On the Computational Power of Circuits of Spiking Neurons

Wolfgang Maass [a,*], Henry Markram [b]

[a] *Institute for Theoretical Computer Science, Technische Universitaet Graz, A-8010 Graz, Austria*

[b] *Brain Mind Institute, Ecole Polytechnique Federale de Lausanne, CH-1015 Lausanne, Switzerland*

Abstract

Complex real-time computations on multi-modal time-varying input streams are carried out by generic cortical microcircuits. Obstacles for the development of adequate theoretical models that could explain the seemingly universal power of cortical microcircuits for real-time computing are the complexity and diversity of their computational units (neurons and synapses), as well as the traditional emphasis on offline computing in almost all theoretical approaches towards neural computation. In this article we initiate a rigorous mathematical analysis of the real-time computing capabilities of a new generation of models for neural computation, liquid state machines, that can be implemented with – in fact benefit from – diverse computational units. Hence realistic models for cortical microcircuits represent special instances of such liquid state machines, without any need to simplify or homogenize their diverse computational units. We present proofs of two theorems about the potential computational power of such models for real-time computing, both on analog input streams and for spike trains as inputs.

1 Introduction

We propose a new conceptual and theoretical framework for the analysis of the power of cortical microcircuits for real-time computing. It has frequently been conjectured, and there also exists empirical evidence to support this

[*] Corresponding author.
Email addresses: maass@igi.tugraz.at (Wolfgang Maass), henry.markram@epfl.ch (Henry Markram).
URL: http://www.igi.tugraz.at/maass (Wolfgang Maass).

Preprint submitted to Elsevier Science *13 May 2004*

conjecture [1], that these neural microcircuits are not built individually for each computational task arising in the cortex, but rather that the same basic circuit architecture is used for diverse computational tasks in different brain areas. This motivates the search for general principles that could endow such generic recurrent neural circuits with seemingly "universal" computational power, especially for real-time computing.

In computer science and mathematical logic there exist well-established theoretical frameworks for making such universality-conjectures precise. One such framework characterizes the class of all digital functions that are in principle computable (this is the class of recursive functions) and shows that Turing machines have universal computational power with regard to this class. But this framework is geared towards offline computations on digital inputs, and is of little use for analyzing parallel real-time computations on analog functions of time (or on spike trains) by neural circuits. The same holds for the classical framework of computational complexity theory, where the class of recursive functions is replaced by the class P of all digital functions that can be computed within an acceptable (=polynomial) computation time.

We would like to argue that in contrast to Turing machines, generic computations by neural circuits are not digital, and are not carried out on static inputs, but rather on functions of time (or time series). A circuit that gets as input an analog function of time $u(\cdot)$ and outputs another function of time $y(\cdot)$ defines a map F between functions of time. Such map is called an operator in mathematics, and a filter F in engineering. In lack of a better term we will use the term filter in this article, although filters are usually viewed in neuroscience as somewhat trivial signal processing or preprossessing devices, and are sometimes associated with specific approaches towards neural computation that are currently not fashionable. However, one should not fall into the trap of identifying the general term of a filter with special classes of filters such as linear filters, quadratic filters, or more generally filters that can be represented by a *finite* Volterra or Wiener series (see [2], [3], [4], [5]), that all have rather limited computational capabilities. Rather one should keep in mind that any input to any organism is a function of time, and any motor output of an organism is a function of time. Hence the computation that they perform is a special case of a filter. The same holds true for any artificial behaving system, such as a robot.

Usually when one discusses filters, one automatically has a specific complexity hierarchy for filters in mind: linear filters are at the lowest level of this hierarchy, the next level of the hierarchy consists of filters that can be represented as a sum of a linear filter and a quadratic filter (i.e., a Volterra polynomial of degree 2), the nth level of the hierarchy consists of all filters that can be represented as Volterra polynomials (or equivalently as Wiener polynomials) of degree n, and the ∞-level of this hierarchy consists of all filters that can only be represented by infinite Volterra series. However even very basic computational tasks, such as for example the computational operation of a threshold gate or a perceptron, see [4], require filters that are located at a fairly high finite level, or even at the ∞-level of this hierarchy. Also the computational operation executed by a single neuron places any realistic computational model for even very simple neural circuits at the ∞-level of this hierarchy.

The approach pursued in this article suggests completely different representations and hierarchies for filters, which appear to be more useful for analyzing neural computation. Basic filters that are of particular importance in neural computation, such as the response of a thresholding device (for example a spiking neuron) to an incoming spike train, can appear at a low level of such alternative hierarchy. In fact, this alternative framework allows us to place any filters, for example those filters that happen to be abundantly available in some specific physical realization domain, at the bottom level of a hierarchy, and to measure the complexity of any target filter F in terms of how many such basis filters are needed to approximate F.

We had introduced in [6] a quite general computational model for filters, that conceptually divides real-time processing into two complementary subtasks:

(1) separation of different input functions by a filter L that is composed of finitely many basis filters B, drawn from some fixed pool B of filters
(2) static transformation of the current output of L into the current target output of the system by some fixed readout map f, chosen from some fixed pool F of "memory less" functions[1].

The resulting computational model $M = \langle L, f \rangle$ is called a *liquid state machine*. In contrast to the familiar *finite state machine* from computer science, which provides the computational core of any Turing machine, it allows a potentially infinite set of different output values $x^M(t)$ of the filter L (that may correspond for example in a biological interpretation to the current activation state of a recurrent neural circuit, or more abstractly to the current internal state of a high dimensional dynamical system). This state $x^M(t)$ is *liquid* in the sense that it will in general change continuously during the presentation of an input function $u(\cdot)$, not just at prespecified discrete time points. It is left up to the readout map f at which resolution this liquid state $x^M(t)$ is "read" and transformed into the output $(Mu)(t)$ of the liquid state machine M at time t.

Thus formally each liquid state machine M computes a filter that maps input functions $u(\cdot)$ onto output functions $(Mu)(\cdot)$.[2] In accordance with common

[1] These functions are called "memoryless" or "static" because they map discrete or analog numbers on numbers, rather than functions of time on numbers or on other functions of time.

[2] We often write $u(\cdot)$ instead of u for the input functions in order to remind the

conventions we write $(Fu)(t)$ for the output of a filter F when F is applied to the input function $u(\cdot)$. In this article all these functions will be interpreted as functions of time. It is important to keep in mind that this filter output $(Fu)(t)$ at time t will in general not just depend on the value of the input function $u(\cdot)$ at time t, but potentially on all values $u(s)$ for $s \leq t$.[3] Using this terminology one can formally describe the two complementary computational operations of a liquid state machine (LSM) $M = \langle L^M, f^M \rangle$ by the equation

$$x^M(t) = (L^M u)(t), \tag{1}$$

i.e., all information about past values $u(s)$, $s \leq t$, of the input function $u(\cdot)$ that might be needed for the output $(Mu)(t)$ of the LSM M at time t is first condensed into the current liquid state $x^M(t)$, and a second equation

$$(Mu)(t) = f^M(x^M(t)), \tag{2}$$

which says that the output of M at time t is produced from $x^M(t)$ by applying its static readout map f^M to the current liquid state. In the application of this framework to computations in recurrent neural circuits the first equation describes (primarily) the task of temporal integration, and the second equation describes the familiar (and less) difficult task of spatial integration of information.

A closely related computational model has independently been proposed in [7] under the name of “echo state network”. This model is formulated for discrete time and iterative updates of network states, rather than directly for filters. We will show in section 2 that the approximation theorems that are proven in this article can also be applied to such networks.

We will prove in this article that, under some mild conditions on the pool B of basis filters and the class F of possible readout functions, the class of LSMs M that are composed from basis filters in B and readout functions in F have *universal computational power* with regard to a very interesting class of computational operations on analog functions of time that we will now define.

reader that these inputs are functions (of time), rather than numbers. These input functions range over some domain U, or the n-fold cross product U^n of such domain U. In the latter case $u(\cdot) \in U^n$ represents a multi-dimensional input, consisting of n functions of time that simultaneously enter the system. The output of all filters that are considered in this article could also be multi-dimensional. However, a filter with m-dimensional output is formally equivalent to m filters with one-dimensional output. Hence we address in our notation only the case where the output of a filter is some function $y(\cdot) \in \mathbb{R}^{\mathbb{R}}$, i.e., a function that maps real numbers (usually interpreted as time) into real numbers.

[3] We restrict our attention in this article on *causal* filters F, where $(Fu)(t)$ does not depend on $u(s)$ for $s > t$.

We will only consider computational operations on functions of time that are input-driven, in the sense that the output does not depend on any absolute internal clock of the computational device. Filters that have this property are called time invariant. Formally one says that a filter F is *time invariant* if any temporal shift of the input function $u(\cdot)$ by some amount t_0 causes a temporal shift of the output function by the same amount t_0, i.e., $(Fu^{t_0})(t) = (Fu)(t+t_0)$ for all $t, t_0 \in \mathbb{R}$, where $u^{t_0}(t) := u(t + t_0)$. Note that if the domain U of input functions $u(\cdot)$ is closed under temporal shifts, then a time invariant filter $F : U^n \to \mathbb{R}^{\mathbb{R}}$ is identified uniquely by the values $y(0) = (Fu)(0)$ of its output functions $y(\cdot)$ at time 0. In other words: in order to identify or characterize a time invariant filter F we just have to observe its output values at time 0, while its input varies over all functions $u(\cdot) \in U^n$. Hence one can replace in the mathematical analysis such filter F by a functional, i.e. a simpler mathematical object that maps input functions on real values (rather than on functions of time).

Various theoretical models for analog computing are of little practical use because they rely on hair-trigger decisions, for example they allow that the output is 1 if the value of some real-valued variable x is ≥ 0, and 0 otherwise. Another unrealistic aspect of some models for computation on functions of time $u(\cdot)$ is that they allow that the output of the computation depends on the full infinitely long history of the input function $u(\cdot)$. On the other hand it was shown in [8] that recurrent analog neural networks automatically acquire a fading memory quality as soon as any realistic type of noise is assumed for their analog processing units. One may argue that this result destroys all hopes that the amazing computational capabilities that have been predicted in some theoretical articles on recurrent neural networks can be implemented in any physical device, such as for example a biological neural system. Therefore, instead of trying to simulate Turing machines, we focus in this article on analog computations of continuous maps that can be expected to degrade gracefully under the influence of noise (this has been empirically supported by computer simulations reported in [6], where noise had been added to the membrane potential of the integrate-and-fire neurons in the circuit). More precisely, we restrict our attention to the computation of filters that have fading memory. One may argue that no biologically relevant computations are eliminated by this restriction. *Fading memory* is a continuity property of filters F, which requires that for any input function $u(\cdot) \in U^n$ the output $(Fu)(0)$ can be approximated by the outputs $(Fv)(0)$ for any other input functions $v(\cdot) \in U^n$ that approximate $u(\cdot)$ on a sufficiently long time interval $[-T, 0]$ going back into the past. Formally one defines that $F : U^n \to \mathbb{R}^{\mathbb{R}}$ has fading memory if for every $u \in U^n$ and every $\varepsilon > 0$ there exist $\delta > 0$ and $T > 0$ so that $|(Fv)(0) - (Fu)(0)| < \varepsilon$ for all $v \in U^n$ with $\|u(t) - v(t)\| < \delta$ for all $t \in [-T, 0]$. Informally, a filter F has fading memory if the most significant bits of its current output value $(Fu)(0)$ depend just on the most significant bits of the values of its input function $u(\cdot)$ in some finite time interval $[-T, 0]$

going back into the past. Thus, in order to compute the most significant bits of $(Fu)(0)$ it is not necessary to know the *precise* value of the input function $u(s)$ for any time s, and it is also not necessary to know *anything* about the values of $u(\cdot)$ for more than a finite time interval back into the past. Note that a filter F that has fading memory is automatically causal, i.e., $(Fu)(0)$ does not depend on values $u(s)$ for $s > 0$.

In section 2 we will consider a corresponding notion of fading memory for circuits, which begin their computation at a specific time point in a specific initial state, and show that all these circuits define fading memory filters. This creates a link between common concepts from circuit complexity theory and the filter-based notation used in this article. Basically one just has to observe that for computational devices with fading memory it does not matter whether they have been driven "forever" by some time-varying input $u : (-\infty, \infty) \to \mathbb{R}^n$ (as suggested by the filter notation), or whether the computation had been started at a specific time point t_I very far back in the past, in some unknown initial state x_I. The filter-oriented notation is mathematically more convenient, since one can eliminate the formal dependence on t_I and x_I, which practically is of little interest for fading memory devices. We will show in section 3 of this article that liquid state machines (LSMs) with basis filters from some class B and readout maps from some class F have under mild conditions on B and F universal computational power with regard to all time invariant fading memory computations on continuous functions of time, in the sense that any time invariant fading memory filter F can be approximated by an LSM M from this class up to any given degree of precision. In section 4 this result is extended to the mathematically more difficult case where the input functions $u(\cdot)$ represent spike trains, rather than continuous functions of time.

The condition on the class B of basis filters that is needed for these results is the following *pointwise separation property*: for any two input functions $u(\cdot), v(\cdot) \in U^n$ with $u(s) \neq v(s)$ for some $s \leq 0$ there exists some filter $B \in B$ that separates $u(\cdot)$ and $v(\cdot)$, i.e., $(Bu)(0) \neq (Bv)(0)$. Note that it is *not* required that there exists a filter $B \in B$ with $(Bu)(0) \neq (Bv)(0)$ for any two functions $u(\cdot), v(\cdot) \in U^n$ with $u(s) \neq v(s)$ for some $s \leq 0$. Simple examples for classes B of filters that have this pointwise separation property are the class of all delay filters $u(\cdot) \mapsto u^{t_0}(\cdot)$ (for $t_0 \in \mathbb{R}$) and the class of all linear filters with impulse responses of the form $h(t) = e^{-at}$ with $a > 0$. A biologically quite interesting class of filters that satisfies the formal requirement of the pointwise separation property is the class of filters defined by standard models for dynamic synapses, see [9]. A liquid filter L^M of an LSM M is said to be *composed* of filters from B if there are finitely many filters $B_1, \ldots, B_m$ in B – to which we refer as basis filters in this context – so that $(L^M u)(t) = \langle (B_1 u)(t), \ldots, (B_m u)(t) \rangle$ for all $t \in \mathbb{R}$ and all input functions $u(\cdot)$ in U^n. Thus for a biological interpretation an array of parallel input synapses with somewhat different synaptic parameters (that control their internal dynamics) would formally suffice to implement all desired liquid filters L^M. However this argument ignores the fact that in reality the basis filters B not just have to make $(Bu)(0) \neq (Bv)(0)$ – as required by the pointwise separation property – but they have to make sure that for all pairs of input functions $u(\cdot), v(\cdot)$ for which the target output of the system has to differ at time 0 there exists some component of the input to the readout module that assumes at time 0 sufficiently different values for circuit inputs $u(\cdot)$ and $v(\cdot)$, so that this difference can be picked up by the readout in spite of noise. Obviously the complex recurrent architecture of cortical microcircuits ("loops within loops") and the diverse temporal responses of different types of neurons and synapses (see for example [10] and [11]) contribute a rich repertoire of amplifiers for input differences.

The condition on the class F of readout functions that is needed for our results is the following *approximation property*, sometimes referred to as universal approximation property in neural network theory: for any $m \in N$, any compact (i.e., closed and bounded) set $X \subseteq \mathbb{R}^m$, any continuous function $h : X \to \mathbb{R}$, and any given $\rho > 0$, there exists some f in F so that $|h(x) - f(x)| \leq \rho$ for all $x \in X$.

We will show in Theorem 3.1 of section 3 that those two conditions (pointwise separation property of B and approximation property of F) together guarantee universal computational power for the corresponding class of LSMs, i.e. the power to approximate any time invariant fading memory filter F on continuous functions of time with any desired precision. In section 4 we define a suitable notion of fading memory computations on spike trains, that allows us to prove in Theorem 4.1 a corresponding result for LSMs that carry out computations on spike trains.

The theoretical approach towards computation on spike trains that is proposed in this article differs strongly from previous approaches that have focused on the construction of specific circuits of spiking neurons that can carry out specific computations on spike trains (see for example [12]). A generic problem in these approaches is the need to construct special mechanisms for absorbing and integrating information encoded in the interspike intervals of biologically realistic input spike trains, whose interspike intervals are typically much larger (in the range of 10 to 100 ms) than the transmission delays between neurons in a cortical microcircuit (that tend to be below 5 ms). Furthermore these explicit constructions of circuits of spiking neurons tend to produce circuits with a dominant feedforward structure, in spite of the fact that biological neural circuits are highly recurrent, simply because it is very hard (if not impossible) to construct a recurrent circuit of spiking neurons that has a given input/output behavior.

On the other hand it had already been shown in [13] that generic recurrent circuits of integrate-and-fire neurons are able to transform temporal input patterns into spatial activity patterns of the circuit. Hence one may argue that it is not necessary to understand, or even control, the full input/output behavior of such circuits. Instead, it suffices to verify that such recurrent circuits have the *pointwise separation property* (practically it even suffices to verify this pointwise separation property for input function $u(\cdot)$ and $v(\cdot)$ for which the target output of the filter should be different). In fact, extensive computer simulations of rather realistic models for biological neural circuits [6] suggest that a randomly drawn recurrent circuit consisting of a few hundred neurons, can separate a very large class of different input spike trains through its state of activation at some later time point t. Hence one can restrict all constructive (or adaptive) effort for approximating a *given* filter F by a recurrent circuit of spiking neurons to the selection or learning of an appropriate readout map f. It is known (see [14] that already a single pool of spiking neurons (with just *feedforward* connections into this pool, and out of this pool) can provide such f: it can approximate any given continuous function, and hence has the desired approximation property. Furthermore there exists a simple local learning rule that can adapt such pool of spiking neurons to approximate a specific given continuous function (see [14], [6]). Alternatively one could approximate by spiking neurons any other class of feedforward neural networks that is known to have the universal approximation property, such as multi-layer perceptrons (see [15] for an approximation of these artificial neural networks by networks of spiking neurons). Thus the theoretical approach that is outlined in this paper suggests to approximate a given input/output behavior F on functions of time (or spike trains) by first picking randomly some sufficiently complex recurrent circuits of spiking neurons, and then adapting the weights of a pool of output neurons to approximate the given target outputs. This approach is biologically more realistic from the point of view of learning than learning algorithms that aim at modifying the state trajectory of the recurrent circuit, since the neurons that produce the actual circuit output have better access to error signals, and it is clearer in which direction a synaptic weight should be changed in order to reduce the output error, compared with neurons deep inside the recurrent circuit.

In addition this approach has the advantage that the same recurrent circuit can be used simultaneously – with the help of additional other readout functions that can be trained independently – to compute in parallel different outputs from the same input $u(\cdot)$. This provides a new paradigm for parallel computation in real-time on time-varying input that appears to be rather attractive from the biological point of view. In other words: m different filters $F_1, \ldots, F_m$ can be implemented with the same recurrent circuit (i.e., the same L), yielding a giant saving of hardware (i.e., neurons).

Altogether the computational approach pursued in this article suggests that for analyzing computations in cortical microcircuits it may be less fruitful trying to understand *how* (in which specific code) different sensory inputs are represented by the very high-dimensional vector that describes the current circuit activity, a question that is usually phrased as the question of neural coding or neural representation of external stimuli. The more important question from the point of view of the neural system (more precisely: from the point of view of the readout modules of the neural system) is how well saliently different inputs are separated through the high-dimensional vectors describing the current neural activity. In this way one arrives on a completely different road at the concept of "intelligence without representation", that has previously been proposed in the context of robotics [16] in order to overcome known deficiencies of traditional approaches from artificial intelligence in coping with the need to deliver adequate output-behaviours in real-time for realistically complex sensory input streams.

2 Recurrent Circuits with Fading Memory Define Filters with Fading Memory

There exists a difference between the type of functions computed by circuits and the type of functions represented by filters. A filter F assigns an output value $(Fu)(t)$ to a time point $t \in \mathbb{R}$ and an input function $u : \mathbb{R} \to \mathbb{R}$ (or $u : \mathbb{R} \to \mathbb{R}^n$), assuming implicitly that input has been coming in "forever". In contrast, computations in circuits are usually assumed to start at some concrete time point $t_I < t$ back in the past, in some concrete initial state x_I, thereby defining a circuit output $C(x_I, u\lceil_{[t_I,t]}, t) \in \mathbb{R}$ at the current time t that depends just on a finite segment $u\lceil_{[t_I,t]}$ of the input function $u(\cdot)$, but in addition on the initial state x_I of the circuit. However many recurrent circuits have fading memory, which implies that the current circuit output $C(x_I, u\lceil_{[t_I,t]}, t)$ depends less and less on its initial state x_I when $t - t_I$ grows. We prove that any such circuit C with fading memory defines a fading memory filter F_C through the definition $(F_C u)(t) := \lim_{t_I \to -\infty} C(x_I, u\lceil_{[t_I,t]}, t)$, where arbitrary initial values x_I may be used on the r.h.s. Therefore it is justified to model a circuit with fading memory whose activity had started a relatively long time ago (which is usually the case for a biological neural circuit) by a fading memory filter. Thus, as long as one restricts the attention to circuits and filters with fading memory, the structural difference between these two computational frameworks disappears (or rather: fades away). This result, which we formulate precisely as Theorem 2.1, creates in particular a bridge between the filter-oriented mathematical analysis pursued in this article, and the circuit-oriented mathematical analysis in [7] of closely related effects for artificial neural network models.

Definition 2.1 *Fix some space U of input functions $u : \mathbb{R} \to \mathbb{R}$, and let C be a circuit that assigns to any initial state x_I from some set I, any $t \in \mathbb{R}$, and any finite segment $u\lceil_{[t_I,t]}$ of a function (or vector of functions) $u(\cdot) \in U^n$ some output $C(x_I, u\lceil_{[t_I,t]}, t) \in \mathbb{R}$. We say that this circuit C has fading memory (relative to U and I) if there exists for every $t \in \mathbb{R}$ and every $\varepsilon > 0$ some $\delta > 0$ and $t_I < t$ so that $|C(x_I, u\lceil_{[t_I,t]}, t) - C(x_I^{'}, v\lceil_{[t_I,t]}, t)| < \varepsilon$ for all $u, v \in U^n$ and for all $x_I, x_I^{'} \in I$, provided that $\|u(t') - v(t')\| \leq \delta$ for all $t' \in [t_I, t]$.*

Note: We assume that any circuit C has associated with it a state transition function S_C so that $S_C(x_I, u\lceil_{[t_I,t]}, t) \in I$ (which represents the internal state of circuit C at time t) has the property that $C(x_I, u\lceil_{[t_I,t]}, t) = C(S_C(x_I, u\lceil_{[t_I,t']}, t'), u\lceil_{[t',t]}, t)$ for any $t' \in [t_I, t]$.

Theorem 2.1 *Any circuit C that has fading memory (relative to U and I) defines a fading memory filter $F_C : U^n \to \mathbb{R}^{\mathbb{R}}$ through the definition*

$$(F_C u)(t) := \lim_{t_I \to -\infty} C(x_I, u\lceil_{[t_I,t]}, t),$$

where arbitrary initial states $x_I \in I$ may be used on the r.h.s.

Remark 2.2 All the concepts and results of this paper can also be formulated for discrete instead of continuous time. For discrete time it was shown in [7] that any network which is uniformly state contracting in the sense of Definition 4 in [7] defines a circuit that has fading memory according to our preceding definition, and hence according to the preceding Theorem 2.1 a fading memory filter. Thus all the sigmoidal networks with weight matrix $\|W\| < 1$ that are considered in [7] provide interesting special cases of fading memory filters.

Proof of Theorem 2.1: We first show that $\lim_{t_I \to -\infty} C(x_I, u\lceil_{[t_I,t]}, t)$ exists, for any choices of states $x_I \in I$ for the initial time points $t_I < t$. Fix some $\varepsilon > 0$. We show that there exists some $t_I < t$ so that $|C(\hat{x}_I, u\lceil_{[t_1,t]}, t) - C(\tilde{x}_I, u\lceil_{[t_2,t]}, t)| < \varepsilon$ for any $\hat{x}_I, \tilde{x}_I \in I$, any $u \in U^n$, and any $t_1, t_2 \leq t_I$. Since C is assumed to have fading memory, there exists some $t_I < t$ so that $|C(x_I, u\lceil_{[t_I,t]}, t) - C(x_I', u\lceil_{[t_I,t]}, t)| < \varepsilon$ for all $x_I, x_I' \in I$ and all $u \in U^n$. Set $x_I := S_C(\hat{x}_I, u\lceil_{[t_1,t_I]}, t_I)$ and $x_I' := S_C(\tilde{x}_I, u\lceil_{[t_2,t_I]}, t_I)$. Then $|C(\hat{x}_I, u\lceil_{[t_1,t]}, t) - C(\tilde{x}_I, u\lceil_{[t_2,t]}, t)| < \varepsilon$ according to the basic property of the state transition function S. Thus we have shown that the filter F_C is well-defined by the definition in the claim of Theorem 2.1. In fact we have shown that the convergence to $(F_C u)(t)$ is uniform in u, which we will need for the second part of this proof.

It remains to be shown that this filter F_C has fading memory. Fix some $u \in U^n$ and $\varepsilon > 0$. We have to prove that there exist $\delta > 0$ and $T > 0$ so that $|(F_C v)(0) - (F_C u)(0)| < \varepsilon$ for all $v \in U^n$ with $\|u(t) - v(t)\| < \delta$ for all $t \in [-T, 0]$. Fix $\Delta > 0$ so that $|(F_C \tilde{u})(0) - C(\tilde{x}_I, \tilde{u}\lceil_{[t_I,0]}, 0)| < \frac{\varepsilon}{3}$ for all $\tilde{x}_I \in I$, all

$\tilde{u} \in U^n$, and all $t_I \leq -\Delta$ (we exploit here that the circuit outputs converge to $(F_C\tilde{u})(0)$ uniformly in $\tilde{u}$ for $t_I \to -\infty$, which follows from the first part of the proof). Furthermore fix $\delta > 0$ and $\Delta' > 0$ according to the definition of fading memory for circuits so that $|C(x_I, u\lceil_{[-\Delta',0]}, 0) - C(x_I', v\lceil_{[-\Delta',0]}, 0)| < \frac{\varepsilon}{3}$ for all $u, v \in U^n$ and $x_I, x_I' \in I$, provided that $\|u(t') - v(t')\| \leq \delta$ for all $t' \in [-\Delta', 0]$. Since $x_I, x_I' \in I$ are allowed to be arbitrarily chosen, the same property holds with the same $\delta > 0$ for any $t_I \leq -\Delta'$ instead of $-\Delta'$ (use the state transition function S_C). Set $t_I := -\max(\Delta, \Delta')$. Then if $\|u(t) - v(t)\| \leq \delta$ for all $t \in [-T, 0]$, we have for any $x_I, x_I' \in I$ that $C(x_I, u\lceil_{[t_I,0]}, 0) - C(x_I', v\lceil_{[t_I,0]}, 0)| < \frac{\varepsilon}{3}$, $|(F_Cu)(0) - C(x_I, u\lceil_{[t_I,0]}, 0)| < \frac{\varepsilon}{3}$, and $|(F_Cv)(0) - C(x_I, v\lceil_{[t_I,0]}, 0)| < \frac{\varepsilon}{3}$, hence $|(F_Cu)(0) - (F_Cv)(0)| < \varepsilon$. □

Remark 2.3 One usually assumes that a circuit C has no absolute dependence on time, i.e., that the internal state and current output of C just depend on the time that has passed since the circuit C had been activated (but of course also on the initial state x_I and the input that has entered the circuit in the meantime). Such circuit C is time invariant in the sense that $C(x_I, u^{t_0}\lceil_{[t_I,t]}, t) = C(x_I, u\lceil_{[t_I+t_0,t+t_0]}, t + t_0)$ for all functions $u^{t_0} \in U^n$ defined by $u^{t_0}(t) = u(t + t_0)$. If a circuit C with fading memory is time invariant in this sense, then the filter F_C that is defined by C has not only fading memory, but is in addition time invariant.

3 Real-Time Computing with Analog Input

In this section we discuss the mathematically simpler case of computations on continuous input functions, such as for example the postsynaptic currents (or contributions to the membrane potential) in neurons of a neural circuit that result from spiking activity in afferent neurons, i.e., neurons that are not components of the considered circuit. These inputs can be modeled by a vector of continuous functions of time $u : \mathbb{R} \to \mathbb{R}$. In this interpretation it is justified to assume that the values of these functions are uniformly bounded, and also that their steepness (= absolute value of the derivative in case that they are differentiable) is uniformly bounded. Both assumptions will be needed for the subsequent theorem. More precisely, we assume that the domain U of input functions has the form

$$U = \{u : \mathbb{R} \to [-K, K] : |u(s) - u(t)| \leq K' \cdot |s - t| \quad \text{for all} \quad s, t \in \mathbb{R}\}$$

for some arbitrary constants $K, K' > 0$. The filters in the subsequent theorem are applied to vectors $\underline{u} = \langle u_1, \ldots, u_n \rangle$ of functions from U for some arbitrary fixed $n \in N$.

In order to be able to approximate any given time invariant fading memory filter F by liquid state machines M with liquid filters L^M composed from basis filters $B_1, \ldots, B_m$ from some fixed class B of basis filters, it is obviously necessary that B has the pointwise separation property, because for any $\underline{u}, \underline{v} \in U^n$ with $\underline{u}(s) \neq \underline{v}(s)$ for some $s \leq 0$ there exists a time invariant fading memory filter F with $(F\underline{u})(0) \neq (F\underline{v})(0)$. Obviously this filter F can be approximated arbitrarily closely by liquid state machines with basis filters from B only if there exists some basis filter $B \in B$ with $(B\underline{u})(0) \neq (B\underline{v})(0)$. The following theorem shows that this necessary condition is essentially also sufficient.

Theorem 3.1 *Assume that B is an arbitrary class of time invariant fading memory filters that has the pointwise separation property. Furthermore assume that F is an arbitrary class of functions that has the approximation property.*

Then any given time invariant fading memory filter F can be approximated arbitrarily closely by LSMs M with liquid filter L^M composed from basis filters in B and readout maps f^M chosen from F. In formal terminology: For every $\varepsilon > 0$ there exist $m \in \mathbb{N}$, $B_1, \ldots, B_m \in B$ and $f^M \in F$ so that the LSM $M = \langle L^M, f^M \rangle$ with L^M composed of $B_1, \ldots, B_m$ satisfies $|(F\underline{u})(t) - (M\underline{u})(t)| < \varepsilon$ for all $\underline{u} \in U^n$ and all $t \in \mathbb{R}$.

Furthermore, if all functions in F are continuous, then a given filter F can be approximated arbitrarily closely by such LSMs if and only if *F is time invariant and has fading memory.*

Remark 3.2 A remarkable consequence of this theorem is that for a large variety of classes B of basis filters (such as delay lines, linear filters, dynamic synapses, or circuits with fading memory) the pointwise separation property, in combination with sufficiently "flexible" readout maps, endows the resulting LSMs with "universal computational power" in the giant class of filters F that are time invariant and have fading memory. In fact, one may argue that any neural computation that may be vital for the survival of an organism can be represented by such time invariant fading memory filter F.

Proof of Theorem 3.1: The last part of the theorem ("if and only if") follows immediately from the first part, because any LSM $M = \langle L^M, f^M \rangle$ with L^M composed from time invariant fading memory filters (from B) and a continuous function f^M represents a time invariant fading memory filter. Furthermore time invariance and fading memory are properties that are inherited by any filter F that can be approximated arbitrarily closely by time invariant fading memory filters.

The first part of the theorem follows from the Stone-Weierstrass Approximation Theorem, similarly as in [17], see also [18]. For simplicity of notation we just consider the case $n = 1$ (the case $n > 1$ is analogous). We apply the Stone-Weierstrass Theorem (see for example [19]) to functionals that map

$$U^- := \{u\lceil_{(-\infty,0]} : u \in U\}$$

into $\mathbb{R}$ (where $u\lceil_{(-\infty,0]}$ is the restriction of the function $u : \mathbb{R} \to \mathbb{R}$ to the domain $(-\infty, 0]$; the set U was defined before Theorem 3.1. The transition to functionals is necessary because the Stone-Weierstrass Theorem can only be applied to functions with values in $\mathbb{R}$, hence not directly to filters (whose values are from $\mathbb{R}^{\mathbb{R}}$, rather than $\mathbb{R}$). Note that a fading memory filter F is automatically causal, hence the value of $(Fu)(0)$ depends only on $u\lceil_{(-\infty,0]}$. Furthermore, since U is closed under shifts in time, any time invariant filter $F : U \to \mathbb{R}^{\mathbb{R}}$ is already completely determined by its values $(Fu)(0)$ for $u \in U$. Hence we can define for any fading memory filter F the value $(Fu\lceil_{(-\infty,0]})(0)$ of F for input functions from U^- uniquely as the value $(Fv)(0)$ for any $v \in U$ with $v\lceil_{(-\infty,0]} = u\lceil_{(-\infty,0]}$. Furthermore, if a filter $F : U \to \mathbb{R}^{\mathbb{R}}$ is time invariant, then it is already uniquely determined by these values $(Fu\lceil_{(-\infty,0]})(0)$ for $u \in U$.

We will apply the following formulation of the Stone-Weierstrass Theorem from [19]: *Assume that E is a compact metric space, and S is a set of continuous functions from E into $\mathbb{R}$ that has the pointwise separation property (i.e., for any $u, v \in E$ with $u \neq v$ there exists some function $B \in S$ with $B(u) \neq B(v)$). Then there exist for any continuous function F from E into $\mathbb{R}$ and for any $\varepsilon > 0$ some functions $B_1, \ldots, B_m$ in S and a polynomial* [4] *p such that $|F(u) - p(B_1, \ldots, B_m)(u)| \leq \varepsilon$ for every $u \in E$.*

In order to apply this result for the proof of Theorem 3.1 we just have to show that U^- can be endowed with a metric d that turns U^- into a compact metric space. Furthermore this metric d needs to have the property that for any fading memory filter F on U the function from U^- into $\mathbb{R}$ defined by $(Fu\lceil_{(-\infty,0]})(0)$ is continuous with respect to this metric d on U^-. In addition we have to show that the polynomials p can be replaced by functions $f \in F$.

We define a function $d : U^- \times U^- \to \mathbb{R}$ by

$$d(u, v) : = \sup_{t \leq 0} \frac{|u(t) - v(t)|}{1 + |t|} \quad .$$

[4] Sums and products of functions $B_i, B_j \in S$ are pointwise defined, e. g. $(B_i \cdot B_j)(u) := (B_i(u)) \cdot (B_j(u))$. Once one has defined sums and products of functions in S, the definition of a polynomial $p(B_1, \ldots, B_m)$ of functions in S is obvious. Note that such polynomial may also have a constant term, with an arbitrary real value.

Lemma 3.3 *The function d defines a metric on U^- that turns U^- into a compact metric space.*

We refer to the Appendix for a *proof of Lemma 3.3*.

If F is any filter on U with fading memory, there exists for every given $u \in U$ and every given $\rho > 0$ some $\delta > 0$ and $T > 0$ so that $|(Fu)(0) - (Fv)(0)| < \rho$ for all $v \in U$ with the property that $|u(t) - v(t)| < \delta$ for all $t \in [-T, 0]$. Set $\delta' := \frac{\delta}{1+T}$. Then it is obvious that for any $v \in U^-$ the assumption $d(u\lceil_{(-\infty,0]}, v\lceil_{(-\infty,0]}) < \delta'$ implies that $\frac{|u(t)-v(t)|}{1+|t|} < \delta'$, and hence $|u(t) - v(t)| < \delta$, for all $t \in [-T, 0]$. Thus $|(Fu\lceil_{(-\infty,0]})(0) - (Fv\lceil_{(-\infty,0]})(0)| < \rho$ for any $u\lceil_{(-\infty,0]}, v\lceil_{(-\infty,0]}$ in U^- with $d(u\lceil_{(-\infty,0]}, v\lceil_{(-\infty,0]}) < \delta'$. Hence the function from U^- into $\mathbb{R}$ defined by $(Fu\lceil_{(-\infty,0]})(0)$ is continuous with respect to the metric d on U^-.

To complete the proof of Theorem 3.1 it just remains to show that the polynomial p, which occurs in the statement of the Stone-Weierstrass Theorem, can be replaced in our context by a function $f \in F$. Since all basis filters $B \in B$ are assumed to have fading memory, they yield continuous functions from the metric space $\langle U^-, d\rangle$ into $\mathbb{R}$. Since $\langle U^-, d\rangle$ is compact according to Lemma 3.3, any continuous function with domain $\langle U^-, d\rangle$ has a bounded range. Hence it suffices to approximate the given polynomial $p(B_1, \ldots, B_m)$ on a compact subset S of $\mathbb{R}^m$. The approximation property of the class F implies that there exists for every $\rho > 0$ some $f \in F$ so that $|p(\underline{x}) - f(\underline{x})| < \rho$ for all $\underline{x} \in S$. This completes the proof of Theorem 3.1. $\square$

4 Real-Time Computing on Spike Trains

Previous theoretical work on computations with spiking neurons has usually focused on computing with single spikes (see [20] for a survey). In this section we address the question which computations on *spike trains* can in principle be carried out by neural circuits.

We show in Theorem 4.1 that *if* there exists for any two different spike trains u and v a neural circuit $B_{u,v}$ that can separate u and v (through its activity at time 0, after the spike train was given as input to the circuit), *then* an extremely large set of computations on spike trains can be carried out by networks of spiking neurons.

The existence of such circuits $B_{u,v}$ has not yet been rigorously proven. But computer simulations (see [6,21]) show that randomly connected circuits of spiking neurons with a sufficient number of recurrent synaptic connections will typically assume different states at time 0 in response to spike input that differed significantly during the preceding 1000 ms. In fact, if the connection density is very high, some networks of spiking neurons tend to be chaotic, i.e., their current state amplifies and preserves even small differences between earlier circuit states for extended periods of time [22]. Some closely related theoretical results for recurrent threshold circuits with online input that characterize parameter regimes where such circuits have good representation properties were recently presented in [23]. Thus altogether one might argue that the separation property of neural circuits for spike input, which is one of the assumptions of the subsequent Theorem 4.1, appears to be a reasonable assumption from a pragmatic point of view (for inputs u, v that differed not too long back in the past, say during the last few seconds), but that it is still an open problem from a theoretical point of view.

The other assumption of Theorem 4.1 is the existence of a class F of possible readout function that has the approximation property. This assumption can in principle be satisfied by pools of spiking neurons since pools of spiking neurons can approximate any given continuous function on static inputs (see [24] and [14] for a proof).

In order to model the computation of a neural circuit on spike input by a fading memory filter, as we do in this section, one also needs to assume that the circuit output depends continuously on the temporal structure of the spike input to the circuit, i.e. moving the arrival time of a spike by an infinitesimal amount changes the circuit output just by an infinitesimal amount. This assumption is approximately satisfied if one takes into account that a biological circuit of neurons is really a stochastic system, and in order to arrive at a formally deterministic output one needs to average: either over space, i.e., over the outputs of several basically identical circuits, or over time, i.e. over several trials with the same input. Obviously only the averaging over space can be carried out by a neural system in real-time. Furthermore we view the effects of the output neurons of the circuit on the membrane potential of postsynaptic neurons as the output of the filter, rather than the output spikes themselves, in order to approximate the computational operation of a neural circuit with spike input by a fading memory filter with spike input.

The theoretical result about the computational power of networks of spiking neurons that is presented in this section has consequences for practical computations with spiking neurons in software- and hardware simulations. In fact, computer simulations suggest that the theoretical predictions about effects that occur “in the limit” become already clearly visible for rather small networks, even in the presence of noise. Although our theoretical result leaves open the question how large a network of spiking neurons needs to be in order to carry out a given computation on spike trains, and also leaves open the question how that computation would be affected by noise, the computer simulations of [25,6,21] suggest that a large variety of complex computational operations on spike trains can be carried out by networks consisting of a few hundred neurons, even with noise added to the membrane potential of these neurons. Actually, to the best of our knowledge the paradigm for computations on spike trains that is suggested by the subsequent Theorem 4.1 provides at present the only strategy for implementing a given complex computation on spike trains by a circuit of spiking neurons.

The analysis of the computational power of LSMs on continuous input functions from section 3 cannot be applied directly to the case where the input u consists of spike trains, i.e., of sequences of point events in time. As mathematical object one can view a spike train u simply as a subset of $\mathbb{R}$: the set of time points where a spike occurs in the spike train, or equivalently (up to some fixed delay): the set of firing times of the neuron that emits this spike train). Alternatively one could represent each spike train u (defined as a subset of $\mathbb{R}$) as a function $f_u : \mathbb{R} \rightarrow \{0, 1\}$ with $f_u(t) = 1 \Leftrightarrow t \in u$. Due to the firing mechanism of biological neurons there exists some minimal distance $\Delta > 0$ (say, $\Delta = 0.1$ms) between any two firing times of the same neuron. Hence it is safe to assume that the sets $u \subseteq \mathbb{R}$ that represent spike trains are not arbitrarily dense, but that $|s - t| \geq \Delta$ for any two different points $s, t \in u$. This rather trivial condition will be essential for the subsequent mathematical analysis. We denote the class of all sets $u \subseteq \mathbb{R}$ with this property by U_Δ, where $\Delta > 0$ is some arbitrary fixed parameter (say: $\Delta = 0.1$ms).

From the point of view of mathematics there exist some pretty weird functions F from spike trains into real numbers, that cannot be expected to be computable by any realistic network of spiking neurons under realistic noisy conditions. Examples are functions F that output 1 if the last interspike interval was $\geq \pi$, and 0 otherwise, or functions F that output 1 if the infinite sequence of spikes in the preceding spike train encodes – one bit per spike – an infinite bit sequence that corresponds to an irrational number, and that output 0 if this infinite bit sequence encodes a rational number. Hence before one can formulate a practically meaningful result about the capability of LSMs to approximate any given “relevant” map F from spike trains into real numbers, one first has to identify a suitable class of such maps F that may serve as “computational universe” for this purpose. Obviously the output of F needs to depend in a continuous way on spike times in the input, since otherwise no physical device can implement it. For the same reason the amount of information that F needs to “remember” from past interspike intervals of the input spike train should be finite. We will show that a variation of the fading memory notion, that was discussed in the preceding section, can be used to define a suitable universe for analyzing realistic computations on spike trains.

We define a suitable notion of fading memory for filters $F : U_\Delta^n \to \mathbb{R}^\mathbb{R}$ that map arrays of n spike trains from U_Δ into arbitrary functions of time. The corresponding definition from section 1 for continuous input functions cannot be applied to functions $f_u, f_v : \mathbb{R} \to \{0,1\}$ that represent spike trains, since these functions have the property that for any $\delta \leq 1$ the condition $|f_u(t) - f_v(t)| < \delta$ implies already that $f_u(t) = f_v(t)$, i.e., $t \in u \Longleftrightarrow t \in v$. One also expects from any function F on spike trains that can be realistically computed by a neural circuit that its output Fu does not depend on hair-trigger decisions regarding the precise timing of spikes in the input u. Instead, one expects that infinitesimal changes of firing times in the input spike trains cause only infinitesimal changes in the output value of F. In addition one does not expect from any biologically realistic computation on spike trains that its current output depends in an essential manner on the spike times of infinitely many input spikes. Both of these conditions are formalized in the subsequent definition of fading memory on spike trains. We assume here always that the output of F is some smooth function of time, such as the currents (or changes in membrane voltage) that a neural circuit triggers in subsequent neurons to which it is synaptically connected. Hence the subsequent theory cannot be applied directly to the case where the output of the computation of a neural circuit is the spike train of some neuron ν, but it can be applied to model the time course of the currents that are injected into such output neuron ν by a neural circuit.

Definition:[5] *Informally* a filter $F : U_\Delta \to \mathbb{R}^\mathbb{R}$ has *fading memory on spike trains* if for any spike trains $u, v \in U_\Delta$ and any time point t the difference $|(Fu)(t) - (Fv)(t)|$ becomes arbitrarily small in case that (for some sufficiently large $m \in \mathbb{N}$) the spike times of the last m spikes in u are sufficiently close to the spike times of the corresponding spikes in v.

The formal definition is somewhat more complex, since we have to take into account that u and v may contain different numbers of spikes. In that case one has to demand that all "extra spikes" in one of the two spike trains occurred

[5] An alternative definition of fading memory on spike trains is made explicit in the proof of Theorem 4.1 (see Lemma 4.6): A time invariant filter $F : U_\Delta \to \mathbb{R}^\mathbb{R}$ has fading memory on spike trains if and only if there exists for every $u \in U_\Delta$ and every $\varepsilon > 0$ some $\delta > 0$ such that $|(Fu \cap (-\infty, 0])(0) - (Fv \cap (-\infty, 0])(0)| < \varepsilon$ for every $v \in U_\Delta$ with

$$\int\limits_{-\infty}^{0} \frac{|C_u(t) - C_v(t)|}{t^2}\, d\, t < \delta \,,$$

where C_u is a smooth function from $\mathbb{R}$ into $\mathbb{R}$ that results form replacing each "spike" $s \leq 0$ in u by some continuous "hill" (C_v is defined analogously).

sufficiently far back in the past. *Formally*, we say that a filter $F : U_\Delta \to \mathbb{R}^\mathbb{R}$ has *fading memory on spike trains* if for every $u \in U_\Delta$ and every $\varepsilon > 0$ the following holds: There exist $\delta > 0$ and $m \in \mathbb{N}$ such that $|(Fu)(t) - (Fv)(t)| < \varepsilon$ for every $v \in U_\Delta$ that satisfies the following conditions:

i) if $|u \cap (-\infty, t]| \geq m$ then $|v \cap (-\infty, t]| \geq m$ and for each $k \leq m$ the k^{th} last point in $u \cap (-\infty, t]$ has distance $\leq \delta$ from the k^{th} last point in $v \cap (-\infty, t]$

ii) if $m_u := |u \cap (-\infty, t]| < m$ then $|v \cap (-\infty, t]| \geq m_u$, and for each $k \leq m_u$ the k^{th} last point in $u \cap (-\infty, t]$ has distance $\leq \delta$ from the k^{th} last point in $v \cap (-\infty, t]$, and all other points in $v \cap (-\infty, t]$ are $\leq -m$.

The class of filters on spike trains that satisfy these conditions is very large. In fact, one might argue that it contains any map from spike trains (e. g. spike trains form sensory neurons) to muscle activations that a behaving organism might need to compute in order to survive.

The following theorem exhibits a sufficient condition for showing that neural circuits can approximate any given map F from spike trains into real numbers that belong to the previously defined “computational universe”. Provided that there exists for any two different spike trains u and v a neural circuit or circuit component $B_{u,v}$ whose current state (e. g. membrane potential of neurons in $B_{u,v}$) is different depending on which of the two spike trains u or v was previously sent to this circuit component, circuits that are assembled from finitely many such components $B_{u,v}$ can approximate with any given degree of precision any given time invariant filter F that has fading memory on spike trains.

Theorem 4.1 *Let U_Δ (for some $\Delta > 0$) be the class of spike trains u with distance $\geq \Delta$ for any two spikes in u. Assume that B is an arbitrary class of time invariant filters over U_Δ^n that have fading memory on spike trains, and that B has the pointwise separation property on U_Δ (i. e., for any $u, v \in U_\Delta$ with $u \cap (-\infty, 0] \neq v \cap (-\infty, 0]$ there exists some $B \in B$ with $(Bu)(0) \neq (Bv)(0)$). Furthermore assume that F is an arbitrary class of functions that has the approximation property. Then any given time invariant filter $F : U_\Delta^n \to \mathbb{R}^\mathbb{R}$ with fading memory on spike trains can be approximated arbitrarily closely by LSMs $M = \langle L^M, f^M \rangle$ with L^M composed from finitely many basis filters in B and $f^M \in F$ (formally: $\forall\ \varepsilon > 0\ \ \exists\ M\ \ \forall\ u \in U_\Delta^n\ \ \forall\ t \in \mathbb{R}\ \ |(Fu)(t) - (Mu)(t)| < \varepsilon$).*

Proof of Theorem 4.1: We will just consider the notationally simpler case $n = 1$; the case $n > 1$ is handled analogously. Since all filters involved are assumed to be time invariant, and since U_Δ is closed under translation, it suffices to focus on the time point $t = 0$. We show that there exists a metric d over $U_\Delta^- := \{u \cap (-\infty, 0] : u \in U_\Delta\}$ that turns this set into a compact metric space, and which also has the property that fading memory on spike trains is equivalent to continuity over this metric space $\langle U_\Delta^-, d\rangle$. The proof of Theorem 4.1 follows then from the Stone-Weierstrass Theorem just like the proof of Theorem 3.1. In order to define the metric d we first associate with any spike train $u \subseteq (-\infty, 0]$ a continuous function $C_u : (-\infty, 0] \to \mathbb{R}$. We define for $s, t \leq 0$ the "tent map" with center s by

$$T_s(t) := \begin{cases} 1 - (s - t)\ , & \text{if} \quad s - 1 \leq t \leq \min(s + 1, 0) \\ 0 & , \quad \text{else}\ , \end{cases}$$

and

$$C_u(t) : = \sum_{s \in u} T_s(t)\ .$$

Thus each $s \in u$ is first replaced by a continuous function $T_s(t)$ that reaches its maximal value at $t = s$, and then these continuous functions $T_s(\cdot)$ are added up for all $s \in u$. Since the density of any $u \in U_\Delta^-$ is limited, this function $C_u(t)$ has a bounded value for any $t \leq 0$, is continuous and piecewise linear.

We define the desired metric d over U_Δ^- by

$$d(u, v) : = \int\limits_{-\infty}^{0} \frac{|C_u(t) - C_v(t)|}{t^2}\, dt\, .$$

Thus we first replace the spike trains u and v by smooth functions C_u and C_v (reminiscent of low pass filtering) and then measure the difference between these smooth functions in some straightforward manner, giving less weight to differences between $C_u(t)$ and $C_v(t)$ for strongly negative values t in order to arrive at an integral that converges for any $u, v \in U_\Delta^-$. The continuous functions C_u, C_v could also be defined in other ways for this purpose, as long as the proofs of subsequent lemmata (especially Lemma 4.2) carry over to these alternative definitions.

It requires a little bit of effort to verify that the previously defined function $d : U_\Delta^- \times U_\Delta^- \to \mathbb{R}$ satisfies the axioms for a metric (see [19]). Whereas in this case it is trivial to verify the triangle inequality, it is harder to verify the seemingly obvious axiom that represents the claim of the following lemma (whose proof is given in the Appendix).

Lemma 4.2 *$d(u, v) = 0 \Rightarrow u = v$ for any $u, v \in U_\Delta^-$.*

In order to prove that the metric space $\langle U_\Delta^-, d\rangle$ is compact, and that continuity with respect to d is equivalent to fading memory on spike trains, we first prove auxiliary lemmata that allow us to express convergence with regard to the metric d in terms of relationships between spike times of the spike trains involved. We write $|v|$ for the number of spikes in a spike train $v \in U_\Delta^-$ (thus $|v| \in \mathbb{N} \cup \{\infty\}$). For any $k \in \mathbb{N}$ and any $v, v_1, v_2, \ldots \in U_\Delta^-$ we formalize the statement that it looks as if $(v_i)_{i\in\mathbb{N}}$ converges to v if one just focuses on the k^{th} most recent spike in each of the spike trains involved as follows:

$$Q(k, (v_i)_{i\in\mathbb{N}}, v) :\Leftrightarrow$$

$$\begin{gathered}(|v| \geq k \Rightarrow \forall \delta > 0 \exists i_0 \forall i \geq i_0 (|v_i| \geq k \text{ and the} \\ k^{th} \text{ last spike in } v_i \text{ has distance } \leq \delta \text{ from the } k^{th} \text{ last spike in } v) \wedge \\ (|v| < k \Rightarrow \forall T > 0 \exists i_0 \forall i \geq i_0\ (|v_i| \geq k \Rightarrow \text{ the } k^{th} \text{ largest spike in } v_i \\ \text{is not contained in } [-T, 0])).\end{gathered}$$

It will be shown in Lemma 4.3 and 4.5 that $\forall k \in \mathbb{N}\ Q(k, (v_i)_{i\in\mathbb{N}}, v)$ is equivalent to $d(v_i, v) \to 0$ for $i \to \infty$.

Lemma 4.3 *For any $v, v_1, v_2, \ldots \in U_\Delta^-$ one has $\forall k \in \mathbb{N}\ Q(k, (v_i)_{i\in\mathbb{N}}, v) \Rightarrow d(v_i, v) \to 0$ for $i \to \infty$.*

The **proof of Lemma 4.3** is given in the Appendix. With the help of Lemma 4.3 it is not difficult to prove:

Lemma 4.4 *The metric space $\langle U_\Delta^-, d\rangle$ is compact.*

[**Proof of Lemma 4.4**] According to Lemma 4.3 it suffices to show that for any $u_1, u_2, \ldots \in U_\Delta^-$ there exists some $u \in U_\Delta^-$ and a subsequence $(\tilde{u}_i)_{i\in\mathbb{N}}$ of $(u_i)_{i\in\mathbb{N}}$ so that $\forall k \in \mathbb{N}\ Q(k, (\tilde{u}_i)_{i\in\mathbb{N}}, u)$ holds. For each $i \in \mathbb{N}$ let $u_i(1) > u_i(2) > \ldots$ be the elements of the set $u_i \subseteq (-\infty, 0]$ in descending order, with $u_i(k) := -\infty$ if u_i is a finite set with fewer than k elements. We construct now by recursion on $l \in \mathbb{N}$ a subsequence $(u_i^{(l)})_{i\in\mathbb{N}}$ of $(u_i)_{i\in\mathbb{N}}$ and a sequence $u^{(l)} := \{u(1), \ldots, u(m)\}$ with $0 \geq u(1) > u(2) > \ldots > u(m)$, where $0 \leq m \leq l$ (the set $u^{(l)}$ is assumed to be empty if $m = 0$) so that $\forall k \leq l\ Q(k, (u_i^{(l)})_{i\in\mathbb{N}}, u^{(l)})$ if $m = l$ and $\forall k \in \mathbb{N}\ Q(k, (u_i^{(l)})_{i\in\mathbb{N}}, u^{(l)})$ otherwise

$\mathbf{l = 1}$:

Consider the set $\{u_i(1) : i \in \mathbb{N}\}$. If there exists some $T > 0$ so that $u_i(1) \in [-T, 0]$ for infinitely many i, choose some $u(1) \in [-T, 0]$ and a subsequence $(u_i^{(1)})_{i\in\mathbb{N}}$ of $(u_i)_{i\in\mathbb{N}}$ so that $u_i^{(1)}(1)$ converges to $u(1)$ for $i \to \infty$. This implies that $Q(1, (u_i^{(1)})_{i\in\mathbb{N}}, u^{(1)})$ for $u^{(1)} := \{u(1)\}$. Otherwise set $m = 0$ (thus $u^{(1)} = \emptyset$) and $u_i^{(1)} = u_i$ for all $i \in \mathbb{N}$. Since either $|\{u_i(1) : i \in \mathbb{N}\}| < \infty$, or $|\{u_i(1) : i \in \mathbb{N}\}| = \infty$ and $\lim_{i\to\infty} u_i(1) = -\infty$, this implies that $\forall k \in \mathbb{N}\ Q(k, (u_i^{(1)})_{i\in\mathbb{N}}, u^{(1)})$.

$\mathbf{l-1 \mapsto l}$:

Assume that $u^{(l-1)} = \{u(1), u(2), \ldots, u(m)\}$ with $m \leq l-1$ and $(u_i^{(l-1)})_{i\in\mathbb{N}}$ have already been constructed so that either $m < l-1$ and $\forall k \in \mathbb{N}\ Q(k, (u_i^{(l-1)})_{i\in\mathbb{N}}, u^{(l-1)})$, or $m = l-1$ and $\forall k \leq l-1\ Q(k, (u_i^{(l-1)})_{i\in\mathbb{N}}, u^{(l-1)})$. In the former case we define $u^{(l)} = u^{(l-1)}$ and $u_i^{(l)} = u_i^{(l-1)}$ for all $i \in \mathbb{N}$, and the claim is obvious. In the latter case we check whether there exists some $T > 0$ so that $u_i^{(l-1)}(l) \in [-T, 0]$ for infinitely many i. If yes, we choose some $u(l) \in [-T, 0]$ with $u(l) < u(l-1)$ and a subsequence $(u_i^{(l)})_{i\in\mathbb{N}}$ of $(u_i^{(l-1)})_{i\in\mathbb{N}}$ so that $u_i^{(l)}(l)$ converges to $u(l)$ for $i \to \infty$ (this implies that $\forall k \leq l\ Q(k, (u_i^{(l)})_{i\in\mathbb{N}}, u^{(l)})$). If not, we set $m := l-1$ and define $u_i^{(l)} := u_i^{(l-1)}$ for all $i \in \mathbb{N}$. We then have $\forall k \in \mathbb{N}\ Q(k, (u_i^{(l)})_{i\in\mathbb{N}}, u^{(l)})$. Finally we define the desired subsequence $(\tilde{u}_i)_{i\in\mathbb{N}}$ of $(u_i)_{i\in\mathbb{N}}$ by setting $\tilde{u}_i := u_i^{(i)}$, and a set $u := \bigcup_{l\in\mathbb{N}} u^{(l)}$. It is then obvious that $\forall k \in \mathbb{N}\ Q(k, (\tilde{u}_i)_{i\in\mathbb{N}}, u)$.

With the help of the preceding Lemma 4.4 we can now prove the converse of Lemma 4.3:

Lemma 4.5 *For any $u, u_1, u_2, \ldots \in U_\Delta^-$ one has $d(v_i, v) \to 0$ for $i \to \infty \Rightarrow \forall k \in \mathbb{N}\ Q(k, (v_i)_{i\in\mathbb{N}}, v)$.*

The **proof of Lemma 4.5** is given in the Appendix. Lemma 4.3. and Lemma 4.5 together allow us to derive the following essential lemma for the proof of Theorem 4.1. It provides a link between the notion of fading memory on spike trains and the notion of continuity with regard to the metric d, to which the Stone Weierstrass Theorem can be applied.

Lemma 4.6 *A time invariant filter $F : U_\Delta \to \mathbb{R}^\mathbb{R}$ has fading memory on spike trains if and only if the function $F^- : U_\Delta^- \to \mathbb{R}$ defined by $F^-(u^-) := (Fu)(0)$ (for some arbitrary $u \in U_\Delta$ with $u \cap (-\infty, 0] = u^-$) is well-defined and continuous with respect to the metric d on U_Δ^-.*

[**Proof of Lemma 4.6**]

"$\Rightarrow$" If F has fading memory on spike trains then the value of $(Fu)(0)$ only depends on $u \cap (-\infty, 0]$, hence F^- is well-defined. In order to prove that F^- is continuous with respect to the metric d we assume that some $u \in U_\Delta^-$ and some $\varepsilon > 0$ have been given. We need to show that there exists some $\delta > 0$ so that $|F^-u - F^-v| < \varepsilon$ for all $v \in U_\Delta^-$ with $d(u, v) < \delta$.

Assume for a contradiction that such δ does not exist. Then there exists for every $m \in \mathbb{N}$ some $u_m \in U_\Delta^-$ with $d(u_m, u) < \frac{1}{m}$ and $|F^-u_m - F^-u| \geq \varepsilon$. Thus we have $d(u_m, u) \to 0$ for $m \to \infty$, and hence $\forall\ k \in \mathbb{N}\ Q(k, (u_m)_{m\in\mathbb{N}}, u)$ according to Lemma 4.5. But that implies $|F^-u_m - F^-u| \to 0$ by the definition of fading memory on spike trains, a contradiction.

"$\Leftarrow$" Since F is assumed to be time invariant it suffices to show that for every $u \in U_\Delta$ and every $\varepsilon > 0$ there exist $\delta > 0$ and $m \in \mathbb{N}$ so that $|(Fu)(0) - (Fv)(0)| < \varepsilon$ for all $v \in U_\Delta$ such that conditions i) and ii) of the definition of fading memory on spike trains hold for u, v, δ, m. Since F^- is continuous with respect to the metric d, we know that there exists some $\delta^- > 0$ so that $|F^-(u \cap (-\infty, 0]) - F^-(v \cap (-\infty, 0])| < \varepsilon$ for all $v \in U_\Delta$ with $d(u \cap (-\infty, 0],\ v \cap (-\infty, 0]) < \delta^-$. But it is obvious that there exist $\delta > 0$ and $m \in \mathbb{N}$ so that one has $d(u \cap (-\infty, 0],\ v \cap (-\infty, 0]) < \delta^-$ for all $v \in U_\Delta$ such that conditions i) and ii) hold for u, v, δ, m. Hence one has $|(Fu)(0) - (Fv)(0)| < \varepsilon$ for all these $v \in U_\Delta$. □

In order to complete the proof of Theorem 4.1 we just have to apply the Stone-Weierstrass Theorem to functions $F^- : U_\Delta^- \rightarrow \mathbb{R}$ that are continuous with respect to the metric d on U_Δ^-. The Stone-Weierstrass Theorem can be applied to these functions since $\langle U_\Delta^-, d \rangle$ is a compact metric space according to Lemma 4.4. According to Lemma 4.6 this yields the desired statement about filters with fading memory on spike trains.

5 Conclusions

We have examined in this article a new framework for the analysis of the computational power of neural circuits. Whereas there exists a well-established computational theory for batch-computing on digital input (see for example [26]), the biologically more realistic case of real-time computing on fast varying analog input hat remained largely unexplored. In order to eliminate unrealistic types of such computations, where infinite bit precision of some input $u(t)$ at time t becomes relevant, or where values of $u(t)$ for an infinitely long interval of time points t matter, we have proposed to focus on fading memory computations where such pathological cases do not occur. On the other hand, since the fading memory concept leaves open how fast dependence on previous input segments is fading, it subsumes also all biologically relevant computations that involve memory or temporal integration. We have shown that within this context of fading memory computations universal computational power can be achieved by a class of circuits under rather weak conditions. One just has to assume that this class of circuits satisfies the obviously necessary condition that any two different inputs u, v can be separated by a subsequent circuit state. Since randomly connected recurrent circuits of spiking neurons tend to have this property for fairly large classes of inputs [21], in particular if they are sufficiently large and heterogeneous, one arrives in this way at a theoretical foundation for the possible computational use of such circuits. A more systematic study of the separation property of randomly connected threshold circuits has recently been given in [23], based on preceding work on computations at the edge of chaos in dynamical systems. There is some hope that some of these results can be transferred to networks of spiking neurons, thereby providing tools for a more principled analysis of the separation property of such circuits for online spike input.

Previous approaches for providing positive evidence for the computational power of circuits of spiking neurons was based on strategies to implement finite automata or specific well-understood and highly structured circuit architectures with circuits of spiking neurons. These approaches had only been able to explain relatively simple computations with spiking neurons in a noise-robust manner. In contrast, the theory presented in this article provides a possible theoretical explanation for the computational power of the complex and highly recurrent circuits consisting of neurons and dynamic synapses with a diverse set of time constants, which emerge as the biologically more realistic models from detailed empirical studies of neocortical microcircuits ([10], [11]).

Another aspect in which the theoretical framework presented in this article differs from previous approaches is that it de-emphasizes the need to identify a clear neural representation for each external stimulus to an organism: it suggests instead that *neural separation is more important than neural representation*. In other words: it suffices that saliently different external stimuli leave significantly different traces in the subsequent activity of neural circuits, even if there is no clear neural code by which these traces encode these stimuli. This approach is consistent with the fact that in many experiments the initial state of the neural circuit varies from trial to trial: traces are piled up on top of other traces that were caused by earlier external or internal inputs to the neural circuit. Neural separation may still be guaranteed even if there exists no invariant neural representation of specific stimuli.

Our theoretical approach suggests that purposeful real-time processing of sensory stimuli is possible just on the basis of neural separation (rather than representation), since readout modules can easily be trained to assign target outputs to complex circuit states, even if there is no simple rule that makes this assignment easy (from the point of view of a human observer). This prediction relies on the fact that almost any classification task can be carried out by a linear separator (or a small pool of linear separators) if the patterns that need to be classified are first projected nonlinearly into a fairly high-dimensional space, even if this nonlinear projection is very complicated (in a biological context this nonlinear projection would be defined by the current state of activation of a neural microcircuit resulting from the injection of some input pattern). The whole approach towards pattern recognition via support vector machines in machine learning, see [27], relies on this effect. Also in this regard the theory presented in this article is complementary to preceding approaches, since it works particularly well for those types of circuits where other approaches have difficulties: for fairly large recurrent circuits consisting of heterogeneous neurons and dynamic synapses that respond in a complex nonlinear way to incoming input. The theoretical prediction that the readout from a recurrent circuit is easier if the circuit is fairly large may suggest on first sight that this computational model is rather uneconomical, since it requires so many neurons in the recurrent circuit. But one should keep in mind that the same recurrent circuit can be used simultaneously by a large number of different readouts, and thereby support simultaneously a large number of different computational tasks (see Figure 4 in [6] and Figure 2 in [21], and [28] for experimental demonstrations of this fact).

According to [9] already feedforward circuits (in fact, already filterbanks consisting of feedforward dynamic synapses with slightly different parameter values) satisfy the separation property for analog inputs. Recurrent connections tend to amplify however the separation of circuit states, hence they tend to make the task of the readout easier and more noise robust. In addition it was argued in [21] that recurrent connections enable a circuit to compute nonlinear combinations of input features. On the basis of such fixed nonlinear preprocessing even simple *linear* readouts gain most of the classification power of a more powerful universal approximator such as multi-layer perceptrons. Hence the recurrent connections within a circuit may provide similar advantages as a fixed nonlinear kernel for support vector machines (see [27]).

The computational theory presented in this article is based on concepts such as fading memory filters that are also relevant for analog computing in a realistic noisy setting, but the theory itself applies so far only to the noisefree (deterministic) case. In a realistic scenario with noise the pointwise separation property becomes less relevant. Instead it matters whether "significantly different" input histories u, v result in current circuit states that are "sufficiently different" so that they can be distinguished even in the presence of noise by a suitably trained readout. Such more quantitative version of the separation property has been studied empirically in [6] (see Fig. 2). Furthermore it has been investigated analytically for the simpler case of randomly connected recurrent threshold circuits in [23].

The investigation of this more quantitative version of the separation property also provides hints for understanding the possible computational advantage of having long term synaptic plasticity *within* the recurrent circuit (rather than just at the synapses to the readout, as considered in this article): Such long term synaptic plasticity within the recurrent circuit may serve to modulate the resolution of the separation property of the circuit so that it achieves larger separation for input differences that are relevant for the organism, and smaller separation for behaviorally irrelevant input differences. In the language of dynamical systems theory one can formulate another potential goal of long term synaptic plasticity within the recurrent circuit: keeping the dynamics of the circuit (for practically relevant online inputs) at the "edge of chaos" [29], where the circuit is sufficiently sensitive to reflect recent salient input differences in its current state, but is not yet chaotic (i.e., it does not produce large separation for any tiny input differences way back in the past).

In contrast to most previous work on computations in models for recurrent neural circuits, the computations considered in this article were online computations, rather than computations on batch-input. Consequently attractors, which commonly encode the results of computations on batch-inputs in recurrent circuits, were not considered in this article. However, new work in progress by Joshi and Maass shows that feedbacks from readouts into the recurrent circuits may be used to create attractors for *some* components of a high-dimensional circuit state, in which intermediate results of computations may be stored for periods of a few seconds and longer ("working memory"). In this way some of the computational advantages of attractor neural networks can be transferred to the model for real-time computing on continuous input streams discussed in this article.

Herbert Jaeger (see [7] and [30]) has discovered independently the power of liquid state machines in artificial neural networks with discrete time, showing for example that they may yield novel solutions to difficult time series prediction and control problems. We have shown in section 2 that circuits with fading memory of the type considered in [7] give also rise to fading memory filters, and hence provide another application domain for the theoretical results of this article.

The concepts and results of this article are not suitable for answering the question how many basis filters $B_1, \ldots, B_m$ from the class B of basis filters may be needed to approximate a given filter F up to a certain degree of precision. Such bounds on the speed of convergence are very rare even in the case of neural computation on static inputs, i.e. for standard artificial neural networks, and the only mathematical results that provide such bounds (see [31] for a discussion) are almost never used practically. At this point it is not clear whether similar theoretical results for the approximation of filters are feasible, and even if they are found it is dubious whether they would be practically relevant. Hence from the practical point of view it appears to be more fruitful to carry out experimental studies. For the case of LSMs whose liquid filter L is simply some generic recurrent circuit consisting of a few hundred spiking neurons and the readout function is implemented by some other pool of spiking neurons, or even by just a single linear neuron, some quite encouraging experimental results are reported in [6,21,32]. In [25] it is shown that liquid filters L that are composed from a very small number of dynamic synapses as basis filters (which have the pointwise separation property according to [9]) endow liquid state machines with rather good approximation capabilities. Thus so far it appears that the approximation results that were derived in this paper are practically relevant, because their assumptions are approximately satisfied by generic models for neural circuits.

The approximation results derived in this article induce new complexity hierarchies for nonlinear filters that appear to be more useful and flexible than the well-known degree-hierarchy of Volterra polynomials. One can fix any collection B of basis filters that are natural computational units from the point of view of a specific theoretical or practical context (e.g. in the context of modeling biological neural computation the set of filters that are computed by neurons). One can then measure the complexity of other filters F in terms of how many basis filters from B have to be composed in order to approximate F up to a certain degree of precision. If the class B of basis filters satisfies the pointwise separation property, the complexity of any time invariant fading memory filter F can be measured in this way (according to Theorems 3.1 and 4.1).

Finally, it turns out that the new approach towards neural computation that is suggested by the theoretical framework of this article is the first one that allows us to carry out complex computations on basically any computer models of biologically realistic neural circuits, thereby opening up new ways of investigating such circuits. Hence this approach may contribute to an experimental and theoretical basis for understanding the computational function of neural microcircuits in the cortex, and it may provide new ideas for capturing their computational capability in artificial devices.

Acknowledgment:

The first author would like to thank Herbert Jaeger for stimulating discussions. Both authors would like to thank an anonymous referee for helpful suggestions. The work was partially supported by project # P15386 of the Austrian Science Fund, PASCAL project # IST2002-506778 of the European Union, the Office of Naval Research, HFSP, Dolfi & Ebner Center and the Edith Blum Foundation.

References

[1] G. Silberberg, A. Gupta, H. Markram, Stereotypy in neocortical microcircuits, Trends Neurosci. 25 (5) (2002) 227–230.

[2] P. Z. Marmarelis, V. Z. Marmarelis, Analysis of Physiological Systems: The White-Noise Approach, Plenum Press, New York, 1978.

[3] W. J. Rugh, Nonlinear Systems Theory, John Hopkins University Press, Baltimore, 1981.

[4] T. Poggio, W. Reichardt, On the representation of multi-input systems: computational properties of polynomial algorithms, Biological Cybernetics 37 (1980) 167–186.

[5] F. Rieke, D. Warland, R. R. D. van Steveninck, W. Bialek, SPIKES: Exploring the Neural Code, MIT Press, Cambridge, MA, 1997.

[6] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Computation 14 (11) (2002) 2531–2560, see # 130 on http://www.igi.tugraz.at/maass/publications.html.

[7] H. Jäger, The "echo state" approach to analyzing and training recurrent neural networks, GMD Report 148, German National Research Center for Information Technology (2001).

[8] W. Maass, E. Sontag, Analog neural nets with Gaussian or other common noise distribution
cannot recognize arbitrary regular languages, Neural Computation 11 (1999) 771–782, see # 95 on http://www.igi.tugraz.at/maass/publications.html.

[9] W. Maass, E. D. Sontag, Neural systems as nonlinear filters, Neural Computation 12 (8) (2000) 1743–1772, see # 107 on http://www.igi.tugraz.at/maass/publications.html.

[10] H. Markram, Y. Wang, M. Tsodyks, Differential signaling via the same axon of neocortical pyramidal neurons, Proceedings of the National Academy of Sciences USA 95 (1998) 5323–5328.

[11] A. Gupta, Y. Wang, H. Markram, Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex, Science 287 (2000) 273–278.

[12] J. J. Hopfield, C. D. Brody, What is a moment? Transient synchrony as a collective mechanism for spatio-temporal integration, Proc. Nat. Acad. Sci. USA 98 (3) (2001) 1282–1287.

[13] D. V. Buonomano, M. M. Merzenich, Temporal information transformed into a spatial code by a neural network with realistic properties, Science 267 (1995) 1028–1030.

[14] P. Auer, H. Burgsteiner, W. Maass, Reducing communication for distributed learning in neural networks, in: J. R. Dorronsoro (Ed.), Proc. of the International Conference on Artificial Neural Networks – ICANN 2002, Vol. 2415 of Lecture Notes in Computer Science, Springer, 2002, pp. 123–128, see # 127 on http://www.igi.tugraz.at/maass/publications.html.

[15] W. Maass, T. Natschläger, A model for fast analog computation based on unreliable synapses, Neural Computation 12 (7) (2000) 1679–1704, see # 102 on http://www.igi.tugraz.at/maass/publications.html.

[16] R. A. Brooks, Intelligence without representation, Artificial Intelligence 47 (1991) 139–159.

[17] S. Boyd, L. O. Chua, Fading memory and the problem of approximating nonlinear oparators with Volterra series, IEEE Trans. on Circuits and Systems 32 (1985) 1150–1161.

[18] I. W. Sandberg, Structure theorems for nonlinear systems, Multidimensional Systems and Signal Processing 2 (1991) 267–286.

[19] J. Dieudonne, Foundations of Modern Analysis, Academic Press, New York, 1969.

[20] W. Maass, Paradigms for computing with spiking neurons, in: J. L. van Hemmen, J. D. Cowan, E. Domany (Eds.), Models of Neural Networks. Early Vision and Attention, Vol. 4, Springer (New York), 2002, Ch. 9, pp. 373–402, see # 110 on http://www.igi.tugraz.at/maass/publications.html.

[21] W. Maass, T. Natschläger, H. Markram, Computational models for generic cortical microcircuits, in: J. Feng (Ed.), Computational Neuroscience: A Comprehensive Approach, Chapman & Hall/CRC, Boca Raton, 2004, Ch. 18, pp. 575–605, see # 149 on http://www.igi.tugraz.at/maass/publications.html.

[22] C. A. van Vreeswijk, H. Sompolinsky, Chaos in neuronal networks with balanced excitatory and inhibitory activity, Science 274 (1996) 1724–1726.

[23] N. Bertschinger, T. Natschläger, Real-time computation at the edge of chaos in recurrent neural networks, Neural ComputationIn press.

[24] W. Maass, On the computational power of winner-take-all, Neural Computation 12 (11) (2000) 2519–2536, see # 113 on http://www.igi.tugraz.at/maass/publications.html.

[25] T. Natschläger, W. Maass, Spiking neurons and the induction of finite state machines, Theoretical Computer Science: Special Issue on Natural Computing 287 (2002) 251–265, see # 136 on http://www.igi.tugraz.at/maass/publications.html.

[26] J. E. Savage, Models of Computation: Exploring the Power of Computing, Addison-Wesley (Reading, MA), 1998.

[27] V. N. Vapnik, Statistical Learning Theory, John Wiley (New York), 1998.

[28] R. A. Legenstein, H. Markram, W. Maass, Input prediction and autonomous movement analysis in recurrent circuits of spiking neurons, Reviews in the Neurosciences 14 (1-2) (2003) 5–19, see # 140 on http://www.igi.tugraz.at/maass/publications.html.

[29] C. G. Langton, Computation at the edge of chaos, Physica D 42 (1990) 12–37.

[30] H. Jäger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and boosting wireless communication, ScienceIn press.

[31] E. D. Sontag, Neural networks for control, in: J. C. W. H. L. Trentelman (Ed.), Essay on Control: Perspectives in the Theory and its Applications, Birkhauser, Boston, 1993, pp. 339–380, online available from: http://www.math.rutgers.edu/˜sontag/papers.html.

[32] S. Häusler, H. Markram, W. Maass, Perspectives of the high dimensional dynamics of neural microcircuits from the point of view of low dimensional readouts, Complexity (Special Issue on Complex Adaptive Systems) 8 (4) (2003) 39–50, see # 137 on http://www.igi.tugraz.at/maass/publications.html.

[33] W. Rudin, Real and Complex Analysis, McGraw-Hill, New York, 1987.

6 Appendix

[**Proof of Lemma 3.3**] One can easily show that the function d defined by $d(u,v) := \sup_{t\leq 0}\frac{|u(t)-v(t)|}{1+|t|}$ satisfies the axioms for a metric. For example, the triangle inequality is verified as follows for any $u,v,w\in U^-$: $d(u,v) := sup_{t\leq 0}\frac{|u(t)-v(t)|}{1+|t|} \leq \sup_{t\leq 0}\frac{|u(t)-w(t)|+|w(t)-v(t)|}{1+|t|} \leq \sup_{t\leq 0}\frac{|u(t)-w(t)|}{1+|t|} + \sup_{t\leq 0}\frac{|w(t)-v(t)|}{1+|t|} = d(u,w)+d(w,v)$.

In order to prove that the metric space $\langle U^-, d\rangle$ is compact, it suffices to verify that there exists for any sequence $(u_k)_{k\in\mathbb{N}}$ of functions from U^- some $v\in U^-$ and some subsequence $(\tilde{u}_i)_{i\in\mathbb{N}}$ of $(u_k)_{k\in\mathbb{N}}$ such that $\lim_{i\to\infty} d(\tilde{u}_i, v) = 0$. But this is an immediate consequence of the Arzela-Ascoli Theorem (see for example [33]).

Technical Remark: The proof of Lemma 3.3 shows that it is not necessary to assume that the functions in U are uniformly Lipschitz-continuous. It suffices if the functions in U are equicontinuous (i.e., for every $\varepsilon\in 0$ there exists a $\delta > 0$ such that $|u(x)-u(y)| < \varepsilon$ for every $u\in U$ and any $x,y\in\mathbb{R}$ with $|x-y| < \delta$).

[**Proof of Lemma 4.2**] The claim is not completely obvious, since in the larger domain U_Δ there exist in fact different spike trains u,v with $C_u = C_v$: define for example $u := \mathbb{Z}, v := \{z + \frac{1}{2} : z\in\mathbb{Z}\}$.

Assume for a contradiction that there exist $u,v\in U_\Delta^-$ with $u\neq v$ but $d(u,v) = 0$. The latter implies that $C_u(t) = C_v(t)$ for all $t\leq 0$ (since C_u and C_v are continuous functions), and hence also that their derivatives $C_u^{'}, C_v^{'}$ have the same value for any $t\leq 0$ where these derivatives exist (since C_u, C_v are piecewise linear, their derivatives are piecewise constant and do not exist at those points where linear pieces with different slopes meet).

Let $s_0\leq 0$ be a maximal point in the symmetric difference $(u-v)\cup(v-u)$ of the two different sets u and v. We can assume without loss of generality that $s_0\in u-v$.

Case 1: $s_0 = 0$

We first show that $u \cap (-1,0) = v \cap (-1,0)$. For that we actually do not even need the assumption $s_0 = 0$. According to the definition of the piecewise linear function C_u every point $t \in u \cap (-1,0)$ causes at t a downwards jump by -2 or -1 of the piecewise constant derivative $C_u{}'$, independently of the other points in u (a downwards jump by -1 occurs at t only if $t-1 \in u$; note that $t > -1$ implies that $t+1 \notin u$). Furthermore these are the only points in $(-1,0)$ where downwards jumps of $C_u{}'$ occur. Hence $C_u{}' = C_v{}'$ implies that $u \cap (-1/0) = v \cap (-1,0)$. But then the assumption $0 = s_0 \in u - v$ implies that $C_u(0) = C_v(0) + 1$, thus $C_u \neq C_v$, a contradiction.

Case 2: $s_0 < 0$
We then have $S := u \cap (s_0, 0] = v \cap (s_0, 0]$. Set $\tilde{u} := u - S$ and $\tilde{v} := v - S$. Since $C_u = C_v$ we also have $C_{\tilde{u}} = C_{\tilde{v}}$, hence $C_{\tilde{u}}^{'} = C_{\tilde{v}}^{'}$. On the other hand $s_0 \in \tilde{u} - \tilde{v}$ and $\tilde{u} \cap (s_0, 0] = \emptyset$ imply that the piecewise constant function $C_{\tilde{u}}^{'}$ has a downwards jump at s_0, whereas $C_{\tilde{v}}^{'}$ has no downwards jump at s_0 (since $s_0 \notin \tilde{v}$), a contradiction to $C_{\tilde{u}}^{'} = C_{\tilde{v}}^{'}$.

[**Proof of Lemma 4.3**] Fix some arbitrary $\varepsilon \in 0$. We will show that there exists some i_0 so that $d(v_i, v) \leq \varepsilon$ for all $i \geq i_0$. Choose $T > 0$ sufficiently large so that $\int_{-\infty}^{-T} \frac{|C_u(t) - C_{\tilde{u}}(t)|}{t^2} dt \leq \frac{\varepsilon}{2}$ for any $u, \tilde{u} \in U_{\Delta}^{-}$.

Case 1: $|v| < \infty$
By the assumption of the Lemma there exists some $\tilde{i}$ such that for every $i \geq \tilde{i}$ and every $k > |v|$ the k^{th} largest point of v_i is $\leq -T-2$. Obviously these points $\leq -T-2$ have no relevance for the value of $\int_{-\infty}^{0} \frac{|C_{v_i}(t) - C_v(t)|}{t^2} dt$. By choosing $i_0 \geq \tilde{i}$ sufficiently large we can achieve that for $i \geq i_0$ all the other points of v_i (i.e., the k^{th} largest points for $k \leq |v|$) lie so close to the corresponding points of v that $\int_{-T}^{0} \frac{|C_{v_i}(t) - C_v(t)|}{t^2} dt \leq \frac{\varepsilon}{2}$.

Case 2: $|v| = 0$
Choose $k_0 \in \mathbb{N}$ so large that the k_0^{th} largest point of v is $\leq -T-2$. Then it suffices to choose i_0 sufficiently large so that for all $i \geq i_0$ and all $k \leq k_0$ the k^{th} largest point in v_i lies so close to the k^{th} largest point in v that $\int_{-T}^{0} \frac{|C_{v_i}(t) - C_v(t)|}{t_2} dt \leq \frac{\varepsilon}{2}$.

[**Proof of Lemma 4.5**] Assume that the claim is wrong, and choose $k \in \mathbb{N}$ minimal so that $Q(k, (v_i)_{i \in \mathbb{N}}, v)$ does not hold.

Case 1: $k \leq |v|$
If there were infinitely many i such that $|v_i| < k$ this would contradict the assumption that $d(v_i, v) \to 0$ for $i \to \infty$ (since C_v is a superposition of $\geq k$ functions T_s). Hence there exists some $\varepsilon > 0$ and infinitely many $i \in \mathbb{N}$ so that $|v_i| \geq k$ and the

k^{th} largest point of v_i has distance $\geq \varepsilon$ from the k^{th} largest point in v. According to Lemma 4.3 and according to the proof of Lemma 4.4 there exists an infinite subsequence $(\tilde{v}_i)_{i\in\mathbb{N}}$ of these v_i and some $u \in U_\Delta^-$ so that $d(\tilde{v}_i, u) \to 0$ for $i \to \infty$ and $\forall k \in \mathbb{N}\ Q(k, (\tilde{v}_i)_{i\in\mathbb{N}}, u)$. Since the k^{th} largest point of $\tilde{v}_i$ and v have distance $\geq \varepsilon$, this implies that $|u| < k$ or that the k^{th} largest point of u has distance $\geq \varepsilon$ from the k^{th} largest point of v, thus in either case $u \neq v$. According to Lemma 4.2 this implies that $d(u, v) > 0$. But then it is impossible that $d(\tilde{v}_i, u) \to 0$ for $i \to \infty$ and $d(\tilde{v}_i, v) \to 0$ for $i \to \infty$ (by the triangle inequality for the metric d).

Case 2: $k > |v|$
Then there exist infinitely many $i \in \mathbb{N}$ and some $T > 0$ that $|v_i| \geq k$ and k^{th} largest point of v_i is $\geq -T$. But since $Q(k', (v_i)_{i\in\mathbb{N}}, v)$ holds for all $k' < k$ (by the minimal choice of k), we have for all $m \leq |v|$ that $|v_i| \geq m$ for sufficiently large i, and the m^{th} largest point of the v_i converge to the m^{th} largest point of v. Define u_i as the subset of v_i consisting of the $|v|$ largest points of v_i. Thus we have $d(u_i, v) \to 0$ for $i \to \infty$. But this yields a contradiction to the assumption that $d(v_i, v) \to 0$ for $i \to 0$, since there are infinitely many i with $|v_i| \geq k > |v|$ and the k^{th} largest point of v_i is $\geq -T$.