neuron was able to precisely reproduce the relative timing of spikes of the target signals, but the particular spikes in $S_o(t)$ were shifted by the fixed time interval Δt with respect to the corresponding spikes in $S_d(t)$, as a function of k_A. In Figure 7c we present a plot of the relationship $\Delta t = f(k_A)$ obtained in the simulations (solid line) as compared to the one calculated directly from equation 3.5 (dashed line). We observe that the experimental results are consistent with our theoretical predictions.

The results presented in this section suggest that spiking neurons can potentially be used not only to reproduce but also to forecast the behavior of some reference objects, such as other spiking neural networks or biological neural structures. This issue, however, requires further exploration.

4 Discussion

In this letter, we considered a general learning problem of how to train neural circuits to generate arbitrary responses to patterned synaptic inputs. This is an important issue in a wide range of specific neural systems and tasks. To address this problem, we discussed supervised learning with the ReSuMe method. We demonstrated that ReSuMe can be successfully

applied to such computational tasks as sequence learning, classification, or spike shifting.

Initial results of our other studies suggest that ReSuMe can also be used in real-life control tasks. This ability was demonstrated in a set of computational experiments where spiking neural networks were trained as adaptive neurocontrollers to generate movement trajectories or to perform trajectory tracking tasks (Ponulak & Kasiński, 2006b; Ponulak et al., 2006, 2008; Belter et al., 2008; Ponulak, 2006b). Real-life applications of SNN, however, require efficient implementations of spiking models and learning algorithms. In this context, hardware VLSI emulators of SNN become an interesting topic of investigation (we refer to Maass & Bishop, 1999, for an overview). The main advantage of the VLSI systems over processor-based designs lies in their truly parallel mode of operation, which fully utilizes the essential property of parallelism found in spiking neural networks. This fact, in combination with the high computational speed of VLSI, can ensure near real-time information processing of hardware SNN even for the models of complex, large spiking networks.

ReSuMe possesses several properties that complement the advantages of VLSI systems in the real-life tasks and make the algorithm a good candidate for hardware implementations: simplicity, scalability, online training suitability, and fast convergence.

Recently a simple spiking neural circuit capable of learning with ReSuMe has been implemented in the FPGA platform to test the described properties (Kraft, Ponulak, & Kasiński, 2006). The implemented system demonstrates fast and stable learning. Due to its high-speed processing ability, the system is able to meet the time restrictions of many real-time tasks. Still, further research is required to implement large-scale spiking neural networks able to deal with the complexity of the real-world applications.

4.1 Biological Relevance of ReSuMe. The learning method considered in this letter proved to be an efficient computational model for the class of spiking neural networks. In this context, the algorithm is useful as such. However, the approach taken in ReSuMe was inspired also by biological mechanisms involved in learning and plasticity. This section discusses some properties of our algorithm in relation to their biological counterparts.

The basic assumption made in our model is that the instructions for the learning neurons are given as template temporal spike patterns to be reproduced (cf. Knudsen, 1994). In the brain, this form of instruction signal is thought to be used mainly for learning internal forward models (see Miall & Wolpert, 1996, for the discussion). Neural activity templates have also been observed to guide the alignment of the visuotectal maps in the binocular visual system (Udin & Keating, 1981). So far there is no direct evidence that these instructions are encoded in precisely timed sequences of spikes. However, several experiments indicate that neurons from the regions that

are potential candidates for providing supervisory signals can generate reproducible firing patterns with high temporal precision (Kolb, Rubia, & Bauswein, 1987; Berry, Warland, & Meister, 1997; Uzzell & Chichilnisky, 2004).

Another assumption made in ReSuMe is that the evolution of excitatory synapses is driven by the joint effect of two opposite processes: the inverse-STDP-like process ($X_{oi}(t)$) induced by pairing the pre- and postsynaptic spikes and the STDP-like process triggered by the temporal correlation between the presynaptic spikes and the instructive signals ($X_{di}(t)$) (see equation 2.3).

The first process ($X_{oi}(t)$) resembles biological homosynaptic anti-STDP, which has been observed in the central nervous system for a long time. The first evidence for anti-STDP came from the work of Bell, Han, Sugawara, and Grant (1997) and was observed in the electrosensory lobe of mormyrid electric fish. Recently anti-STDP has also been found in the mammalian dorsal cochlear nucleus (Tzounopoulos, Rubio, Keen1and, & Trussell, 2007) and in the cortical pyramidal cells (Kampa, Letzkus, & Stuart, 2007).

The second process considered in ReSuMe ($X_{di}(t)$) can be interpreted in a biological fashion as a presynaptically induced heterosynaptic plasticity—a phenomenon experimentally observed, for example, in the hippocampus (Judge & Hasselmo, 2004; Dudman, Tsay, & Siegelbaum, 2007) or in the amygdala (Humeau, Shaban, Bissiere, & Lüthi, 2003). In this form of plasticity, a simultaneous activation of synaptic inputs converging on a single postsynaptic neuron induces associative long-term plasticity in one or both inputs. This plasticity can be completely independent of the postsynaptic activity (Humeau et al., 2003; Dudman et al., 2007), as assumed also in our model.

Our model requires that the instructive signals induce heterosynaptic plasticity, but have a marginal or no direct effect on the postsynaptic somatic membrane potential. Recent work by Dudman et al. (2007) provides evidence for the existence of such a phenomenon in the hippocampal CA1 pyramidal neurons, where it has been observed that the distal perforant path inputs are able to induce long-term potentiation at the CA1 proximal Schaffer collateral synapses if the two inputs are paired at a precise interval. At the same time, these distal inputs are known to contribute little to the somatic spiking. Dudman and colleagues hypothesized that the observed heterosynaptic plasticity could contribute to the supervised learning process occurring at the CA1 pyramidal neurons, and the direct sensory information arriving at distal CA1 synapses through the perforant path could provide instructive signals for this process.

The short survey presented in this section reveals that all mechanisms necessary to implement ReSuMe in a biological manner exist in the brain. Still, direct evidence for the ReSuMe-like learning in the central nervous system is missing. Our analysis suggests, however, that if supervised learning rules similar to ReSuMe exist there, they should specifically be sought in

the systems where the opposite heterosynaptic and homosynaptic plasticity processes converge on the same synapses.

4.2 Relation to Other Methods. A review of the supervised learning methods for SNN reveals that only a few of the existing algorithms demonstrate the ability to learn and reproduce precisely timed sequences of spikes (Kasiński & Ponulak, 2006). In this group of algorithms, several different approaches have been used to learn temporal patterns of spikes.

Carnell and Richardson (2005) proposed to apply linear algebra formalisms to the task of spike time learning. Booij and Nguyen (2005) introduced a method based on the error backpropagation technique. The method was designed for specific neuron models, and some extra constraints had to be defined to deal with the discontinuity of the spiking process. Schrauwen and Campenhout (2006) derived another algorithm based on error backpropagation, where smooth approximation of spiking neurons was used to enable calculation of the error gradient.

These methods represent an interesting approach to spike sequence learning. However, they are designed either to perform training in a batch (off-line) mode (Carnell & Richardson, 2005; Booij & Nguyen, 2005), and hence are not suitable for the class of applications that require efficient, online adaptive learning or they are limited to the analytically tractable neuron models (Booij & Nguyen, 2005; Schrauwen & Campenhout, 2006). Their biological plausibility also raises serious concerns.

Another group of supervised learning algorithms is represented by the methods inspired by the biological plasticity mechanisms. Legenstein, Naeger, and Maass (2005) introduced an STDP-based supervised learning algorithm with supervision provided by clamping the postsynaptic neuron to a target signal. The clamping currents forced the learning neuron to fire at the target times and prevented it from firing at other times. It was demonstrated that this approach enabled the trained neuron to approximate the given transformations of input to output spike trains. The authors reported, however, some drawbacks of the algorithm. Since the clamping currents suppressed all undesired firings during the training, the only correlations of pre- and postsynaptic activities could occur around the target firing times. At other times, there was no correlation, and thus no mechanism to weaken these synaptic weights that led the neuron to fire at undesired times during the testing phase. Another reported problem common to all supervised Hebbian approaches is that synapses continue to change their parameters even if the neuron fires exactly at the desired times. This results in a bimodal distribution of weights, where each weight assumes either its minimal or its maximal possible value. As a proposed solution to this problem, the algorithm was tested with a multiplicative variation of STDP (Gütig, Aharonov, Rotter, & Sompolinsky, 2003), which produced more stable intermediate weight values. However, the authors reported that learning

with this modified version of STDP was still highly sensitive to input signal distributions.

Due to the different learning strategy used in ReSuMe, our algorithm does not experience the described problems. The discussion of this topic in relation to the model of Legenstein et al. was presented in Ponulak and Kasiński (2006a)

An alternative, probabilistic version of the spike-based supervised Hebbian learning was introduced in Pfister, Toyoizumi, Barber, and Gerstner (2006). The method was derived by optimizing the likelihood of postsynaptic firing at one or several desired firing times. However, it is hard to estimate a potential, practical ability of the method to learn complex sequences of spikes, since all experiments in Pfister et al. (2006) were illustrated with the examples of the target spike trains consisting of at most two spikes.

Recently another interesting method for spike sequence learning has been proposed in a framework of reinforcement learning. The algorithm, known as reward-modulated spike-timing-dependent-plasticity (RM-STDP), has been introduced in Florian (2007) and Izhikevich (2007) (a similar method has also been described in Farries & Fairhall, 2007). RM-STDP proved to be capable of learning complex sequences of spikes with a high precision. However, this ability is constrained by several limitations as pointed out by Legenstein, Pecevski, and Maass (2008). It has been indicated that the algorithm requires a certain level of noise to converge. This noise is necessary, for example, to cope with a problem of silent neurons.[4] Analysis of the algorithm demonstrates also that it is not able to learn arbitrary firing patterns, but only those characterized by some minimum firing rate. Finally, the learning convergence of the algorithm in the task of spike sequence learning is slow and typically requires thousands of learning epochs to converge. Despite these limitations, RM-STDP proves to be a powerful and promising learning method.

The algorithms discussed in this section represent different approaches to supervised learning in spiking neural networks and are efficient in the specific computational tasks. From this perspective, they constitute interesting alternatives to ReSuMe.

4.3 Conclusions and Future Work. The ReSuMe concept discussed here introduced the following enhancement to the theory of neural computation. First, it offers an efficient supervised learning algorithm that not only enables the neurons to reproduce target sequences of spikes with a high precision, but also controls a time lag between the target and reproduced

[4]In associative learning rules like RM-STDP, which rely on the homosynaptic STDP only, a silent neuron cannot recover from its silent state if there is no temporal correlation of pre- and postsynaptic activity. This problem has been efficiently solved in ReSuMe by employing another term based on the correlation of presynaptic and target activity.

spikes. Second, it provides continuity between the well-established principles of supervised learning theory and the physiological mechanisms able to implement the learning algorithm in a biologically plausible way. Third, it predicted a new form of heterosynaptic plasticity where the instructive signal at one input triggers long-term changes at other synaptic inputs, and where at the same time the instructive signal is assumed not to contribute to the action potential generation at the postsynaptic neuron. The existence of the described phenomenon in the central nervous system has recently been confirmed in experimental studies on hippocampal neurons (Dudman et al., 2007).

As an interesting direction of future work on ReSuMe, we consider the enhancement of the learning algorithm to enable consistent modifications of the synaptic connections in a whole network. It is expected that such an approach would improve the learning and memory capacity of spiking networks and, as a consequence, would reduce the size of the networks required to complete particular learning tasks, as compared to recently used approaches with reservoir networks.

Another line of investigation involves extensions of the ReSuMe learning rule to the possible delay of the instructive signal. This issue has not been addressed so far, since we assumed that in the tasks considered here, there was no delay or incorporation of the delay was a desirable property of a neural network. Still, there exist many other tasks where the delay of the target signal needs to be taken into account. However, direct implementation of the recently proposed concepts addressing this issue (see, e.g., Izhikevich, 2007) in ReSuMe is impossible, and new mechanisms need to be investigated.

We hope that the learning framework and its implications discussed in this letter will contribute to the progress of the theory of supervised learning and stimulate the study of real-world applications of SNN. We also believe that the research on ReSuMe can provide some further insights into the neural mechanisms governing learning in the brain.

Appendix: Details of Models and Simulations

A.1 Neuron Models. In our simulations we used leaky integrate-and-fire units (Gerstner & Kistler, 2002) with the dynamics defined by the following formula:

$$\tau_m \frac{dV_m}{dt} = -(V_m - V_r) + R_m(I_{syn} + I_{ns} + I_{in}),$$

where V_m is a membrane potential, $\tau_m = C_m R_m$ is the membrane time constant, $C_m = 1$ nF and $R_m = 10\,\mathrm{M}\Omega$ are the membrane conductance and resistance, respectively, $V_r = -60$ mV is a membrane potential at rest, I_{syn} is a sum of the currents supplied by the particular synapses entering the

given neuron, $I_{in} = 0.1$ nA is a constant injected current, and I_{ns} is a non-specific background current modeled as a gaussian process with zero mean and variance chosen in the different experiments in the range $[0, 30]$ nA. At time $t = 0$, the membrane potential is set to $V_{init} = -60 \pm 1$ mV. If V_m exceeds the threshold voltage $\vartheta = -55$ mV, it is reset to $V_{res} = -65$ mV and held there for the length $t_{ref} = 5$ ms of the absolute refractory period. The membrane potential and the currents are in general functions of time and should formally be denoted by $V_m(t)$, $I(t)$, and so forth. Here and in the following we omit the t symbols for clarity.

For the Hodgkin-Huxley units, we adopted a model implementation from CSIM (2002). The neuron dynamics is described by the following equation:

$$C_m \frac{V_m}{dt} = -R_m^{-1}(V_m - V_r) - \bar{g}_{Na} m^3 h (V_m - E_{Na}) - \bar{g}_K n^4 (V_m - E_K) + (I_{syn} + I_{ns} + I_{in}),$$

with $C_m = 1$ nF, $R_m = 1\,\mathrm{M}\Omega$, $I_{in} = 0.1$ nA, I_{ns} with mean zero and variance 2 nA. The maximum current conductances g_{Na} and g_K for the Na and K channels, respectively, the reversal potentials E_{Na}, E_K and the gating variables m, n, h are defined as in Hodgkin and Huxley (1952).

The model of neuron proposed by Izhikevich (2003) and used in our simulations is given by the following formula:

$$\begin{cases} \dfrac{dV_m}{dt} = 0.04V_m^2 + 5V_m + 140 - V_{rec} + (I_{syn} + I_{ns} + I_{in}), \\ \dfrac{dV_{rec}}{dt} = a(bV_m - V_{rec}), \end{cases}$$

with the auxiliary after-spike resetting condition: if $V_m \geq 30$ mV, then $V_m \leftarrow c$, $V_{rec} \leftarrow (V_{rec} + d)$. Here V_m again represents the membrane potential, and V_{rec} is a membrane recovery variable. The constants $a = 3 \cdot 10^{-3}$, $b = 0.35$, $c = -50$, $d = 2$ are set such that the neuron exhibits bursting properties (see Izhikevich, 2003). The remaining parameters are set to $I_{in} = -0.5$ nA, mean of I_{ns} is zero, and variance 2 nA, $V_{init} = -95 \pm 1$ mV, $t_{ref} = 5$ ms.

A.2 Model of Synaptic Response. We implement a synaptic response as $I_{syn} = w \cdot \exp(-\tau_d / t)$ for each spike that hits the synapse at time t with an amplitude of w and a decay time constant of τ_d. The default value of τ_d in our experiments is 3 ms. It is assumed that the responses of all spikes are added up linearly.

A.3 Error Measures. To quantitatively evaluate the performance of learning, we adopted a correlation-based metric C proposed by Schreiber,

Fellous, Whitmer, Tiesinga, and Sejnowski (2003) as a measure of the distance between the target and output spike trains. The measure C equals one for the identical spike trains and drops toward zero for the loosely correlated trains. The metric is calculated after every learning epoch according to

$$C = \frac{v_d \cdot v_o}{|v_d||v_o|},$$

where the quantities v_d, v_o are vectors representing a convolution (in discrete time) of the target and actual output spike trains with a second-order low-pass filter with time constants $\tau_1 = 2$ ms and $\tau_2 = 4$ ms; $v_d \cdot v_o$ is the inner product, and $|v_d|, |v_o|$ are the Euclidean norms of v_d and v_o, respectively.

We also introduce another measure, which we call shift error and denote by $e(t)$. It measures the time shift between the individual spikes in the target pattern $S_d(t)$ and the corresponding spikes in the output signal $S_o(t)$, provided that all spikes in $S_d(t)$ have been reproduced in $S_o(t)$ up to some bounded precision. We formally define $e(t)$ in the following way: for every spike $f = 1, \ldots, N$ in $S_d(t)$, we define $e^f(t) = (t_d^f - t_o^f)$, where t_d^f, t_o^f are the times of the fth spikes in $S_d(t)$ and $S_o(t)$, respectively.

A.4 Learning Rule Parameters. In all experiments, except in section 3.3, learning was performed according to the rule given by equation 2.10. Typical parameter values used in our simulations are $a_d = 0.005$, $A_{di} = 20 \cdot 10^{-11}$, $\tau_{di} = 5$ ms, $A_{id} = 0$, and $\tau_{id} = 0$.

In the experiment illustrating the ability of ReSuMe to shift spikes in time (see section 3.3), we performed training according to equation 3.1 with the parameter values $a_d = 0.004$, $A_{oi} = 20 \cdot 10^{-11}$, $\tau_{oi} = 2$ ms.

In all experiments we assumed that inhibitory connections are represented by negative synaptic weights and ReSuMe was allowed to change the sign (and thus a type) of the trained synapses (as discussed in section 2).

A.5 Details to Network Implementation. In sections 3.1.1 to 3.1.3 and 3.3, we considered single-layer networks with multiple inputs converging on the trained neurons. In the experiments, 60% to 100% of connections have been assumed excitatory. Efficacies of all synaptic connections have been initialized randomly according to gaussian distributions $N(w_\mu, w_{\sigma 2})$, with the typical parameter values for mean $w_\mu = \pm[1, 5] \cdot 10^{-10}$ and variance $w_{\sigma 2} = [5, 25] \cdot 10^{-20}$ (+ refers to the excitatory connections and − to the inhibitory ones). In all cases, zero synaptic delays have been assumed.

In sections 3.1.4 and 3.2, we used reservoir networks with parameters listed below. The size of the reservoir was $8 \times 5 \times 20$ neurons. Parameters of connections from inputs to the reservoir were as follows: probability of connections from any input to any reservoir neuron $p = 0.3$, fraction of excitatory connections $f_{exc} = 100\%$, strength of synaptic connections drawn from gaussian distribution $N(1 \cdot 10^{-8}, 9 \cdot 10^{-18})$, synaptic delays drawn

from gaussian distribution $N(1.5, 2 \cdot 10^{-3})$ ms, and synaptic time constant $\tau_d = 3$ ms. Parameters of connections within the reservoir were as follows: $f_{exc} = 85\%$, synaptic weights were chosen according to $N(\pm 4 \cdot 10^{-9}, 64 \cdot 10^{-20})$, and synaptic delays were chosen according to $N(1.5, 2 \cdot 10^{-3})$ ms, $\tau_d = 6$ ms. The probability of a synaptic connection from neuron a to neuron b, and from b to a, is defined as $C_{scale} \cdot c \cdot \exp(-D(a,b)^2/\lambda)$, where C_{scale}, c, and λ are positive constants and $D(a,b)$ is the Euclidean distance between the neurons a and b. Here, $C_{scale} = 10$, $\lambda = 2.5$, and depending on whether neurons a and b are excitatory (E) or inhibitory (I), the value of c is 0.3 (EE), 0.2 (EI), 0.4 (IE), 0.1 (II).

Parameters of connections from the reservoir to readouts were $f_{exc} = 80\%$, synaptic efficacies initialized with $N(\pm 3.5 \cdot 10^{-10}, 25 \cdot 10^{-20})$, $d = 0$ ms, $\tau_d = 1$ ms, all reservoir neurons have been assumed to converge onto all readout neurons.

A.6 Details to the LMS and LDA Algorithms Used in Section 3.2. We used the implementation of the least mean square and Fisher linear discriminant algorithms provided with a CSIM simulator (CSIM, 2002). The reservoir activity was transformed into the continuous state with a low-pass filter with the time constant $\tau = 30$ ms.

A.7 Simulation Details. All simulations considered in this letter were performed with CSIM (a neural Circuit SIMulator) (CSIM, 2002). We used a discrete time simulation with a resolution of 10^{-5} s in all experiments except section 3.2, where we used a resolution $10^{-4}s$ for ReSuMe and $10^{-3}s$ for the experiment with the LMS and LDA algorithms.

Acknowledgments

We are grateful to Razvan Florian and Andrew Meso for their critical comments on the early version of the manuscript and to Ioannis Vlachos for proofreading the final draft. We thank Joshua Dudman and Andreas Lüthi for stimulating discussions on presynaptically triggered heterosynaptic plasticity. Numerous helpful comments from two anonymous reviewers are gratefully acknowledged. This work was partially supported by the German Federal Ministry of Education and Research (grant 01GQ0420 to BCCN Freiburg).

References

Belatreche, A., Maguire, L. P., McGinnity, M., & Wu, Q. X. (2003). A method for supervised training of spiking neural networks. In *Proceedings of IEEE Cybernetics Intelligence—Challenges and Advances (CICA) 2003* (pp. 39–44). Piscataway, NJ: IEEE.

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

Belter, D., Ponulak, F., & Rotter, S. (2008). Adaptive movement control with spiking neural networks, part II: Composite control. In *Proceedings of Recent Advances in Neuro-Robotics, Symposium: Sensorimotor Control*. Freiburg: Freiburg University.

Berry, M. J., Warland, D. K., & Meister, M. (1997). The structure and precision of retinal spike trains. *Proceedings of the National Academy of Science of USA*, *94*, 5411–5416.

Bi, G.-Q. (2002). Spatiotemporal specificity of synaptic plasticity: Cellular rules and mechanisms. *Biological Cybernetics*, *87*, 319–332.

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

Bohte, S., Kok, J., & Poutré, H. L. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, *48*, 17–37.

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

Borst, A., & Theunissen, F. E. (1999). Information theory and neural coding. *Nature Neuroscience*, *2*(11), 947–957.

Bottou, L., & Vapnik, V. (1992). Local learning algorithms. *Neural Computation*, *4*(6), 888–900.

Carey, M. R., Medina, J. F., & Lisberger, S. G. (2005). Instructive signals for motor learning from visual cortical area MT. *Nature Neuroscience*, *8*, 813–819.

Carnell, A., & Richardson, D. (2005). Linear algebra for time series of spikes. In *Proceedings of the European Symposium on Artificial Neural Networks, ESANN'2005* (pp. 363–368). Brussels: D-Facto.

CSIM. (2002). CSIM: *A neural circuit SIMulator*. Graz: IGI LSM Group, Technical University. Available online at http://www.lsm.tugraz.at.

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex. *Neural Networks*, *12*, 961–974.

Duda, R. O., Hart, P. E., & Stork, D. G.(2000). *Pattern classification* (2nd ed.). Hoboken, NJ: Wiley Interscience.

Dudman, J. T., Tsay, D., & Siegelbaum, S. A. (2007). A role for distal synaptic inputs: Instructive signals for hippocampal synaptic plasticity. *Neuron*, *56*(5), 866–879.

Farries, M., & Fairhall, A. (2007). Reinforcement learning with modulated spike timing-dependent synaptic plasticity. *Journal of Neurophysiology*, *98*, 3648–3665.

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

Florian, R. (2008). Tempotron-like learning with ReSuMe. In *Proceedings of the 18th International Conference on Artificial Neural Networks, ICANN'2008* (pp. 368–375). Berlin: Springer-Verlag.

Furukawa, S., & Middlebrooks, J. C. (2002). Cortical representation of auditory space: Information-bearing features of spike patterns. *Journal of Neurophysiology*, *87*, 1749–1762.

Gaze, R., Keating, M., Szekely, G., & Beazley, L. (1970). Binocular interaction in the formation of specific intertectal neuronal connexions. *Proceedings of the Royal Society [Biol]*, *175*, 107–147.

Georgopoulos, A. P. (1986). On reaching. *Annual Reviews in Neuroscience*, *9*, 147–170.

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

Ghosh-Dastidar, S., & Adeli, H. (2007). Improved spiking neural networks for EEG classification and epilepsy and seizure detection. *Integrated Computer-Aided Engineering*, *14*(3), 187–212.

Gütig, R., Aharonov, R., Rotter, S., & Sompolinsky, H. (2003). Learning input correlations through non-linear temporally asymmetric Hebbian plasticity. *Journal of Neuroscience*, *23*, 3697–3714.

Gütig, R., & Sompolinsky, H. (2006). The tempotron: A neuron that learns spike timing–based decisions. *Nature Neuroscience*, *9*, 420–428.

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the theory of neural networks*. Reading, MA: Addison-Wesley.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, *117*, 500–544.

Hopfield, J. J., & Brody, C. D. (2001). What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. *Proceedings of the National Academy of Sciences of USA*, *98*, 1282–1287.

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

Ito, M. (2000a). Mechanisms of motor learning in the cerebellum. *Brain Research*, *886*, 237–245.

Ito, M. (2000b). Neural control of cognition and language. In A. Marantz, Y. Miyashita, & W. O'Neil (Eds.), *Image, language, brain*. Cambridge, MA: MIT Press.

Ito, M. (2008). Control of mental activities by internal models in the cerebellum. *Nature Reviews Neuroscience*, *9*, 304–313.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transaction on Neural Networks*, *14*(6), 1569–1572.

Izhikevich, E. M (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex*, *17*(10), 2443–2452.

Jaeger, H. (2001). *The "echo state" approach to analysing and training recurrent neural networks* (Computing Science Tech. Rep. GMD 148). Sankt Augustin: Fraunhofer Institute for Autonomous Intelligent Systems, German National Research Center for Information Technology.

Jaeger, H., Maass, W., & Principe, J. (2007). Editorial. *Neural Networks*, *20*, 287–432.

Judge, S., & Hasselmo, M. (2004). Theta rhythmic stimulation of stratum lacunosum-moleculare in rat hippocampus contributes to associative LTP at a phase offset in stratum radiatum. *Journal of Neurophysiology*, *92*, 1615–1624.

Kampa, B. M., Letzkus, J. J., & Stuart, G. J. (2007). Dendritic mechanisms controlling spike-timing-dependent synaptic plasticity. *Trends in Neuroscience*, *30*(9), 456–463.

Kasiński, A., & Ponulak, F. (2005). Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks. In

Proceedings of the 15th International Conference on Artificial Neural Networks, ICANN'2005 (pp. 145–153). Berlin: Springer-Verlag.

Kasiński, A., & 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.

Kawato, M., & Gomi, H. (1992a). A computational model of four regions of the cerebellum based on feedback-error-learning. *Biological Cybernetics*, *68*, 95–103.

Kawato, M., & Gomi, H. (1992b). The cerebellum and VOR/OKR learning models. *Trends in Neurosciences*, *15*, 445–453.

Kistler, W. (2002). Spike-timing dependent synaptic plasticity: A phenomenological framework. *Biological Cybernetics*, *87*, 416–427.

Kistler, W., & van Hemmen, L. J. (2000). Modeling synaptic plasticity in conjunction with the timing of pre- and postsynaptic action potentials. *Neural Computation*, *12*, 385–405.

Knudsen, E. I. (1991). Visual instruction of the neural map of auditory space in the developing optic tectum. *Science*, *5015*(253), 85–87.

Knudsen, E. I. (1994). Supervised learning in the brain. *Journal of Neuroscience*, *14*(7), 3985–3997.

Knudsen, E. I. (2002). Instructed learning in the auditory localization pathway of the barn owl. *Nature*, *417*, 322–328.

Kolb, F. P., Rubia, F. J., & Bauswein, E. (1987). Comparative analysis of cerebellar unit discharge patterns in the decerebrate cat during passive movements. *Experimental Brain Research*, *68*(2), 219–233.

Kraft, M., Ponulak, F., & Kasiński, A. (2006). FPGA implementation of ReSuMe learning in spiking neural networks. In *Proceedings of EPFL LATSIS Symposium 2006, Dynamical Principles for Neuroscience and Intelligent Biomimetic Devices* (pp. 97–98). N. p.

Kroese, B., & van der Smagt, P. (1996). *An introduction to neural networks* (8th ed.). Amsterdam: The University of Amsterdam.

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

Legenstein, R., Pecevski, D., & Maass, W. (2008). A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLoS Computational Biology*, *4*(10), 1–27.

Lisberger, S. G. (1995). Learning: A mechanism of learning found?. *Current Biology*, *5*(3), 221–224.

Maass, W., & Bishop, C. (Eds.). (1999). *Pulsed neural networks*. Cambridge, MA: MIT Press.

Maass, W., & Markram, H. (2003). Temporal integration in recurrent microcircuits. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (2nd ed., pp. 1159–1163). Cambridge, MA: MIT Press.

Maass, W., Natschlaeger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*(11), 2531–2560.

Maass, W., Natschlaeger, T., & Markram, H. (2004). Computational models for generic cortical microcircuits. In J. Feng (Ed.), *Computational neuroscience: A comprehensive approach* (pp. 575–605). Boca Raton, FL: Chapman and Hall/CRC.

Mainen, Z. F., & Sejnowski, T. J. (1995). Reliability of spike timing in neocortical neurons. *Science*, *268*, 1503–1506.

Mauk, M. D., & Buonomano, D. V. (2004). The neural basis of temporal processing. *Annual Review of Neuroscience*, *27*, 307–340.

Miall, C. R., & Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural Networks*, *9*(8), 1265–1279.

Montemurro, M. A., Panzeri, S., Maravall, M., Alenda, A., Bale, M. R., & Brambilla, M., et al. (2007). Role of precise spike timing in coding of dynamic vibrissa stimuli in somatosensory thalamus. *Journal of Neurophysiology*, *98*, 1871–1882.

Montgomery, J., Carton, G., & Bodznick, D. (2002). Error-driven motor learning in fish. *Biological Bulletin*, *203*(2), 238–239.

Muresan, R. (2002). Complex object recognition using a biologically plausible neural model. In *Proceedings of the 2nd WSEAS International Conference on Robotics, Distance Learning and Intelligent Communication Systems*. N.p.: WSEAS Press.

Natschlaeger, T., Markram, H., & Maass, W. (2003). Computer models and analysis tools for neural microcircuits. In R. Koetter (Ed.), *Neuroscience databases. A practical guide* (pp. 123–138). Norwell, MA: Kluwer.

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

Ponulak, F. (2005). *ReSuMe—new supervised learning method for spiking neural networks* (Tech. Rep.). Poznań: Institute of Control and Information Engineering, Poznań University of Technology. Available online at http://d1.cie.put.poznan.pl/∼fp/.

Ponulak, F. (2006a). *ReSuMe—Proof of convergence* (Tech. Rep.). Poznań: Institute of Control and Information Engineering, Poznań University of Technology. Available online at: http://d1.cie.put.poznan.pl/∼fp/.

Ponulak, F. (2006b). *Supervised learning in spiking neural networks with ReSuMe method*. Unpublished doctoral dissertation, Poznań University of Technology, Poland. Available online at: http://d1.cie.put.poznan.pl/∼fp/.

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

Ponulak, F. (2009). *Memory capacity of spiking neurons*. Manuscript in preparation.

Ponulak, F., Belter, D., & Kasiński, A. (2006). Adaptive central pattern generator based on spiking neural networks. In *Proceedings of EPFL LATSIS Symposium 2006, Dynamical Principles for Neuroscience and Intelligent Biomimetic Devices* (pp. 121–122). N.p.

Ponulak, F., Belter, D., & Rotter, S. (2008). Adaptive movement control with spiking neural networks, part I: Feedforward control. In *Proceedings of Recent Advances in Neuro-Robotics, Symposium: Sensorimotor Control*. Freiburg: Freiburg University.

Ponulak, F., & Kasiński, A. (2006a). Generalization properties of SNN trained with ReSuMe. In *Proceedings of European Symposium on Artificial Neural Networks, ESANN'2006* (pp. 623–629) Brussels: D-Facto.

Ponulak, F., & Kasiński, A. (2006b). ReSuMe learning method for spiking neural networks dedicated to neuroprostheses control. In *Proceedings of EPFL LATSIS Symposium 2006, Dynamical Principles for Neuroscience and Intelligent Biomimetic Devices* (pp. 119–120). N.p.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1997). *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press.

Roberts, P. D., & Bell, C. C. (2002). Spike timing dependent synaptic plasticity in biological systems. *Biological Cybernetics*, *87*(5–6), 392–403.

Rojas, R. (1996). *Neural networks: A systematic introduction*. Berlin: Springer-Verlag.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.

Schneidman, E. (2001). *Noise and information in neural codes*. Unpublished doctoral dissertation, Hebrew University.

Schrauwen, B., & Campenhout, J. V. (2006). Backpropagation for population-temporal coded spiking neural networks. In *Proceedings of the International Conference on Neural Networks* (pp. 1797–1804) Piscataway, NJ: IEEE.

Schreiber, S., Fellous, J., Whitmer, D., Tiesinga, P., & Sejnowski, T. (2003). A new correlation-based measure of spike timing reliability. *Neurocomputing*, *52–54*, 925–931.

Shidara, M., Kawano, K., Gomi, H., & Kawato, M. (1993). Inverse-dynamics model of eye movement control by Purkinje cells in the cerebellum. *Nature*, *365*, 50–52.

Shmiel, T., Drori, R., Shmiel, O., Ben-Shaul, Y., Nadasdy, Z., Shemesh, M., et al. (2005). Neurons of the cerebral cortex exhibit precise interspike timing in correspondence to behavior. *Proceedings of the National Academy of Sciences of USA*, *102*(51), 18655–18657.

Thach, W. T. (1996). On the specific role of the cerebellum in motor learning and cognition: Clues from PET activation and lesion studies in man. *Behavioral Brain Science*, *19*, 411–431.

Tiesinga, P., Fellous, J.-M., & Sejnowski, T. J. (2008). Regulation of spike timing in visual cortical circuits. *Nature Reviews Neuroscience*, *9*, 97–107.

Tiňo, P., & Mills, A. J. (2005). Learning beyond finite memory in recurrent networks of spiking neurons. In L. Wang, K. Chen, & Y. Ong (Eds.), *Advances in natural computation, ICNC'2005* (pp. 666–675). Berlin: Springer-Verlag.

Tzounopoulos, T., Rubio, M. E., Keen1and, J. E., & Trussell, L. O. (2007). Coactivation of pre- and postsynaptic signaling mechanisms determines cell-specific spike-timing-dependent plasticity. *Neuron*, *54*(2), 291–301.

Udin, S. B. (1985). The role of visual experience in the formation of binocular projections in frogs. *Cellular and Molecular Neurobiology*, *5*, 85–102.

Udin, S. B., & Keating, M. (1981). Plasticity in a central nervous pathway in Xenopus: Anatomical changes in the isthmotectal projection after larval eye rotation. *Journal on Computational Neurology*, *203*(4), 575–594.

Uzzell V. J., & Chichilnisky, E. J. (2004). Precision of spike trains in primate retinal ganglion cells. *Journal of Neurophysiology*, *92*, 780–789.

van Rossum, M. C., O'Brien, B. J., & Smith, R. G. (2003). Effects of noise on the spike timing precision of retinal ganglion cells. *Journal of Neurophysiology*, *89*, 2406–2419.

Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis*. Unpublished doctoral dissertation, Harvard University.

Widrow, B. (1962). Generalization and information storage in networks of Adaline "neurons." In M. Yovitz, G. Jacobi, & G. Goldstein (Eds.), *Self-organizing systems* (pp. 435–461). Spartan Books.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *IRE WESCON Convention Record*, *4*, 96–104.

Wolpert, D. M., Miall, C. R., & Kawato, M. (1998). Internal models in the cerebellum. *Trends in Cognitive Sciences*, *2*, 338–347.

Xin, J., & Embrechts, M. J. (2001). Supervised learning with spiking neuron networks. In *Proceedings IEEE International Joint Conference on Neural Networks, IJCNN'01* (pp. 1772–1777). Piscataway, NJ: IEEE.

Received November 5, 2008; accepted May 5, 2009.