Theoretical Computer Science 261 (2001) 157–178

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the relevance of time in neural computation and learning

Wolfgang Maass

Institute for Theoretical Computer Science, Technische Universitaet Graz, Inffeldgasse 16b, A-8010 Graz, Austria

Abstract

We discuss models for computation in biological neural systems that are based on the current state of knowledge in neurophysiology. Differences and similarities to traditional neural network models are highlighted. It turns out that many important questions regarding computation and learning in biological neural systems cannot be adequately addressed in traditional neural network models. In particular, the role of time is quite different in biologically more realistic models, and many fundamental questions regarding computation and learning have to be rethought for this context. Simultaneously, a somewhat related new generation of VLSI-chips is emerging ("pulsed VLSI") where new ideas about computing and learning with temporal coding can be tested in an engineering context. Articles with details to models and results that are sketched in this article can be found at http://www.tu-graz.ac.at/igi/maass/. We refer to Maass and Bishop (Eds., Pulsed Neural Network, MIT Press, Cambridge, MA, 1999) for a collection of survey articles that contain further details and references. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Neural computation; Temporal coding; Spiking neurons; Computational complexity

1. Introduction

An analysis of the role of a gate g in a computation on a familiar computational model, such as a boolean circuit or an artificial neural net, is likely to focus on the question: *What* is the output of g during this computation?

It turns out that an analysis of the role of a biological neuron v in a neural computation has to proceed differently. One structural difference arises from the fact that most biological neurons can produce just one output signal. They can "fire" and thereby generate a short electrical pulse, which is called "action potential", or "spike" (see

E-mail address: maass@igi.tu-graz.ac.at (W. Maass).

PII: S0304-3975(00)00137-7

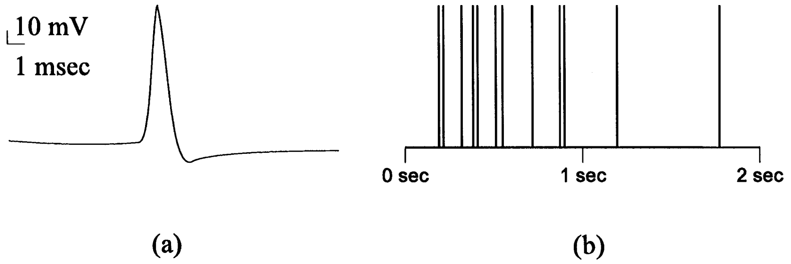

Fig. 1. (a) Typical action potential (spike). (b) A typical spike train produced by a neuron (each firing time marked by a bar).

Fig. 1(a)). Each spike has the same shape. Hence the key question regarding the contribution of a single neuron v to a computation in a biological neural system is: *At what times* does v fire?

Thus, whereas time has deliberately been removed as a resource for encoding information in most traditional computational models (through synchronization or some other pre-assigned schedule), it plays an essential role in biological neural computation. One may illustrate the significance of time in biological neural computation with the analogy of a symphony in music, where one gains little insight if one just analyzes *which* tones are played during a performance of the piece. Instead, one has to analyze *at what times* each tone is played. More precisely, one has to analyze how each tone is embedded into the temporal pattern of the “outputs” of all the other instruments.

Traditional neural network models, such as threshold circuits or sigmoidal neural nets are usually employed in a synchronized mode, just like other circuit models. They can be viewed as models for biological neural computation if one of the following two assumptions holds:

Assumption (A). There exists a clock in the biological neural system, which synchronizes the computation and partitions it into global *computation steps* in such a way, that one can interpret the output of a neuron v at each computation step t as 1 or 0, depending on whether v fires or does not fire at step t. Furthermore, whether a neuron fires at step t depends only on firings of other neurons at step $t-1$ (and it does not matter when exactly a neuron fires during a computation step).

Assumption (B). The individual firing times in the biological neural system carry no information. The *number* of firings of a neuron v within a time window of a certain length (i.e. the current “firing rate” of v) is the only relevant variable.

A computation in a biological neural system which satisfies Assumption (A) can be modeled quite well by a threshold circuit. Valiant’s formal model [59] for a biological neural system is largely based on this Assumption (A).

A computation in a biological neural system which satisfies Assumption (B) can be modeled quite well by a sigmoidal (i.e. analog) neural net. The analog output value of a sigmoidal gate models the current firing rate of a neuron in the biological system.

On the basis of currently available data one cannot argue that Assumptions (A) and (B) are wrong for all biological neural systems. There exists a large variety of different biological neural systems, and some may very well satisfy Assumption (A) or (B). For example, recent recordings from the olfactory system of the locust demonstrate the existence of a stimulus-evoked oscillation which may serve as a "clock" for that neural system [61]. On the other hand, during the last few years ample empirical evidence has accumulated which shows that many important biological neural systems satisfy neither Assumption (A) nor Assumption (B).

Until a few years ago most experimental and theoretical work in neurophysiology was based on Assumption (B), and apart from the work of Abeles and his group [3, 4] little effort was made to investigate whether individual firing times also carried information. One interesting case is documented in [8], where Bair and Koch have re-analyzed some older data where spike trains from a higher area (area MT) of the visual cortex of monkeys had been recorded while various random dot moving patterns were shown to the monkey. Originally, one had thought that only the firing rates carried information about the stimulus. But when Bair and Koch analyzed those spike trains that had been recorded in response to repetitions of the *same* random dot moving pattern (generated by the *same random seed*), a specific *temporal structure* became visible in the recorded spike trains that was *characteristic for the specific random dot moving pattern*.

A verification or falsification of Assumption (B) is complicated by the fact that the *same* biological neural system may employ *different coding mechanisms* for *different tasks*. Whereas firing rates may be used by a neural system to encode information about static or simple dynamic stimuli, a neural code that encodes additional information through the firing times of neurons may be used by the same system in response to more complex dynamic stimuli, as they occur in the natural environment [10]. In particular, a neural system may prefer a code where the timing of individual spikes matters for those tasks where it has to respond very fast (see e.g. [54]). It has been shown in [49] that the neuron H1 in the blowfly responds to a stimulus consisting of a pattern of random bars that moves across the visual field at *constant* velocity with a stochastic spike train whose individual spike times carry no information about the stimulus, wheras its spike frequency (i.e. firing rate) encodes information about the constant velocity of the stimulus. In contrast to that, the same neuron H1 responds to a stimulus consisting of the same pattern of random bars but now moving with a *time-dependent* velocity with a spike train whose spike times are much more reproducible, and in which on average the *timing* of each spike carries several bits of information about this more complex and more "natural" stimulus. In fact, it is demonstrated in [45] that one can "decode" the information contained in the firing times of this neuron H1, and thereby *reconstruct* with amazing fidelity the time series encoded in a visual stimulus with time-dependent velocity.

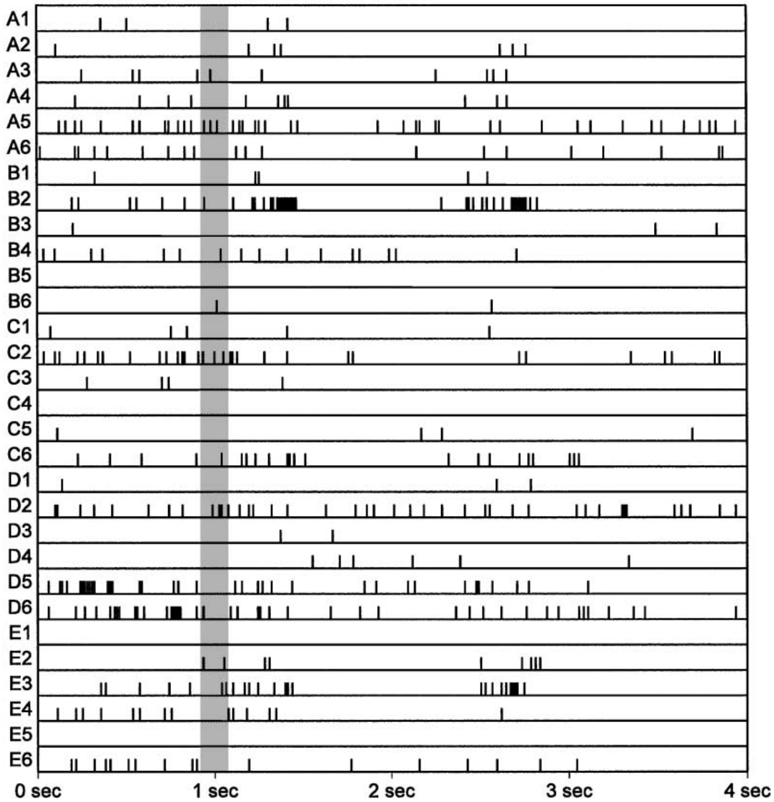

Fig. 2. Simultaneous recordings (over 4 s) of the firing times of 30 neurons from monkey striate cortex by Krüger and Aiple [24]. Each firing is denoted by a vertical bar, with a separate row for each neuron. For comparison we have shaded an interval of 150 ms. This time span is known to suffice for the completion of some complex multilayer cortical computations.

These data (and many other recordings) also show that typical firing rates of biological neurons are relatively low – typically well below 100 Hz. They are especially low in higher cortical areas of more advanced species. On the other hand complex multi-layer biological neural systems with at least 10 synaptic stages ("layers") are able to complete complex computations within just 150 ms [11, 54]. Each neuron can contribute at most 1 or 2 spikes to such computation. This is demonstrated in Fig. 2, which also shows that inter-spike intervals tend to be quite irregular. These data provide a contradiction to Assumption (B) for fast cortical computations, since a neuron in this system does not have time to "read" the firing rates of its predecessors before it produces its own output as contribution to a fast multi-layer computation. In addition, recent neurophysiological results [2] suggest that the dynamical behaviour of synapses (see Section 6) makes them not very suitable for analog neural computation in terms of firing rates.

Thus, we arrive at the challenge to understand a *new class of computational models* in which *time is used as a resource for coding information*. One might view these models as a *third generation of neural network models*, following threshold circuits as the first generation and sigmoidal neural nets as the second generation of neural network models. Obviously, computation and learning has to proceed quite differently in this new generation of neural network models. We have to rethink even the most basic concepts and algorithmic ideas, and provide new tools that are adequate for this unfamiliar computing environment. Apart from the goal of unraveling the structure of biological neural computation, this challenge also arises through experiments with a new generation of electronic hardware, where information is encoded in *temporal patterns* of electric pulses (see e.g. [32, 40, 42, 63]).

The goal of this survey article is to discuss formal models that capture various essential aspects of computation with temporal coding, and to give an overview of currently known theoretical results about computation and learning in such models. More detailed results can be found at http://www.cis.tu-graz.ac.at/igi/maass/. With regard to further details about biological neural systems we refer to Abeles [3], Arbib [7], Churchland [11], Johnston [20], Koch [21], Maass and Bishop [32], Rieke et al. [45], Shepherd [52] and Tuckwell [57].

2. A formal model for a network of spiking neurons

If one ignores all temporal aspects then a spiking neuron v has some structural similarity to the familiar threshold gate. A threshold gate outputs 1 if and only if the weighted sum of its inputs reaches some threshold Θ. The membrane potential P_v of a spiking neuron v can be modeled by a weighted sum of pulses, which result from the "firing" of other neurons. The neuron v "fires", i.e. v generates a spike which is propagated through its axon to other neurons, if this membrane potential P_v reaches a threshold Θ_v. The output of v consists of the set $F_v \subseteq \mathbf{R}^+$ of points in time when v "fires" (where $\mathbf{R}^+ = \{x \in \mathbf{R}\colon\ x \geqslant 0\}$).

In the simplest (deterministic) model of a spiking neuron one assumes that a neuron v fires whenever P_v (which models the electric membrane potential at the "trigger zone" of neuron v) reaches Θ_v. P_v is the sum of socalled excitatory postsynaptic potentials (EPSP's) and inhibitory postsynaptic potentials (IPSPs), which result from the firing of other neurons u that are connected through a "synapse" to neuron v. The firing of a "presynaptic" neuron u at time s contributes to the potential P_v at time t an amount that is modeled by the term $w_{u,v}(t) \cdot \varepsilon_{u,v}(t-s)$, which consists of a "*weight*" $w_{u,v}(t) \geqslant 0$ and a *response-function* $\varepsilon_{u,v}(t-s)$. Biologically realistic shapes of such response functions are indicated in Fig. 3. In most mathematical models for spiking neurons one ignores the time dependence of the factor $w_{u,v}(t)$, and views it as a constant $w_{u,v}$ which only changes on the slow time scale of learning. Thus $w_{u,v}$ corresponds to a "weight" in traditional models for artificial neural nets. For the sake of simplicity we will adopt this view for the moment. However, we will discuss in Section 6 new results which show that this view is quite problematic.

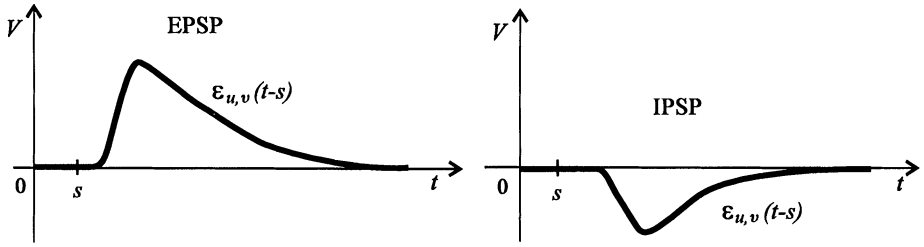

Fig. 3. Typical shape of response functions (EPSP and IPSP) of a biological neuron.

The restriction of $w_{u,v}$ to non-negative values (in combination with positive or negative response functions $\varepsilon_{u,v}(t-s)$) is motivated by the assumption that a biological synapse is either "excitatory" or "inhibitory", and that it does not change its "sign" in the course of a "learning-process". In addition, for most biological neurons u, either all response-functions $\varepsilon_{u,v}(t-s)$ for postsynaptic neurons v are "excitatory" (i.e. positive), or all of them are "inhibitory" (i.e. negative). Obviously, these constraints have little impact on theoretical complexity investigations (just consider pairs of excitatory and inhibitory neurons instead of single neurons), unless one cares about small constant factors in the size of networks, or one wants to model the actual architecture of cortical circuits (see [52]).

In a "typical" biological neuron the resting membrane potential is around -70 mV, the firing threshold of a "rested" neuron is around -50 mV, and a postsynaptic potential (i.e. EPSP or IPSP) changes the membrane potential temporarily by at most a few mV. However, it is mathematically more convenient to assume that the potential P_v has value 0 in the absence of postsynaptic potentials, and that the threshold value Θ_v is always >0. We adopt this normalization in the following.

We have already seen that P_v does not really correspond to the weighted sum of a threshold gate since it varies over time. The same holds true for the threshold Θ_v. If a neuron v has fired at time t', it will not fire again for a few ms after t', no matter how large its current potential $P_v(t)$ is ("*absolute refractory period*"). Then for a few further ms it is still "reluctant" to fire, i.e. a firing requires a larger value of $P_v(t)$ than usual ("*relative refractory period*"). Both of these refractory effects are modeled by a suitable "*threshold function*" $\Theta_v(t-t')$, where t' is the time of the most recent firing of v. A typical shape of the function $\Theta_v(t-t')$ for a biological neuron is indicated in Fig. 4. We assume that $\Theta_v(t-t')=\Theta_v(0)$ for large values of $t-t'$. In the deterministic (i.e. noise free) version of the spiking neuron model one assumes that v fires whenever $P_v(t)$ crosses from below the function $\Theta_v(t-t')$.

A formal *Spiking Neuron Network* (SNN) – which was introduced in [26, 27] – consists of a finite set V of *spiking neurons*, a set $E \subseteq V \times V$ of *synapses*, a *weight* $w_{u,v} \geqslant 0$ and a *response function* $\varepsilon_{u,v}: \mathbf{R}^+ \to \mathbf{R}$ for each synapse $\langle u,v \rangle \in E$, and a *threshold function* $\Theta_v: \mathbf{R}^+ \to \mathbf{R}^+$ for each neuron $v \in V$.

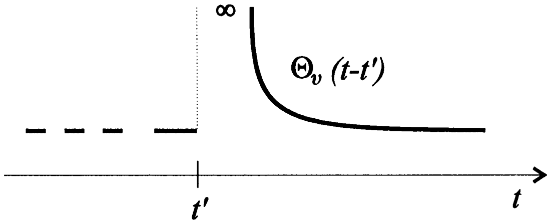

Fig. 4. Typical shape of the threshold function of a biological neuron.

If $F_u \subseteq \mathbf{R}^+$ is the set of *firing times* of a neuron u, then the *potential* at the trigger zone of neuron v at time t is given by

$$P_v(t) := \sum_{u:\langle u,v\rangle \in E} \ \sum_{s \in F_u : s<t} w_{u,v} \cdot \varepsilon_{u,v}(t-s).$$

In a noise-free model a neuron v fires at time t as soon as $P_v(t)$ reaches $\Theta_v(t-t')$, where t' is the time of the most recent firing of v.

For some specified subset $V_{\text{in}} \subseteq V$ of *input neurons* one assumes that the firing times ("spike trains") F_u for neurons $u \in V_{\text{in}}$ are not defined by the preceding convention, but are given from the outside as *input* to the network. The firing times F_v for all other neurons $v \in V$ are determined by the previously described rules, and the output of the network is given in the form of the spike trains F_v for the neurons v in a specified set of *output neurons* $V_{\text{out}} \subseteq V$.

Experiments have shown that in vitro biological neurons fire with slightly varying delays in response to repetitions of the same current injection. Only under certain conditions neurons are known to fire in a more reliable manner [39]. Therefore one also considers *noisy spiking neurons* [29, 28], where the difference $P_v(t) - \Theta_v(t-t')$ just governs the *probability* that neuron v fires at time t. The choice of the exact firing times is left up to some unknown stochastic processes, and it may for example occur that v does *not* fire in a time interval I during which $P_v(t) - \Theta_v(t-t') > 0$, or that v fires "spontaneously" at a time t when $P_v(t) - \Theta_v(t-t') < 0$.

The previously described noisy version of the SNN model is essentially equivalent to the *spike response model* in [16, 17], and to the other common mathematical models for networks of spiking neurons (see e.g. [3, 7, 57]). Subtle differences exist between these models with regard to their treatment of refractory effects and the "reset" of the membrane potential after a firing. An important advantage of the spike response model is its mathematical simplicity. This makes it suitable for an analysis of computations in networks of spiking neurons.

We would like to point out that the formal model for a spiking neuron that we have discussed so far is a coarse simplification. In particular, the membrane potential P_v at the trigger zone of a neuron v is in general not a *linear* sum of incoming pulses. Both *sublinear* and *superlinear* summation occur in biological neurons. We will discuss in Section 5 some possible computational consequences of these effects. In Section 6, we will discuss computational consequences of taking the *temporal dynamic of synapses* into account.

Sections 4–6 can be read independently.

3. Schemes for temporal coding in neural computation

If one accepts the premise that the firing times of neurons in a biological neural system encode relevant information, there are still many possible coding schemes that have to be distinguished.

One efficient way of encoding analog information in the firing times of a massively parallel neural system is through differences in the firing times of different neurons (sometimes referred to as "delay coding", "latency coding", or "firing order coding", [15, 19, 55]). For example a vector $\langle x_1, \ldots, x_n \rangle \in [0,1]^n$ can be encoded by the firing times $T - c \cdot x_i$ of n neurons, where T is some reference time – or simply the time when the last one of the n neurons fires (see [19]), and c is a suitable scaling factor. The neurophysiological evidence for the use of this coding method in biological neural systems is still rather sparse, partially due to the lack of data from experiments where one has been able to record spike trains from *many neurons in parallel*. However, this coding scheme has captured the attention of many researchers because it is very simple and because it is one of very few coding methods that might theoretically be used for *very fast* neural computation [26, 55]. We will discuss computation and learning with this neural code in Section 4.

There also exists substantial evidence that on a larger time scale statistical correlations between firing times of different neurons encode relevant information ("correlation coding"); see e.g. [12, 23, 44]. We will discuss a theoretical model for computation with this neural code in Section 5.

Another neural code is the one alluded to in Section 1: a spike train can encode a time series (i.e. a sequence of analog numbers) through its pattern of interspike intervals. There exists strong evidence that temporal coding schemes of this type are employed by a variety of biological neural systems (see [45]). But one encounters some obstacles if one wants to analyze the computational power of neural systems employing this code. One obstacle is the fact that we have in computational complexity theory no standard reference models for analyzing computations on *time series*. Another obstacle is the fact that the simple model for a network of spiking neurons from Section 3 is too crude to capture the essence of computations on spike trains. A synapse can no longer be modeled by a constant weight $w_{u,v}$ when a *sequence of spikes* with specific interspike intervals is processed by the synapse. We will discuss in Section 6 amendments to the model from Section 3 that make the model more suitable for investigating computations on *spike trains*.

We refer to Abbott [1], Abeles [3], Arbib [7], Koch [21], Maass and Natschläger [34], Phillips and Singer [44] and Rieke et al. [45] for further information about neural coding.

4. Neural computation with delay coding

We will show that networks of spiking neurons with delay coding have large computational power by proving that they can simulate arbitrary sigmoidal neural nets. The *key mechanism* for this simulation is based on the well-known fact that EPSPs and IPSPs are able to *shift* the firing time of a spiking neuron. This mechanism can be demonstrated very clearly in our formal model if one assumes that EPSPs rise (and IPSPs fall) *linearly* during a certain initial time period. Hence we assume in the following that there exists some constant $\Delta > 0$ such that each response function $\varepsilon_{u,v}(x)$ is of the form $sign_{u,v} \cdot (x - d_{u,v})$ for $x \in [d_{u,v}, d_{u,v} + \Delta]$ with $sign_{u,v} \in \{-1, 1\}$, and $\varepsilon_{u,v}(x) = 0$ for $x \in [0, d_{u,v}]$. The parameters $d_{u,v}$ corresponds to the time that passes between a firing of neuron u and the arrival of the resulting EPSP or IPSP at the soma of neuron v. The values of the functions $\varepsilon_{u,v}(x)$ for $x > d_{u,v} + \Delta$ are not relevant for the subsequent discussion. They describe an exponential decay as indicated in Fig. 3.

Consider a spiking neuron v that receives postsynaptic potentials from n presynaptic neurons $a_1, \ldots, a_n$. For simplicity we assume that interspike intervals are so large that the firing time t_v of neuron v depends just on a single firing time t_{a_i} of each neuron a_i, and Θ_v has returned to its "resting value" $\Theta_v(0)$ before v fires again. Then if the next firing of v occurs at a time when the postsynaptic potentials described by $w_{a_i,v} \cdot \varepsilon_{a_i,v}(t - t_{a_i})$ are all in their initial *linear* phase, its firing time t_v is determined in the noise-free model for $w_i := w_{a_i,v} \cdot sign_{a_i,v}$ by the equation $\sum_{i=1}^{n} w_i \cdot (t_v - t_{a_i} - d_{a_i,v}) = \Theta_v(0)$, or equivalently

$$t_v = \frac{\Theta_v(0)}{\sum_{i=1}^{n} w_i} + \frac{\sum_{i=1}^{n} w_i(t_{a_i} + d_{a_i,v})}{\sum_{i=1}^{n} w_i}. \tag{4.1}$$

This equation reveals the somewhat surprising fact that (for a certain range of their parameters) spiking neurons can compute a *weighted sum* in terms of *firing times* t_{a_i}. This mechanism can be employed to compute a weighted sum in delay coding. Finally, according to (4.1) the coefficients of the presynaptic firing times t_{a_i} are automatically *normalized*, which appears to be of biological interest. We may assume here that all delays $d_{a_i,v}$ have the same values. However, (4.1) also shows that differences in these delays could also be used to modulate the effective "weights" of the presynaptic neurons $a_1, \ldots, a_n$.

In the simplest scheme for delay coding (which is closely related to that in [19]) an analog variable $x_i \in [0, 1]$ is encoded by the firing time $T - c \cdot x_i$ of a neuron. In contrast to Hopfield [19] we consider here a computational model where both the inputs *and the outputs* of computations are encoded in this fashion. This has the advantage that one can *compose* computational modules.

We will first focus in Theorem 4.1 on the simulation of sigmoidal neural nets that employ the piecewise linear "linear saturated" activation function $\pi : \mathbf{R} \to [0, 1]$ defined by $\pi(y) = 0$ if $y < 0$, $\pi(y) = y$ if $0 \leqslant y \leqslant 1$, and $\pi(y) = 1$ if $y > 1$. The subsequent Theorem 4.2 will imply that one can simulate with spiking neurons also sigmoidal neural nets that employ *arbitrary* continuous activation functions.

Theorem 4.1. *For any given $\varepsilon, \delta > 0$ one can simulate any given feedforward sigmoidal neural net N consisting of s units with activation function π by a network $\mathcal{N}_{N,\varepsilon,\delta}$ of $s + \mathrm{O}(1)$ noisy spiking neurons with delay coding. More precisely, for any network input $x_1, \ldots, x_n \in [0,1]$ in delay coding the output of $\mathcal{N}_{N,\varepsilon,\delta}$ – given in delay coding – differs with probability $\geqslant 1 - \delta$ by at most ε from that of N. Furthermore, the computation time of $\mathcal{N}_{N,\varepsilon,\delta}$ depends neither on the number of gates in N nor on the parameters ε, δ, but only on the number of layers of the sigmoidal neural network N.*

It is known [25] that feedforward sigmoidal neural nets whose gates employ the activation function π can approximate with a single hidden layer for any $n, k \in \mathbf{N}$ any given continuous function $F : [0,1]^n \rightarrow [0,1]^k$ within any $\varepsilon > 0$ with regard to the L_∞-norm (i.e. uniform convergence). Hence we can derive the following result from Theorem 4.1.

Theorem 4.2. *Any given continuous function $F : [0,1]^n \rightarrow [0,1]^k$ can be approximated within any given $\varepsilon > 0$ with arbitrarily high reliability in delay coding by a network of noisy spiking neurons with a single hidden layer (and hence within 20 ms for biologically realistic values of their time-constants).*

Because of its generality this theorem implies the same result also for *more general schemes of coding analog variables by the firing times of neurons*, besides the particular scheme for delay coding that we have considered so far. In fact, it implies that the same result holds for any other coding scheme C that is "continuously related" to the previously considered one in the sense that the transformation between firing times that encode a vector $\langle x_1, \ldots, x_n \rangle$ of analog variables, in the coding scheme considered here and in the coding scheme C can be described by uniformly continuous functions in both directions.

Finally, we show that the converse of Theorem 4.1 is not true: one cannot simulate any network of s spiking neurons that employ delay coding by a network of $\mathrm{O}(s)$ sigmoidal neurons, in fact not even by a network with polynomially in s many sigmoidal neurons.

We consider the "*element distinctness function*" $ED_n : (\mathbf{R}^+)^n \rightarrow \{0,1\}$ defined by

$$ED_n(s_1, \ldots, s_n) = \begin{cases} 1 & \text{if } s_i = s_j \text{ for some } i \neq j, \\ 0 & \text{if } |s_i - s_j| \geqslant 1 \text{ for all } i,j \text{ with } i \neq j, \\ \text{arbitrary} & \text{else.} \end{cases}$$

If one encodes the value of input variable s_i by a firing of input neuron a_i at time $T - c \cdot s_i$, then for sufficiently large values of the constant $c > 0$ *a single noisy spiking neuron v can compute ED_n* with arbitrarily high reliability. This holds for any reasonable type of response functions, e.g. the ones shown in Fig. 3. The binary output of this computation is assumed to be encoded by the firing/non-firing of v. Hair-trigger situations are avoided since no assumptions have to be made about the firing or non-firing of v if EPSPs arrive with a temporal distance *between* 0 and c.

On the other hand, the following result shows that a fairly large sigmoidal neural net is needed to compute the same function ED_n. Its proof provides the first application for Sontag's recent result [53] about a new type of "dimension" d of a neural network N, where d is chosen maximal so that *every* subset of d inputs is shattered by N. Furthermore, it expands a method due to Koiran [22] for using the VC-dimension to prove lower bounds on network size.

Theorem 4.3. *Any sigmoidal neural net* $\mathcal{N}$ *that computes* ED_n *has at least* $(n-4)/2-1$ *hidden units.*

This result provides the largest known lower bound for *any* concrete function with n inputs on a sigmoidal neural net. The largest previously known lower bound for sigmoidal neural nets was $\Omega(n^{1/4})$, due to Koiran [22].

We refer to Maass [26, 30] for details of these results. Further theoretical results about neural computation with delay coding can be found in [19, 27, 28, 33, 35, 48]. Several of these theoretical predictions have been tested through computer simulations with more detailed models for biological neurons [33, 43, 46].

With regard to *learning* in the context of neural computation with delay coding the following issue arises: besides the formal weights $w_{u,v}$ of synapses also the *transmission delays* $d_{u,v}$ between neurons u and v determine the function that is computed by a network of spiking neurons with delay coding. There exists substantial evidence (see e.g. [5, 18]) that these delays $d_{u,v}$ are tuned by adaptive mechanisms in biological neural systems. In [36] it is shown that the VC-dimension of a neuron grows faster in terms of its number of programmable delays than in its number of programmable weights: a spiking neuron with n programmable *delays* may have VC-dimension $\Omega(n \log n)$, whereas the VC-dimension of a threshold gate can grow only linearly with its number of *weights*. Corresponding results are shown in [37] for networks of spiking neurons. These results can be interpreted as evidence that tuning of delays in a network of spiking neurons may contribute even more to the diversity of functions computed by the system than changing its synaptic weights. This suggests that one should investigate learning algorithms for tuning the delays in network of spiking neurons. Some first results can be found in [18, 36, 37, 43, 47].

Zador and Pearlmutter [62] have investigated the VC-dimension of spiking neurons in terms of another class of parameters that are relevant for neural computation: the integration time constants and the firing threshold of spiking neurons.

5. Neural coding through correlations in firing times

A number of results in experimental neurophysiology suggest that *correlations* between firing times of neurons are relevant for neural coding and computation (see for example [12, 23, 44, 58]). These data show that for example two neurons may partially synchronize their firing times (without changing their firing rates) in response to a particular visual or auditory stimulus. Note that this is a method for using time as a resource for encoding information which strongly differs from the preceding one: individual firing times are not too critical here since no perfect correlation in firing times of different neurons is required to convey relevant information, just a statistically significant correlation is needed. Hence this coding mechanism is quite noise robust. This coding mechanism is also interesting from the conceptual point of view, since it provides the means for carrying output analog computation with direct codes for "fuzzy" *relations*.

Several models that reflect computational effects of firing correlations in a simplified setting have already previously been proposed [3, 14, 41, 44, 51, 60]. Note that more detailed models for networks of spiking neurons are less suitable for analyzing the computational power of computations with large scale statistical correlations in firing times, since they keep track of too many details. We consider in this section a new approach towards modeling neural computation with firing rates and correlations in firing times. We introduce a simple extension of the familiar neural network models from the first two generations (i.e., of threshold circuits and sigmoidal neural nets) that allows us to model salient computational features of correlations between firing times in this simplified setting. We have seen in Section 2 that there exist substanially more detailed models for networks of spiking neurons. But these are less suitable for analyzing the computational power of computations with large scale statistical correlations in firing times, since they keep track of too many details.

We assume that an arbitrary directed graph is given that describes the architecture of a neural network $\mathcal{N}$. We reserve for each gate (or "unit") u of the network a formal variable $o(u)$ which denotes the output of u. In a biological interpretation this variable $o(u)$ models the current firing rate of a neuron u. We assume in our extended neural network model that one has in addition to the formal variables $o(u)$ for each gate u of $\mathcal{N}$ a second type of formal variable $c(S)$ for various *sets* S of gates in $\mathcal{N}$. In a biological interpretation the formal variable $c(S)$ models the current correlation in the firing times of neurons in this set S. A characteristic feature of this new type of variables is that no additional computational units are needed to compute their values.

Each gate v of the network $\mathcal{N}$ receives both types of variables as input, i.e. in addition to the variables $o(u)$ for immediate predecessors u it also receives the variables $c(S)$ for subsets S of its set of immediate predecessors. Thus, besides the "real" variables $o(u)$, gates in $\mathcal{N}$ also handle a second type of "imaginary" variables $c(S)$. Furthermore, the computational operation of a gate v cannot be decomposed into its operation on "real" variables and its operation on "imaginary" variables, since both its "real" output $o(v)$ and its "imaginary" output $c(T)$ depend on *both* types of input variables. Hence the computational operation of such gate v is reminiscent of a *complex* function in mathematics (for example $z \mapsto \mathrm{e}^z$, where the real and imaginary component of its output e^z for an input $z = x + \mathrm{i}y$ depend on both x and y). Because of this loose analogy we refer to the new type of gates that we are introducing as *complex* gates.

The "real" component $o(v)$ of the output of a complex gate v is described by the equation

$$o(v) = \sigma\left(\sum_{u \in U} \alpha_{uv} o(u) + \sum_{S \subseteq U} \alpha_{Sv} c(S) \prod_{u \in S} o(u) + \alpha_v\right), \tag{5.2}$$

where U is the set of immediate predecessors of v in the directed graph that describes the architecture of the network $\mathcal{N}$. The parameters α_{uv}, α_{Sv} and α_v may have arbitrary real values. The parameter α_{Sv} scales the impact that correlated firing of the neurons in S have on the firing rate of v. In a biological interpretation the value of α_{Sv} depends on the locations of the synapses between neurons $u \in S$ and v on the dendritic tree or soma of the postsynaptic neuron v. In addition, it depends on the biochemical structure of these synapses, and on the distribution of voltage-dependent channels on the dendritic tree of v. For example, if all neurons in S have synapses close together on the dendritic tree of v, and if there exists an accumulation of voltage-dependent channels in a close-by branching point on the way to the soma of v (so that a "dendritic spike" can be generated at such "hot spot"), then an increase in the firing correlation of the neurons in S is likely to have a significant impact on the firing rate of neuron v, and α_{Sv} should be given a relatively large value. We refer to Bernander et al. [9], Johnston and Wu [20], Koch [21] and Shepherd [52] for further details about dendritic integration.

The letter σ in (5.2) denotes some arbitrary activation function $\sigma: \mathbf{R} \to \mathbf{R}$, that can be chosen as in traditional neural network models. If σ is the sign-function then we refer to the gate v as a *complex threshold gate*.

The next equation describes the "imaginary" part of the output of the same complex gate v, i.e. the values of the variables $c(T)$ for arbitrary sets T of gates in $\mathcal{N}$ with $v \in T$:

$$c(T) = \sigma_c\left(\sum_{u \in U} \alpha_{uT} o(u) + \sum_{S \subseteq U} \alpha_{ST} c(S) \prod_{u \in S} o(u) + \alpha_T\right). \tag{5.3}$$

In a biological interpretation the parameters α_{uT} and α_{ST} in Eq. (5.3) can be used to model details of the geometrical and biochemical structure of the neurons in T and U. The first term $\sum_{u \in U} \alpha_{uT} o(u)$ in (5.3) reflects the fact that the firing correlation of the neurons in T can be increased through common input from a neuron u. Hence, the value of α_{uT} should be chosen positive if neuron u has excitatory synapses to all neurons in T. The second term $\sum_{S \subseteq U} \alpha_{ST} c(S) \prod_{u \in S} o(u)$ in (5.3) reflects an alternative way in which firing correlation among neurons in T can be achieved: if each neuron in T receives input from some neuron $u \in S$ where S is a set of presynaptic neurons that fire with a fairly large firing correlation (i.e., $c(S) \prod_{u \in S} o(u)$ is large). The parameter α_{ST} depends on the connectivity structure between the neuron sets S and T, and on biochemical details of their synapses and of the dendritic trees of the neurons in T. It is not yet known how many of the parameters $\alpha_{vu}, \alpha_{Sv}, \alpha_{uT}, \alpha_{ST}$ can be chosen independently for a biological neuron.

If one assumes that the firing rate $o(u)$ of each neuron u is proportional to the *probability* $p_u(I)$ that neuron u fires during some fixed short time-window I (say of length 2 ms), and if the spike trains from a set S of such neurons u can be modeled by *independent* stochastic processes, then the probability that all neurons $u \in S$ fire during this time-window I is given by $\prod_{u \in S} p_u(I)$. Hence one can for example interpret the value of the variable $c(S)$ as

$$\frac{Pr[\text{all neurons } u \in S \text{ fire during } I]}{\prod_{u \in S} p_u(I)}. \tag{5.4}$$

Then $c(S) = 1$ if the neurons $v \in S$ fire independently. Furthermore, the term $c(S) \prod_{u \in S} o(u)$ in Eqs. (5.2) and (5.3) is proportional to Pr[all neurons $u \in S$ fire during I]. This is desirable since an increase in Pr[all neurons $u \in S$ fire during I] is likely to increase the firing rate $o(v)$ of a neuron v (if all neurons in S are connected to v) and the correlation $c(T)$ of a set T of subsequent neurons to which the neurons in S are connected.

The following result provides some first information about the computational power of our model for computation with firing rates and firing correlations. Let $\mathscr{C}$ be some arbitrary class of nonempty subsets of $\{u_1, \ldots, u_n\}$. We consider the boolean function $F_{\mathscr{C}} : \{0,1\}^m \to \{0,1\}$ for $m := n + |\mathscr{C}|$. $F_{\mathscr{C}}$ gives for input vectors $\langle x_i \rangle_{i=1,\ldots,n} {}^\frown \langle x_S \rangle_{S \in \mathscr{C}}$ from $\{0,1\}^m$ the output 1 if and only if $x_S \prod_{u_i \in S} x_i = 1$ for some $S \in \mathscr{C}$. The operation "$^\frown$" denotes the concatenation of vectors. The logarithm that occurs in the subsequent bound is taken with regard to base 2.

Theorem 5.1. *The function $F_{\mathscr{C}} : \{0,1\}^m \to \{0,1\}$ can be computed by a single complex threshold gate.*

Let $\tilde{\mathscr{C}} \subseteq \mathscr{C}$ be an arbitrary collection of sets in $\mathscr{C}$ so that no $S, S' \in \tilde{\mathscr{C}}$ exist with $S \subsetneq S'$. Then any feedforward threshold circuit computing $F_{\mathscr{C}}$ needs to have at least $|\tilde{\mathscr{C}}| / \log(|\tilde{\mathscr{C}}| + 1)$ gates, and any feedforward sigmoidal neural net computing $F_{\mathscr{C}}$ with piecewise rational activation functions needs to have $\Omega(|\tilde{\mathscr{C}}|^{1/2})$ gates.

In [31] we discuss applications of Theorem 5.1 to concrete classes $\mathscr{C}$ of subsets of $\{u_1, \ldots, u_n\}$, for which the computation of the associated boolean function $F_{\mathscr{C}}$: $\{0,1\}^{n+|\mathscr{C}|} \to \{0,1\}$ appears to be of interest in the context of biological neural systems.

The computational model that we have sketched in this section also throws new light on questions about learning in neural systems. Besides the parameters α_{uv} that correspond to the weights in traditional neural network models, there emerge new types of parameters $\alpha_{Sv}, \alpha_{uT}, \alpha_{ST}$ that are relevant for computations with firing correlations. So far nothing is known about suitable learning algorithms for tuning these parameters.

Details to the results of this section can be found in [31].

6. Synapses are not static during a computation

So far we have modeled the computational role of a synapse as a multiplication with a static scalar parameter: the "weight" of that synapse. However, biological synapses are in fact rather complex computational devices. When a spike arrives at a presynaptic terminal, it sometimes triggers the *release* of a neurotransmitter-filled packet or *vesicle* of neurotransmitter from a release site, but it also sometimes *fails* to trigger a release. Only when a vesicle is released does the signal from the presynaptic neuron propagate to the postsynaptic neuron. The probability that a presynaptic action potential fails to elicit a postsynaptic response is different at different synapses, and ranges in the hippocampus from less than 0.1–0.9 or higher [6]. Moreover, the release probability at a synapse is strongly modulated by the recent history of activity at that synapse. This modulation occurs through a variety of distinct biophysical mechanisms including *facilitation* (i.e., a transient increase in release probability) and *depression* (i.e., a transient decrease in release probability) [64]. Previous formal models [2, 50, 56] account only for the average probability across a population of synapses.

In this section, which describes a part of recent joint work with Zador [38], we will be interested in the computational implications of the more complex model of a synapse that is emerging. For this we consider the biophysical mechanisms underlying the modulation of probability at individual release sites. We assume that release probability is determined by two distinct processes: facilitation, and depression. Facilitation in our model depends on the number of spikes that arrive at the presynaptic terminal. This is consistent with a mechanism in which the facilitation is determined by the amount of calcium that has entered the presynaptic terminal [64], which in turn depends on the number of spikes that invade the presynaptic terminal. Depression in our model depends on the number of vesicles actually released by the synapse; such a dependence could arise for example if depression was the result of depletion of the pool of readily releasable vesicles [13]. The resulting history-dependent temporal development of the release probability of a synapse through the competition between these two complementary mechanisms can be quote complex as the subsequent formal analysis shows.

A synapse S carries out computations on spike trains, more precisely on trains of spikes that arrive at the presynaptic terminal. A spike train is represented as a sequence $\underline{t}$ of firing times, i.e. as increasing sequences of numbers $t_1 < t_2 < \cdots$ from $\mathbf{R}^+$. If the time t_1 of the first spike in the spike train $\underline{t}$ is irrelevant, one can represent $\underline{t}$ just as well by the associated sequences $I_1, I_2, \ldots$ of interspike intervals I_j, which are defined by $I_j := t_{j+1} - t_j$. For each spike train $\underline{t}$ the output of synapse S consists of the sequence $S(\underline{t})$ of those $t_i \in \underline{t}$ on which vesicles are "released" by S, i.e. of those $t_i \in \underline{t}$ which cause an excitatory or inhibitory postsynaptic potential (EPSP or IPSP, respectively). The map $\underline{t} \to S(\underline{t})$ may be viewed as a stochastic function that is *computed* by synapse S. Alternatively, one can characterize the output $S(\underline{t})$ of a synapse S through its *release pattern* $\underline{q} = q_1 q_2 \ldots \in \{R, F\}^*$, where R stands for release and F for failure of release. For each $t_i \in \underline{t}$ one sets $q_i = R$ if $t_i \in S(\underline{t})$, and $q_i = F$ if $t_i \notin S(\underline{t})$.

The central Equation in our dynamic synapse model gives the probability $p_S(t_i)$ that the ith spike in a presynaptic spike train $\underline{t} = \langle t_1, \ldots, t_k \rangle$ triggers the release of vesicle at time t_i at synapse S,

$$p_S(t_i) = 1 - \mathrm{e}^{-C(t_i) \cdot V(t_i)}. \tag{6.5}$$

The functions $C(t) \geqslant 0$ and $V(t) \geqslant 0$ describe, respectively, the states of facilitation and depletion at the synapse at time t.

The dynamics of facilitation are given by

$$C(t) = C_0 + \sum_{t_i < t} c(t - t_i), \tag{6.6}$$

where C_0 is some parameter $\geqslant 0$ that can for example be related to the resting concentration of calcium in the synapse. The exponential response function $c(s)$ models the response of $C(t)$ to a presynaptic spike that had reached the synapse at time $t - s$:

$$c(s) = \alpha \mathrm{e}^{-s/\tau_C}, \tag{6.7}$$

where the positive parameters τ_C and α give the decay constant and magnitude, respectively, of the response. The function C models in an abstract way internal synaptic processes underlying presynaptic facilitation, such as the concentration of calcium in the presynaptic terminal. The particular exponential form used for $c(s)$ could arise for example if presynaptic calcium dynamics were governed by a simple first-order process.

The dynamics of depletion are given by

$$V(t) = \max\left(0, V_0 - \sum_{t_i : t_i < t \text{ and } t_i \in S(\underline{t})} v(t - t_i)\right) \tag{6.8}$$

for some parameter $V_0 > 0$. $V(t)$ depends on the subset of those $t_i \in \underline{t}$ with $t_i < t$ on which vesicles were actually released by the synapse, i.e. $t_i \in S(\underline{t})$. The function $v(s)$ models the response of $V(t)$ to a preceding release of the same synapse at time $t - s \leqslant t$. Analogously, as for $c(s)$ one may choose for $v(s)$ a function with exponential decay

$$v(s) = \mathrm{e}^{-s/\tau_V}, \tag{6.9}$$

where $\tau_V > 0$ is the decay constant. The function V models in an abstract way internal synaptic processes that support presynaptic depression, such as depletion of the pool of readily releasable vesicles. In a more specific synapse model one could interpret V_0 as the maximal number of vesicles that can be stored in the readily releasable pool, and $V(t)$ as the expected number of vesicles in the readily releasable pool at time t.

In summary, the model of synaptic dynamics presented here is described by five parameters: C_0, V_0, τ_C, τ_V and α. The dynamics of a synaptic computation and its internal variables $C(t)$ and $V(t)$ is indicated in Fig. 5.

Theorem 6.1. *Let $\langle t_1, t_2 \rangle$ be some arbitrary spike train consisting of two spikes, and let $p_1, p_2 \in (0, 1)$ be some arbitrary given numbers with $p_2 > p_1(1 - p_1)$. Furthermore,*

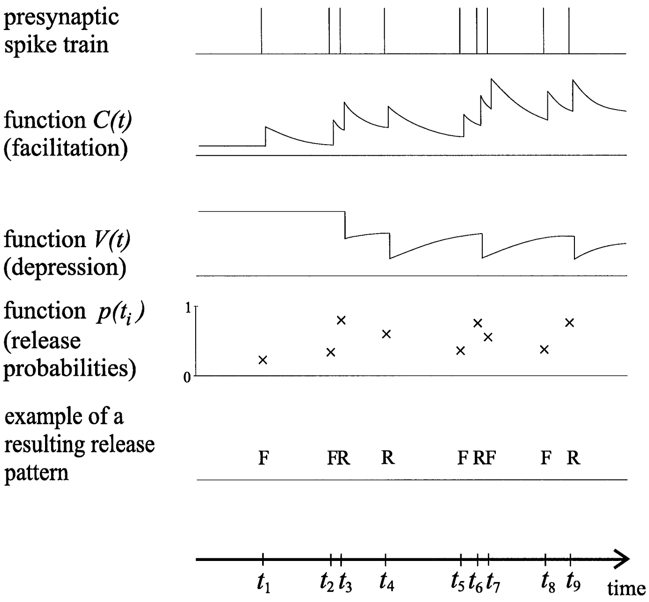
presynaptic spike train
function C(t) (facilitation)
function V(t) (depression)
function p(t_i) (release probabilities)
1
0
example of a resulting release pattern
F FR R F RF F R
t1 t2 t3 t4 t5 t6 t7 t8 t9
time

Fig. 5. Synaptic computation on a spike train $\underline{t}$, together with the temporal dynamics of the internal variables C and V of our model. Note that $V(t)$ changes its value only when a presynaptic spike causes release.

assume that arbitrary positive values are given for the parameters α, τ_C, τ_V of a synapse S. Then one can always find values for the two parameters C_0 and V_0 of the synapse S so that $p_S(t_1) = p_1$ and $p_S(t_2) = p_2$.

Furthermore the condition $p_2 > p_1(1 - p_1)$ is necessary in a strong sense. If $p_2 \leqslant p_1 (1 - p_1)$ then no synapse S can achieve $p_S(t_1) = p_1$ and $p_S(t_2) = p_2$ for any spike train $\langle t_1, t_2 \rangle$ and for any values of its parameters $C_0, V_0, \alpha, \tau_C, \tau_V$.

If one associates the current sum of release probabilities of multiple synapses between two neurons u and v with the current value of the "connection strength" $w_{u,v}$ between two neurons in a formal neural network model, then the preceding result points to a significant difference between the dynamics of computations in biological circuits and formal neural network models. Whereas in formal neural network models it is commonly assumed that the value of a synaptic weight stays fixed during a computation, the release probabilities of synapses in biological neural circuits may change on a fast time scale within a single computation.

We now look at the variety of release patterns that a synapse can produce for spike trains $t_1, t_2, t_3, \ldots$ with at least three spikes. We show that a synapse with a *fixed* parameter setting can respond quite differently to spike trains with different interspike intervals. Hence a synapse can serve as *pattern detector* for temporal patterns in spike trains.

Fig. 6A shows the most likely release pattern for each given pair of interspike intervals $\langle I_1, I_2 \rangle$, given a particular *fixed* set of synaptic parameters. One can immediately

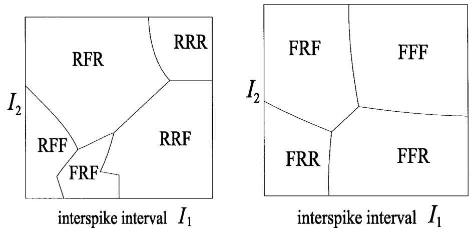

Fig. 6. (A, left) Most likely release pattern of a synapse in dependence of the interspike intervals I_1 and I_2. The synaptic parameters are $C_0 = 1.5$, $V_0 = 0.5$, $\tau_C = 5$, $\tau_V = 9$, $\alpha = 0.7$. (B, right) Release patterns for a synapse with other values of its parameters ($C_0 = 0.1$, $V_0 = 1.8$, $\tau_C = 15$, $\tau_V = 30$, $\alpha = 1$).

see that a synapse with fixed parameter values is likely to respond quite differently to spike trains with different interspike intervals. For example even if one just considers spike trains with $I_1 = I_2$ one moves in Fig. 6A through three different release patterns that take their turn in becoming the most likely release pattern when I_1 varies. Similarly, if one only considers spike trains with a fixed time interval $t_3 - t_1 = I_1 + I_2 = \Delta$, but with different positions of the second spike within this time interval of length Δ, one sees that the most likely release pattern is quite sensitive to the position of the second spike within this time interval Δ.

Fig. 6B shows that a different set of synaptic parameters gives rise to a completely different assignment ofrelease patterns.

We show in the next Theorem that the boundaries between the zones in these figures are "plastic": by changing the values of C_0, V_0, α the synapse can move the zone for most of the release patterns $\underline{q}$ to any given point $\langle I_1, I_2 \rangle$. This result provides an example for a new type of synaptic plasticity that can no longer be described in terms of a decrease or increase of the synaptic "weight".

Theorem 6.2. *Assume that an arbitrary number $p \in (0,1)$ and an arbitrary pattern $\langle I_1, I_2 \rangle$ of interspike intervals is given. Furthermore, assume that arbitrary fixed positive values are given for the parameters τ_C and τ_V of a synapse S. Then for any pattern $\underline{q} \in \{R,F\}^3$ except RRF, FFR one can assign values to the other parameters α, C_0, V_0 of this synapse S so that the probability of release pattern $\underline{q}$ for a spike train with interspike intervals I_1, I_2 becomes larger than p.*

The proof of Theorem 6.2 in [38] is *constructive* in the sense that it exhibits constructive methods for the assignment of suitable values to the parameters α, C_0, V_0 (an analogous remark also holds for Theorem 6.1). It is also shown in [38] that the preceding result is not valid for the two patterns RRF and FFR.

Finally, we show that the computational power of a spiking (e.g. integrate-and-fire) neuron with stochastic dynamic synapses is strictly larger than that of a spiking neuron with traditional "static" synapses. Let T be a some given time window, and consider the computational task of detecting whether at least one of n presynaptic neurons $a_1, \ldots, a_n$ fire at least twice during T ("burst detection"). To make this task computationally feasible we assume that none of the neurons $a_1, \ldots, a_n$ fires outside of this time window.

Theorem 6.3. *A spiking neuron v with stochastic dynamic synapses can solve this burst detection task* (*with arbitrarily high reliability*). *On the other hand, no spiking neuron with static synapses can solve this task* (*for any assignment of* "*weights*" *to its synapses.*) [1]

7. Conclusions

We have shown that computations in models for biological neural systems differ in essential aspects from computations in artificial neural networks. Many of these differences result from the more central role that *time* plays in biological neural systems.

This has drastic consequences for questions about *learning* in biological neural systems. If one takes the inherent temporal dynamics of neurons and synapses into account, then one can no longer characterize the computational role of a synapse by a *single parameter* (its "weight"). Rather, other types of parameters emerge that determine the role of a synapse in a neural computation (for example the parameters $C_0, V_0, \tau_C, \tau_V, \alpha$ that we have described in our formal synapse model in Section 6). In addition, we have demonstrated in Sections 4 and 5 that various other "programmable parameters" are relevant for shaping the structure of a neural computation, which have no relationship to "weights" in traditional neural network models. Examples are the *transmission delays* between neurons (see Section 4), and various parameters that depend on the *geometrical arrangement* of synapses on the dendritic tree of a neuron (see Section 5).

None of the abovementioned new types of parameters that govern biological neural computation correspond to any of the parameters that have been considered in the context of traditional neural network models. This suggests that "learning" in biological neural systems may employ rather different mechanisms and algorithms.

The results reported in this article indicate that the organization of computing and learning in biological neural systems differs strongly from that in our current generation of artificial computing machinery. We hope that theoretical computer science will play an essential role in the transmission of new insight from biological neural systems to the design of improved artificial computing machinery.

References

[1] L.F. Abbott, Decoding neuronal firing and modelling neural networks, Quart. Rev. Biophys. 27(3) (1994) 291–331.

[1] We assume here that neuronal transmission delays differ by less than $(n-1)T$, where by *transmission delay* we refer to the temporal delay between the firing of the presynaptic neuron and its effect on the postsynaptic target.

[2] L.F. Abbott, J.A. Varela, K.A. Sen, S.B. Nelson, Synaptic depression and cortical gain control, Science 275 (1997) 220–224.

[3] M. Abeles, Corticonics: Neural Circuits of the Cerebral Cortex, Cambridge University Press, Cambridge, 1991.

[4] M. Abeles, H. Bergman, E. Margalit, E. Vaadia, Spatiotemporal firing patterns in frontal cortex of behaving monkeys, J. Neurophys. 70(4) (1993) 1629–1638.

[5] H. Agmon-Snir, I. Segev, Signal delay and input synchronization in passive dendritic structures, J. Neurophys. 70 (1993) 2066–2085.

[6] C. Allen, C.F. Stevens, An evaluation of causes for unreliability of synaptic transmission, Proc. Natl. Acad. Sci. USA 91 (1994) 10 380–10 383.

[7] M.A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, 1995.

[8] W. Bair, C. Koch, Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey, Neural Comput. 8 (1996) 1185–1202.

[9] O. Bernander, C. Koch, M. Usher, The effect of synchronized inputs at the single neuron level, Neural Comput. 6 (1994) 622–641.

[10] G. Buracas, A. Zador, M. DeWeese, T. Albright, Efficient discrimination of temporal patterns by motion-sensitive neurons in primate visual cortex, Neuron 20 (1998) 959–969.

[11] P.S. Churchland, T.J. Sejnowski, The Computational Brain, MIT-Press, Cambridge, MA, 1993.

[12] R.C. deCharms, M.M. Merzenich, Primary cortical representation of sounds by the coordination of action-potential timing, Nature 381 (1996) 610–613.

[13] L. Dobrunz, C. Stevens, Heterogeneity of release probability, facilitation and depletion at central synapses, Neuron 18 (1997) 995–1008.

[14] R. Eckhorn, H.J. Reitboeck, M. Arndt, P. Dicke, Feature linking via synchronization among distributed assemblies: simulations of results from cat visual cortex, Neural Comput. 2 (1990) 293–307.

[15] T.J. Gawne, T.W. Kjaer, B.J. Richmond, Latency: another potential code for feature binding in striate cortex, J. Neurophysiol. 76(3) (1996) 1356–1360.

[16] W. Gerstner, Time structure of the activity in neural network models, Phys. Rev. E. 51 (1995) 738–758.

[17] W. Gerstner, J.L. van Hemmen, How to describe neuronal activity: spikes, rates, or assemblies? in: J.D. Cowan, G. Tesauro, D. Alspector (Eds.), Advances in Neural Information Processing Systems, vol. 6, Morgan Kaufmann, San Mateo, 1994, pp. 463–470.

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

[19] J.J. Hopfield, Pattern recognition computation using action potential timing for stimulus representations, Nature 376 (1995) 33–36.

[20] D. Johnston, S.M. Wu, Foundations of Cellular Neurophysiology, MIT-Press, Cambridge, MA, 1995.

[21] C. Koch, Biophysics of Computation: Information Processing in Single Neurons, Oxford University Press, Oxford, 1999.

[22] P. Koiran, VC-dimension in circuit complexity, Proc. 11th IEEE Conf. on Computational Complexity, 1996, pp. 81–85.

[23] A.K. Kreiter, W. Singer, Stimulus-dependent synchronization of neuronal responses in the visual cortex of the awake macaque monkey, J. Neurosci. 16(7) (1996) 2381–2396.

[24] J. Krüger, F. Aiple, Multielectrode investigation of monkey striate cortex: spike train correlations in the infragranular layers, J. Neurophysiol. 60 (1988) 798–828.

[25] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Networks 6 (1993) 861–867.

[26] W. Maass, On the computational complexity of networks of spiking neurons, in: G. Tesauro, D.S. Touretzky, K.L. Todd (Eds.), Advances in Neural Information Processing Systems, vol. 7, MIT Press, Cambridge, MA, 1995, pp. 183–190.

[27] W. Maass, Lower bounds for the computational power of networks of spiking neurons, Neural Comput. 8(1) (1996) 1–40.

[28] W. Maass, On the computational power of noisy spiking neurons, in: Advances in Neural Information Processing Systems, vol. 8, MIT Press, Cambridge, MA, 1996, pp 211–217.

[29] W. Maass, Fast sigmoidal networks via spiking neurons, Neural Comput. 9 (1997) 279–304.

[30] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural Networks 10(9) (1997) 1659–1671.

[31] W. Maass, A simple model for neural computation with firing rates and firing correlations, Network: Comput. Neural Systems 9(3) (1998) 381–397.

[32] W. Maass, C.M. Bishop (Eds.), Pulsed Neural Networks, MIT Press, Cambridge, MA, 1999.

[33] W. Maass, T. Natschläger, Networks of spiking neurons can emulate arbitrary Hopfield nets in temporal coding, Proc. 6th Annu. Conf. on Computational Neuroscience 1997 (CNS'97) in Big Sky, Montana, USA, 1998, pp. 221–226; Network: Comput. Neural Systems 8(4) (1997) 355–372.

[34] W. Maass, T. Natschläger, A model for fast analog computation based on unreliable synapses, Neural Comput., in press.

[35] W. Maass, B. Ruf, On computation with pulses, Inform. Comput. 148 (1999) 202–218.

[36] W. Maass, M. Schmitt, On the complexity of learning for a spiking neuron, Proc. 10th Conf. on Computational Learning Theory 1997, ACM Press, New York, 1997, pp. 54–61; see also Electron Proc. 5th Internat. Symp. on Artificial Intelligence and Mathematics, http://rutcor.rutgers.edu/ ∼amai/1998.

[37] W. Maass, M. Schmitt, On the complexity of learning for spiking networks, temporal coding, Inform. Comput. 153 (1999) 26–46.

[38] W. Maass, A.M. Zador, Dynamic stochastic synapses as computational units, in: M.I. Jordan, M.J. Kearns, S.A. Solla (Eds.), Advances in Neural Information Processing Systems, vol. 10, 1998, pp. 194 –200; Neural Comput., 11 (1999) 903–917.

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

[40] C. Mead, Analog VLSI and Neural Systems, Addison-Wesley, Reading, MA, 1989.

[41] P.M. Milner, A model for visual shape recognition, Psychol. Rev. 81(6) (1974) 521–535.

[42] A. Murray, L. Tarassenko, Analogue Neural VLSI: A Pulse Stream Approach, Chapman & Hall, London, 1994.

[43] T. Natschläger, B. Ruf, Learning radial basis functions with spiking neurons using action potential timing, in: L.S. Smith, A. Hamilton (Eds.), Neuromorphic Systems: Engineering Silicon from Neurobiology, World Scientific, Singapore, 1998.

[44] W.A. Phillips, W. Singer, In search of common foundations for cortical computation, Behav. Brain Sci. 20 (1997) 657–722.

[45] F. Rieke, D. Warland, R. van Stevenick, W. Bialek, SPIKES: Exploring the Neural Code, MIT Press, Cambridge, MA, 1996.

[46] B. Ruf, Computing functions with spiking neurons in temporal coding, Proc. Internet Work-Conf. on Artificial and Natural Neural Networks IWANN'97, Lecture Notes in Computer Science, vol. 1240, Springer, Berlin, 1997, pp. 265–272.

[47] B. Ruf, M. Schmitt, Hebbian learning in networks of spiking neurons using temporal coding, Proc. Internet Work-Conf. on Artificial and Natural Neural Networks IWANN'97, Lecture Notes in Computer Science, vol. 1240, Springer, Berlin, 1997, pp. 380–389.

[48] B. Ruf, M. Schmitt, Self-organizing maps of spiking neurons using temporal coding, in: J. Bower (Ed.), Computational Neuroscience: Trends in Research 1998, Plenum Press, New York, 1998, 509–514.

[49] R.R. de Ruyter van Steveninck, G.D. Lewen, S.P. Strong, R. Koberle, W. Bialek, Reproducibility and Variability in Neural Spike Trains, Science 275 (1997) 1805–1808.

[50] K. Sen, J.C. Jorge-Rivera, E. Marder, L.F. Abbott, Decoding synapses, J. Neurosci. 16(19) (1996) 6307–6318.

[51] L. Shastri, V. Ajjanagadde, From simple associations to systematic reasoning: a connectionist representation of rules, variables and dynamic bindings using temporal synchrony, Behav. Brain Sci. 16 (1993) 417–494.

[52] G.M. Shepherd (Ed.), The Synaptic Organization of the Brain, 3rd Edition, Oxford University Press, New York, 1990.

[53] E.D. Sontag, Shattering all sets of k points in 'general position' requires $(k-1)/2$ parameters, Neural Comput. 9 (1997) 337–348.

[54] S. Thorpe, D. Fize, C. Marlot, Speed of processing in the human visual system, Nature 381 (1996) 520–522.

[55] S. Thorpe, J. Gautrais, Rapid visual processing using spike asynchrony, in: C.M. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, MIT Press, Cambridge, MA, 1997, pp. 901–907.

[56] M.V. Tsodyks, H. Markram, The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability, Proc. Natl. Acad. Sci. USA 94 (1997) 719–723.

[57] H.C. Tuckwell, Introduction to Theoretical Neurobiology, vols. 1 and 2, Cambridge University Press, Cambridge, 1988.

[58] E. Vaadia, I. Haalman, M. Abeles, H. Bergman, Y. Prut, H. Slovin, A. Aertsen, Dynamics of neuronal interactions in monkey cortex in relation to behavioural events, Nature 373 (1995) 515–518.

[59] L.G. Valiant, Circuits of the Mind, Oxford University Press, Oxford, 1994.

[60] C. von der Malsburg, The correlation theory of brain function, Internal Report 81-2 of the Department of Neurobiology of the Max Planck Institute for Biophysical Chemistry in Göttingen, Germany, 1981.

[61] M. Wehr, G. Laurent, Odour encoding by temporal sequences of firing in oscillating neural assemblies, Nature 384 (1996) 162–166.

[62] A.M. Zador, B.A. Pearlmutter, VC dimension of an integrate-and-fire neuron model, Neural Comput. 8(3) (1996) 611–624.

[63] M.L. Zaghloul, J.L. Meador, R.W. Newcomb (Eds.), Silicon Implementations of Pulse Coded Neural Networks, Kluwer Academic Publishers, Dordrecht, 1994.

[64] R. Zucker, Short-term synaptic plasticity, Annu. Rev. Neurosci. 12 (1989) 13–31.