

2007 Special Issue

Edge of chaos and prediction of computational performance for neural circuit models

Robert Legenstein*, Wolfgang Maass

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

Abstract

We analyze in this article the significance of the edge of chaos for real-time computations in neural microcircuit models consisting of spiking neurons and dynamic synapses. We find that the edge of chaos predicts quite well those values of circuit parameters that yield maximal computational performance. But obviously it makes no prediction of their computational performance for other parameter values. Therefore, we propose a new method for predicting the computational performance of neural microcircuit models. The new measure estimates directly the kernel property and the generalization capability of a neural microcircuit. We validate the proposed measure by comparing its prediction with direct evaluations of the computational performance of various neural microcircuit models. The proposed method also allows us to quantify differences in the computational performance and generalization capability of neural circuits in different dynamic regimes (UP- and DOWN-states) that have been demonstrated through intracellular recordings in vivo.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Neural networks; Spiking networks; Edge of chaos; Microcircuits; Computational performance; Network dynamics

1. Introduction

What makes a neural microcircuit computationally powerful? Or more precisely, which measurable quantities could explain why one microcircuit C is better suited for a particular family of computational tasks than another microcircuit C' ? Rather than constructing particular microcircuit models that carry out particular computations, we pursue in this article a different strategy, which is based on the assumption that the computational function of cortical microcircuits is not fully genetically encoded, but rather emerges through various forms of plasticity (“learning”) in response to the actual distribution of signals that the neural microcircuit receives from its environment. From this perspective the question about the computational function of cortical microcircuits C turns into the questions:

(a) What functions (i.e. maps from circuit inputs to circuit outputs) can particular neurons (“readout neurons”, see below) in conjunction with the circuit C learn to compute.

(b) How well can readout neurons in conjunction with the circuit C generalize a specific learned computational function to new inputs?

We propose in this article a conceptual framework and quantitative measures for the investigation of these two questions. In order to make this approach feasible, in spite of numerous unknowns regarding synaptic plasticity and the distribution of electrical and biochemical signals impinging on a cortical microcircuit, we make in the present first step of this approach the following simplifying assumptions:

1. Particular neurons (“readout neurons”) learn via synaptic plasticity to extract specific information encoded in the spiking activity of neurons in the circuit.
2. We assume that the cortical microcircuit itself is highly recurrent, but that the impact of feedback that a readout neuron might send back into this circuit can be neglected.¹

* Corresponding author. Tel.: +43 316 873 5824; fax: +43 316 873 5805.
E-mail addresses: legi@igi.tugraz.at (R. Legenstein), maass@igi.tugraz.at (W. Maass).

¹ This assumption is best justified if such readout neuron is located for example in another brain area that receives massive input from many neurons in this microcircuit and only has diffuse backwards projection. But it is certainly problematic and should be addressed in future elaborations of the present approach.

3. We assume that synaptic plasticity of readout neurons enables them to learn arbitrary linear transformations. More precisely, we assume that the input to such readout neuron can be approximated by a term $\sum_{i=1}^{n-1} w_i x_i(t)$, where $n - 1$ is the number of presynaptic neurons, $x_i(t)$ results from the output spike train of the i th presynaptic neuron by filtering it according to the low-pass filtering property of the membrane of the readout neuron,² and w_i is the efficacy of the synaptic connection. Thus $w_i x_i(t)$ models the time course of the contribution of previous spikes from the i th presynaptic neuron to the membrane potential at the soma of this readout neuron. We will refer to the vector $\mathbf{x}(t)$ as the *circuit state at time t* . Note that the readout neurons do not have access to the analog state of the circuit neurons, but only to the filtered version of their output spike trains.

Under these unpleasant but apparently unavoidable simplifying assumptions we propose in Sections 4 and 5 new quantitative criteria based on rigorous mathematical principles for evaluating a neural microcircuit C with regard to questions (a) and (b). We will compare in Sections 6 and 8 the predictions of these quantitative measures with the actual computational performance achieved by 102 different types of neural microcircuit models, for a fairly large number of different computational tasks. All microcircuit models that we consider are based on biological data for generic cortical microcircuits (as described in Section 2), but have different settings of their parameters. It should be noted that the models for neural circuits that are discussed in this article are subject to noise (in the form of randomly chosen initial values of membrane voltages, and in the form of biologically realistic models for background noise, see the precise definition in Section 2, and exploration of several noise levels in Section 8). Hence the classical theory for computations in noise-free analog circuits (see, e.g., Siegelmann and Sontag (1994)) cannot be applied to these models. Rather, the more negative results for computations in analog circuits with noise (see, e.g., Maass and Orponen (1998), Maass and Sontag (1999)) apply to the circuit models that are investigated in this article.

For the sake of simplicity, we consider in this article only classification tasks, although other types of computations (e.g. online computations where the target output changes continuously) are at least of equal importance for neural systems. But actually, a theoretical analysis of the capability of neural circuits to approximate a given online computation (that maps continuous input streams onto continuous output streams), see Maass, Natschläger, and Markram (2002) and in more detail Maass and Markram (2004), has shown that the so-called separation property of circuit components is a necessary (and in combination with a condition on the readout also sufficient) condition for being able to approximate a given online computation that maps continuous input streams onto continuous output streams with fading memory. Hence one

can view the computational tasks that are considered in this article also as tests of the separation property of small generic circuits of neurons, and hence of their capability to serve as a rich reservoir of “basis filters” in the context of that theory, and hence as subcircuits for online computing with continuous output streams.

Several results of this article had previously been sketched in Maass, Legenstein, and Bertschinger (2005).

2. Models for generic cortical microcircuits

Our empirical studies were performed on a large variety of models for generic cortical microcircuits. All circuit models consisted of leaky-integrate-and-fire neurons³ and biologically quite realistic models for dynamic synapses.⁴ Neurons (20% of which were randomly chosen to be inhibitory) were located on the grid points of a 3D grid of dimensions $6 \times 6 \times 15$ with edges of unit length. The probability of a synaptic connection from neuron a to neuron b was proportional to $\exp(-D^2(a, b)/\lambda^2)$, where $D(a, b)$ is the Euclidean distance between a and b , and λ is a spatial connectivity constant. Synaptic efficacies w were chosen randomly from distributions that reflect biological data (as in Maass et al. (2002)), with a common scaling factor W_{scale} .

Linear readouts from circuits with $n - 1$ neurons were assumed to compute a weighted sum $\sum_{i=1}^{n-1} w_i x_i(t) + w_0$ (see Section 1). In order to simplify notation we assume that the vector $\mathbf{x}(t)$ contains an additional constant component $x_0(t) = 1$, so that one can write $\mathbf{w} \cdot \mathbf{x}(t)$ instead of $\sum_{i=1}^{n-1} w_i x_i(t) + w_0$. In the case of classification tasks we assume that the readout outputs 1 if $\mathbf{w} \cdot \mathbf{x}(t) \geq 0$, and 0 otherwise.

In order to investigate the influence of synaptic connectivity on computational performance, neural microcircuits were drawn from the distribution of circuits discussed above for 10 different values of λ (which scales the number and average distance of synaptically connected neurons) and 9 different values of W_{scale} (which scales the efficacy of all synaptic connections). 20 microcircuit models C were drawn for each of these 90 different assignments of values to λ and W_{scale} . For each circuit a linear readout was trained to perform one (randomly chosen) out of 2^{80} possible classification tasks on noisy variations u of 80 fixed spike patterns as circuit inputs u . See Fig. 1 for two examples of such spike patterns. The target performance of a linear readout with any such circuit was to output at time $t = 200$ ms the class (0 or 1) of the spike pattern from which the preceding circuit input had been generated (for some arbitrary partition of the 80 fixed spike patterns into two classes). Each spike pattern u consisted of 4 Poisson spike

³ Membrane voltage V_m modeled by $\tau_m \frac{dV_m}{dt} = -(V_m - V_{\text{resting}}) + R_m \cdot (I_{\text{syn}}(t) + I_{\text{background}} + I_{\text{noise}})$, where $\tau_m = 30$ ms is the membrane time constant, I_{syn} models synaptic inputs from other neurons in the circuits, $I_{\text{background}}$ models a constant unspecific background input and I_{noise} models noise in the input. The membrane resistance R_m was chosen as 1 M Ω in all sections except for Section 8.

⁴ Short term synaptic dynamics was modeled according to Markram, Wang, and Tsodyks (1998), with distributions of synaptic parameters U (initial release probability), D (time constant for depression), F (time constant for facilitation) chosen to reflect empirical data (see Maass et al. (2002), for details).

² One can be even more realistic and filter it also by a model for the short term dynamics of the synapse into the readout neuron, but this turns out to make no difference for the analysis proposed in this article.

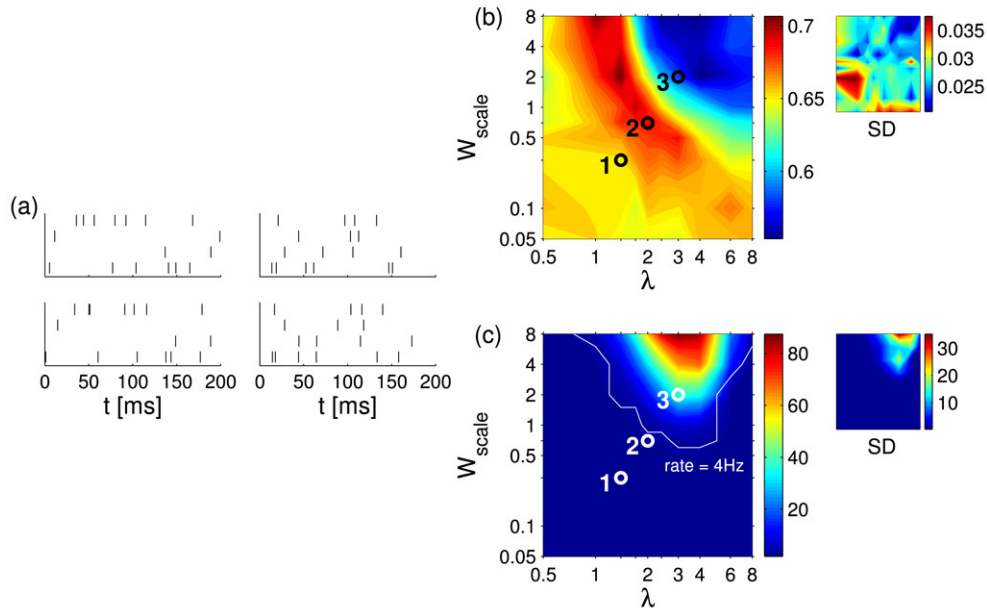


Fig. 1. Performance of different types of neural microcircuit models with linear readouts for classification of spike patterns. (a) In the top row are two examples of the 80 spike patterns that were used as templates for noisy variations (each consisting of 4 Poisson spike trains at 20 Hz over 200 ms), and in the bottom row are examples of noisy variations (Gaussian jitter with SD 10 ms) of these spike patterns which were used as circuit inputs. (b) Fraction of examples (for 500 test examples) for which the output of a linear readout (trained by linear regression with 2000 training examples) agreed with the target classification. Results are shown for 90 different types of neural microcircuits C with λ varying on the x -axis and W_{scale} on the y -axis (20 randomly drawn circuits and 20 target classification functions randomly drawn from the set of 2^{80} possible classification functions were tested for each of the 90 different circuit types, and resulting correctness-rates were averaged). Circles mark three specific choices of λ , W_{scale} -pairs for comparison with other figures, see Fig. 3. The standard deviation of the result is shown in the inset on the upper right. (c) Average firing rate of the neurons in the same circuits and with the same inputs as considered in (b). One sees that the computational performance becomes best at relatively low average firing rates around 4 Hz (see contour line), and that the region of best performance cannot be characterized in terms of the resulting average firing rate.

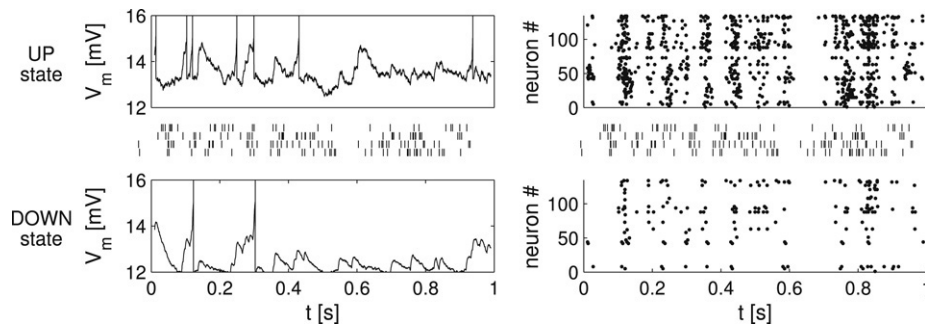


Fig. 2. UP- and DOWN-states in neural microcircuit models. Membrane potential (for a firing threshold of 15 mV) of two randomly selected neurons from circuits in two parameter regimes labeled as UP- and DOWN-states, as well as spike rasters for the same two parameter regimes (with the actual circuit inputs shown between the two rows).

trains over 200 ms. Details on this classification task are given in Appendix A.1. Performance results are shown in Fig. 1(b) for 90 different types of neural microcircuit models together with a linear readout. Fig. 1(c) shows for comparison the resulting average firing rates for the same circuits and inputs, demonstrating that it would be impossible to characterize the regions of best computational performance in terms of the resulting firing rates. We will refer to this setup in Sections 3–6.

We will investigate in Section 8 a completely different parametrization of regimes of neural circuits that allows us to switch their dynamics between UP- and DOWN-states. These dynamic regimes, which have been identified in numerous

intracellular recordings in vivo, differ with regard to the mean and variance of the membrane potential and the membrane conductance of the neurons in the circuit. The UP-state is assumed to result from a massive bombardment by background synaptic inputs in the awake state of the organism. We have simulated these different dynamic regimes in circuits of size $3 \times 3 \times 15$ by varying the membrane resistance R_m , the background current $I_{background}$, and the noise current I_{noise} in parallel according to Destexhe, Rudolph, and Pare (2003). Fig. 2 shows that one can simulate in this way different dynamic regimes of the same circuit where the time course of the membrane potential qualitatively matches

data from intracellular recordings in UP- and DOWN-states (see, e.g., Anderson, Lampl, Reichova, Carandini, and Ferster (2000), Destexhe et al. (2003), Shu, Hasenstaub, Badoual, Bal, and McCormick (2003)). We will discuss in Section 8 the prediction and the actual computational performance of circuits with linear readouts in 13 different regimes of the parameters R_m , $I_{\text{background}}$, I_{noise} .

We would like to point out that in this set-up the performance of the circuit with the best parameter setting is still far from optimal, especially for the task considered in Fig. 8(f). The reason is that the size of the circuits that we consider in this article is relatively small in comparison with the complexity of the computational tasks. Empirical tests (see Fig. 10 in Maass, Natschläger, and Markram (2004) and Fig. 6 in Häusler and Maass (2007)) have shown that the computational performance of models for generic cortical microcircuits tends to increase with circuit size for computational tasks of this type. However since our focus is in this article on the gradient of computational performance, rather than on its absolute level, the investigation of relatively small circuits appears to be particularly instructive.

3. The edge of chaos in neural microcircuit models

A recurrent neural circuit is a special case of a dynamical system. Related dynamical systems have been studied extensively in various contexts in physics, e.g. cellular automata (Langton, 1990; Packard, 1988), random Boolean networks (Kauffman, 1993), and Ising-spin models (networks of threshold elements) (Derrida, 1987). By changing some global parameters of the system, e.g. connectivity structure or the functional dependence of the output of an element on the output of other elements, one can change the dynamics of the system from ordered to chaotic. The cited studies suggest that dynamical systems achieve optimal computational capabilities if their dynamics lies on the transition boundary between order and chaos. This transition boundary was termed the “edge of chaos”. Later, it was argued that the region of best computational performance should depend on the task at hand. Therefore, best computational power does not necessarily correspond to the edge of chaos (Mitchell, Hraber, & Crutchfield, 1993). We will show how to include some task-specificity in the analysis of neural circuits in Sections 4 and 5. Almost all preceding work on computations in dynamical systems focused on the autonomous case, i.e. on the evolution from some initial state without any external influences (except for the initial state of the system). From the computational point of view, such autonomous systems represent computations on batch input, comparable to the batch processing by Turing machines. However, computations in cortex have a strong online flavor, i.e. inputs are time series and constantly impinging onto the circuit. Networks of threshold elements with online input were recently considered in Bertschinger and Natschläger (2004). In this study, a network with online input is called chaotic if arbitrary small differences in a (initial) network state $\mathbf{x}(0)$ are highly amplified and do not vanish. In contrast, an ordered network forgets the initial network state $\mathbf{x}(0)$ very quickly and the current network state $\mathbf{x}(t)$ is determined largely

by the current input. The results of this study also suggest that for online computing also superior computational performance can be found in systems with dynamics located at the edge of chaos. We refer to Legenstein and Maass (2007) for a review of this preceding work about the relationship between the edge of chaos and the computational performance of various types of dynamical systems.

An analysis of the temporal evolution of state differences that result from fairly large input differences cannot identify those parameter values in the map of Fig. 1(b) that yield circuits which have (in conjunction with a linear readout) large computational performance (Maass et al., 2005). The reason is that large initial state differences (as they are typically caused by different spike input patterns) tend to yield for most values of the circuit parameters nonzero state differences not only while the online spike inputs are different, but also long afterwards when the online inputs agree during subsequent seconds (even if the random internal noise is identical in both trials). But if one applies the definition of the edge of chaos via Lyapunov exponents (see Kantz and Schreiber (1997)), that calls for the analysis of state differences that result from infinitesimal (or at least very small) input differences, the resulting edge of chaos lies for the previously introduced type of computations (classification of noisy spike templates by a trained linear readout) in the region of the best computational performance (see the map in Fig. 1(b), which is repeated for easier comparison in Fig. 3(d)). According to this definition one looks for the exponent $\mu \in \mathbb{R}$ which provides through the formula

$$\delta_{\Delta T} \approx \delta_0 \cdot e^{\mu \Delta T}$$

the best estimate of the state separation $\delta_{\Delta T}$ at time ΔT after the computation was started in two trials with an “infinitesimal” initial state difference δ_0 . We generalize this analysis to the case with online input by choosing exactly the same online input (and the same random noise) during the intervening time interval of length ΔT for the two trials which start with a small initial state difference δ_0 . We then average the resulting state differences $\delta_{\Delta T}$ over many random choices of such online inputs (and internal noise). As in the classical case with offline input it turns out to be essential to apply this estimate for $\delta_0 \rightarrow 0$, since $\delta_{\Delta T}$ tends to saturate when ΔT grows for each fixed value of δ_0 . The saturation occurs because the distance between any two circuit states is bounded in practice. This can be seen in Fig. 3(a), which shows outcomes of this experiment for a δ_0 that results from moving a single spike that occurs in the online input at time $t = 1$ s by 0.5 ms. This experiment was repeated for 3 different circuits with parameters chosen from the 3 locations marked on the map in Fig. 3(c). By determining the best fitting μ for $\Delta T = 1.5$ s for 3 different values of δ_0 (resulting from moving a spike at time $t = 1$ s by 0.5, 1, 2 ms) one gets the dependence of this Lyapunov exponent on the circuit parameter λ shown in Fig. 3(b) (for values of λ and W_{scale} on a straight line between the points marked in the map of Fig. 3(c)). The middle curve in Fig. 3(c) shows for which values of λ and W_{scale} the Lyapunov exponent is estimated to have the value 0. This estimate is quite robust with respect to

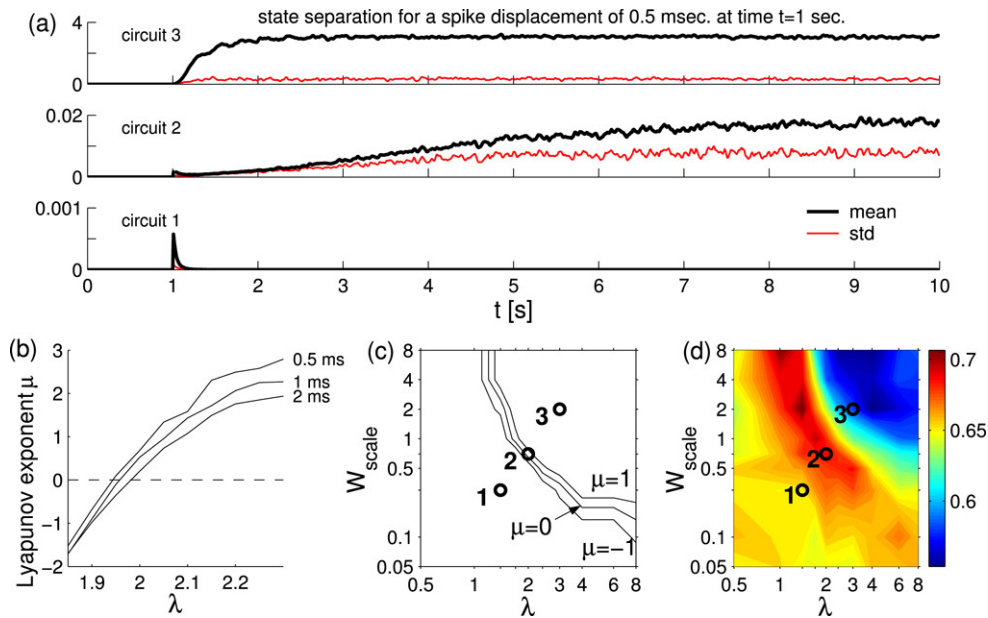


Fig. 3. Analysis of changes in the dynamics resulting from small input differences for different types of neural microcircuit models as specified in Section 2. Each circuit C was tested for two arrays u and v of 4 input spike trains at 20 Hz over 10 s that differed only in the timing of a single spike at time $t = 1$ s. (a) A spike at time $t = 1$ s was delayed by 0.5 ms. Temporal evolution of Euclidean differences between resulting circuit states $\mathbf{x}_u(t)$ and $\mathbf{x}_v(t)$ with 3 different values of λ , W_{scale} according to the 3 points marked in panel (c). For each parameter pair, the average state difference (thick line) and standard deviation (thin line) of 40 randomly drawn circuits is plotted. (b) Lyapunov exponents μ along a straight line between the points marked in panel (c) with different delays of the delayed spike. The delay is denoted on the right of each line. The exponents were determined for the average state difference of 40 randomly drawn circuits. (c) Lyapunov exponents μ for 90 different types of neural microcircuits C with λ varying on the x -axis and W_{scale} on the y -axis (the exponents were determined for the average state difference of 20 randomly drawn circuits for each parameter pair). A spike in u at time $t = 1$ s was delayed by 0.5 ms. The contour lines indicate where μ crosses the values -1 , 0 , and 1 . (d) Computational performance of a linear readout with these circuits (same as Fig. 1(b)), shown for comparison with panel (c).

the choice of δ_0 and ΔT .⁵ By comparing the middle curve in Fig. 3(c) with those regions on this parameter map where the circuits have the largest computational performance (for the classification of noisy spike patterns, see Fig. 3(d)), one sees that this line runs through those regions which yield the largest computational performance for these computations. We refer to Mayor and Gerstner (2005) for other recent work on the relationship between the edge of chaos and the computational performance of spiking neural circuit models.

Although this estimated edge of chaos coincides quite well with points of best computational performance, it remains an unsatisfactory tool for predicting parameter regions with large computational performance for three reasons:

- (i) Since the edge of chaos is a lower-dimensional manifold in a parameter map (in this case a curve in a 2D map), it cannot predict the (full-dimensional) extent of regions within a parameter map that yield high computational performance (e.g. the regions with light shading in Fig. 1(b)).
- (ii) The edge of chaos does not provide intrinsic reasons *why* points of the parameter map yield small or large computational performance.

- (iii) It turns out that in some parameter maps *different* regions provide circuits with large computational performance for linear readouts for *different* classes of computational tasks (as shown in Section 8 for computations on spike patterns and for computations with firing rates). But the edge of chaos can at best single out peaks for *one* of these regions. Hence it cannot possibly be used as a universal predictor of maximal computational performance for different types of computational tasks.

These three deficiencies suggest that more sophisticated tools are needed in order to solve the question of what makes a neural microcircuit model computationally superior.

4. A measure for the kernel-quality

One expects from a powerful computational system that significantly different input streams cause significantly different internal states, and hence may lead to different outputs. Most real-world computational tasks require that a readout gives a desired output not just for 2, but for a fairly large number m of significantly different inputs. One could of course test whether a readout on a circuit C can separate each of the $\binom{m}{2}$ pairs of such inputs. But even if the readout can do this, we do not know whether a neural readout from such circuit would be able to produce given target outputs for these m inputs.

Therefore we propose here the *linear separation property* as a more suitable quantitative measure for evaluating the computational performance of a neural microcircuit (or more

⁵ The zero crossing of the Lyapunov exponent is at $\lambda = 1.94$, 1.93 , and 1.91 for $\Delta T = 1$, 1.5 , and 2 s respectively ($\delta_0 = 0.5$ ms). We used relatively large ΔT well-suited for circuits at the edge of chaos in our setup. For the chaotic circuit “circuit 3” in Fig. 3(a), a smaller ΔT would be more appropriate in order to avoid saturation of the state separation. But since we are mainly interested in the edge of chaos, we opted for $\Delta T = 1.5$ s.

precisely: the kernel-quality of a circuit; see below). To evaluate the linear separation property of a circuit C for m different inputs u_1, \dots, u_m (which are in this article always functions of time, i.e. input streams such as for example multiple spike trains) we compute the rank of the $n \times m$ matrix M whose columns are the circuit states $\mathbf{x}_{u_i}(t_0)$ resulting at some fixed time t_0 for the preceding input stream u_i . If this matrix has rank m , then it is *guaranteed* that *any* given assignment of target outputs $y_i \in \mathbb{R}$ at time t_0 for the inputs u_i can be implemented by this circuit C (in combination with a linear readout). In particular, each of the 2^m possible binary classifications of these m inputs can then be carried out by a *linear* readout from this fixed circuit C . Obviously such insight is much more informative than a demonstration that some *particular* classification task can be carried out by a readout neuron on such circuit C . If the rank of this matrix M has a value $r < m$, then this value r can still be viewed as a measure for the computational performance of this circuit C , since r is the number of “degrees of freedom” that a linear readout has in assigning target outputs y_i to these inputs u_i (in a way which can be made mathematically precise with concepts of linear algebra). Note that this rank-measure for the linear separation property of a circuit C may be viewed as an empirical measure for its *kernel-quality*, i.e. for the complexity and diversity of nonlinear operations carried out by C on its input stream in order to boost the classification performance of a subsequent *linear* decision-hyperplane (see Vapnik (1998)).

5. A measure for the generalization capability

Obviously the preceding measure addresses only one component of the computational performance of a neural circuit C with linear readout. Another component is its capability to *generalize* a learned computational function to *new* inputs. Mathematical criteria for generalization capability are derived in Vapnik (1998) (see Ch. 4 of Cherkassky and Mulier (1998) for a compact account of results relevant for our arguments). According to this mathematical theory one can quantify the generalization capability of any learning device in terms of the VC-dimension of the class \mathcal{H} of hypotheses that are potentially used by that learning device.⁶ More precisely: if VC-dimension (\mathcal{H}) is substantially smaller than the size of the training set S_{train} , one can prove that this learning device generalizes well, in the sense that the hypothesis (or input–output map) produced by this learning device is likely to have for new examples an error rate which is not much higher than its error rate on S_{train} , provided that the new examples are drawn from the same distribution as the training examples (see Eq. (4.22) in Cherkassky and Mulier (1998)).

We apply this mathematical framework to the class \mathcal{H}_C of all maps from a set S_{univ} of inputs u into $\{0, 1\}$ which can be

implemented by a circuit C with linear readout. More precisely: \mathcal{H}_C consists of all maps from S_{univ} into $\{0, 1\}$ that a linear readout from circuit C with fixed internal parameters (weights etc.) but arbitrary weights $\mathbf{w} \in \mathbb{R}^n$ of the readout (that classifies the circuit input u as belonging to class 1 if $\mathbf{w} \cdot \mathbf{x}_u(t_0) \geq 0$, and to class 0 if $\mathbf{w} \cdot \mathbf{x}_u(t_0) < 0$) could possibly implement.

Whereas it is very difficult to achieve tight theoretical bounds for the VC-dimension of even much simpler neural circuits, see Bartlett and Maass (2003), one can efficiently estimate the VC-dimension of the class \mathcal{H}_C that arises in our context for some finite ensemble S_{univ} of inputs (that contains all examples used for training or testing) by using the following mathematical result (which can be proved with the help of Radon’s Theorem).

Theorem 5.1. *Let r be the rank of the $n \times s$ matrix consisting of the s vectors $\mathbf{x}_u(t_0)$ for all inputs u in S_{univ} (we assume that S_{univ} is finite and contains s inputs). Then $r \leq \text{VC-dimension}(\mathcal{H}_C) \leq r + 1$.*

Proof. Fix some inputs u_1, \dots, u_r in S_{univ} so that the resulting r circuit states $\mathbf{x}_{u_i}(t_0)$ are linearly independent. The first inequality is obvious since this set of r linearly independent vectors can be shattered by linear readouts from the circuit C . To prove the second inequality assume the contradiction that there exists a set v_1, \dots, v_{r+2} of $r + 2$ inputs in S_{univ} so that the corresponding set of $r + 2$ circuit states $\mathbf{x}_{v_i}(t_0)$ can be shattered by linear readouts. This set M of $r + 2$ vectors is contained in the r -dimensional space spanned by the linearly independent vectors $\mathbf{x}_{u_1}(t_0), \dots, \mathbf{x}_{u_r}(t_0)$. Radon’s Theorem implies that M can be partitioned into disjoint subsets M_1, M_2 whose convex hulls intersect. Since these sets M_1, M_2 cannot be separated by a hyperplane, it is clear that no linear readout exists that assigns value 1 to points in M_1 and value 0 to points in M_2 . Hence $M = M_1 \cup M_2$ is not shattered by linear readouts, which is a contradiction to our assumption. ■

We propose to use the rank r defined in Theorem 5.1 as an estimate of VC-dimension (\mathcal{H}_C), and hence as a measure that informs us about the generalization capability of a neural microcircuit C (for arbitrary probability distributions over the set S_{univ}). It is assumed here that the set S_{univ} contains many noisy variations of the same input signal, since otherwise learning with a randomly drawn training set $S_{\text{train}} \subseteq S_{\text{univ}}$ has no chance to generalize to new noisy variations. This use of noisy variations stands in contrast to the estimate of the kernel-quality where we tested how well the circuit can distinguish different input streams. Hence, the rank r in Theorem 5.1 is in general different from the rank which we used to estimate the kernel-quality because of the different input distributions. Note that each family of computational tasks induces a particular notion of what aspects of the input are viewed as noise, and what input features are viewed as signals that carry information which is relevant for the target output for at least one of these computational tasks. For example for computations on spike patterns some small jitter in the spike timing is viewed as noise. For computations on firing rates even the sequence of interspike intervals and temporal relations between spikes that

⁶ The VC-dimension (of a class \mathcal{H} of maps H from some universe S_{univ} of inputs into $\{0, 1\}$) is defined as the size of the largest subset $S \subseteq S_{\text{univ}}$ which can be *shattered* by \mathcal{H} . One says that $S \subseteq S_{\text{univ}}$ is shattered by \mathcal{H} if for every map $f : S \rightarrow \{0, 1\}$ there exists a map H in \mathcal{H} such that $H(u) = f(u)$ for all $u \in S$ (this means that every possible binary classification of the inputs $u \in S$ can be carried out by some hypothesis H in \mathcal{H}).

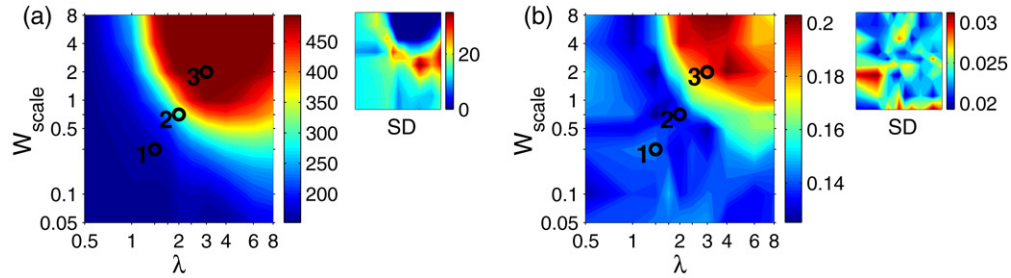


Fig. 4. VC-dimension as predictor for the generalization capability of neural microcircuit models. (a) Prediction of the generalization capability for 90 different types of neural microcircuits (as in Fig. 1(b)): estimated VC-dimension (of the hypothesis class \mathcal{H}_c for a set S_{univ} of inputs consisting of 500 jittered versions of 4 spike patterns), each of the 90 data points for 90 different circuit types was computed as the average over 20 circuits; for each circuit, the average over 5 different sets of spike patterns was used. The standard deviation is shown in the inset on the upper right. See Section 6 for details. (b) Actual generalization capability ((error on test set) – (error on training set)) of the same neural microcircuit models for a particular learning task, quantified by the difference of test error and training error (error defined as the fraction of examples that are misclassified) in the spike pattern classification task discussed in Section 2. The standard deviation is shown in the inset on the upper right.

arrive from different input sources are viewed as noise, as long as these input spike trains represent the same firing rates. Consequently different sets S_{univ} become relevant for estimates of the generalization capability of circuits for these two types of computational tasks. Hence, in contrast to the edge of chaos analysis, the proposed measures are dependent on the actual family of tasks considered.

An example for the former computational task was discussed in Section 2. This task was to output at time $t = 200$ ms the class (0 or 1) of the spike pattern from which the preceding circuit input had been generated (for some arbitrary partition of the 80 fixed spike patterns into two classes, see Section 2). For a poorly generalizing network, the difference between train and test error is large. One would suppose that this difference becomes large as the network dynamics become more chaotic. This is indeed the case, see Fig. 4(b). The transition is well-predicted by the estimated VC-dimension of \mathcal{H}_c , see Fig. 4(a). An example for a computational task on firing rates is given in Section 8.

On first sight it seems difficult to estimate the kernel-quality and the generalization capability with the rank of state matrices in practice if there is noise in the circuit. One particular concern is that with noisy neurons, the state matrix will almost always have full rank because of noisy state vectors. It turns out that at least in our models (where a noise current is injected into the neurons), this concern is not justified. Our simulations show that the state matrix does not have high rank in the ordered dynamical regime, even in the case of strong noise currents, see Section 8. The reason is that for a given input, a relatively large percentage of neurons is not activated at all. The noise has only an influence on the state value of activated neurons, the state value of neurons which are not activated remains zero as long as the noise is not strong enough to activate them (an advantage of spiking neurons over analog neurons). For a more detailed analysis of this issue, see Section 6.

6. Evaluating the influence of synaptic connectivity on computational performance

We now test the predictive quality of the two proposed measures for the computational performance of a microcircuit

with linear readout on spike patterns. One should keep in mind that the proposed measures do not attempt to test the computational capability of a circuit with linear readout for one particular computational task, but for *any* distribution on S_{univ} and for a very large (in general infinitely large) family of computational tasks that only have in common a particular bias regarding which aspects of the incoming spike trains may carry information that is relevant for the target output of computations, and which aspects should be viewed as noise. The details on how we evaluated the measures are given in Appendix A.2. Fig. 5(a) explains *why* the lower left part of the parameter map in Fig. 1(b) is less suitable for any such computation, since there the kernel-quality of the circuits is too low. Fig. 5(b) explains *why* the upper right part of the parameter map in Fig. 1(b) is less suitable, since a higher VC-dimension (for a training set of fixed size) entails poorer generalization capability. We are not aware of a theoretically founded way of combining both measures into a single value that predicts overall computational performance. But if one just takes the difference of both measures then the resulting number (see Fig. 5(c)) predicts quite well which types of neural microcircuit models together with a linear readout perform well for the particular computational tasks considered in Fig. 1(b). Note that the difference of these two measures does not just reconfirm the well-known fact from statistical learning theory that neither a too small nor a too large hypothesis class is adequate. It also produces quite good predictions concerning the question *where* exactly in the fairly large range in between the best computational performance of neural microcircuit models can be expected.

The rather small rank of the state matrix, especially in the ordered regime, can be explained in part by the small number of neurons which get activated (i.e., emit at least one spike) for a given input pattern. For some input pattern u , let the activation vector $\mathbf{x}_u^{act} \in \{0, 1\}^n$ be the vector with the i th entry being 1 if neuron i was activated during the presentation of this pattern (i.e., up to t_0) and 0 otherwise (where n is the total number of neurons in the circuit). The number of neurons activated for a given input u is then the number of ones in \mathbf{x}_u^{act} . Fig. 6(a) shows that in the ordered regime, approximately 100 of the 540 neurons get activated

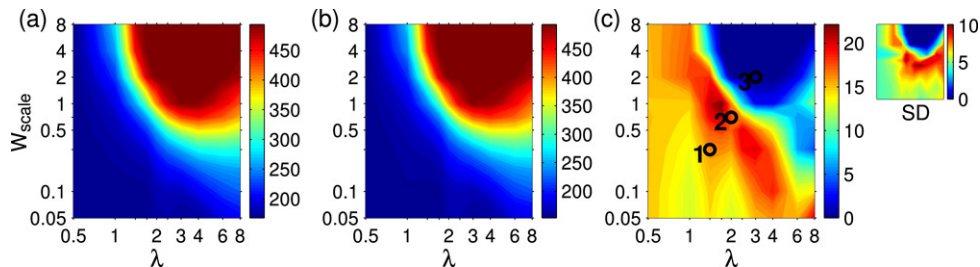


Fig. 5. Values of the proposed measures for computations on spike patterns. (a) Kernel-quality for spike patterns of 90 different circuit types (average over 20 circuits, mean $SD = 13$).⁷ (b) Generalization capability for spike patterns: estimated VC-dimension of \mathcal{H}_C (for a set $\mathcal{S}_{\text{univ}}$ of inputs u consisting of 500 jittered versions of 4 spike patterns), for 90 different circuit types (same as Fig. 4(a)). (c) Difference of both measures (the standard deviation is shown in the inset on the upper right). This should be compared with actual computational performance plotted in Fig. 1(b).

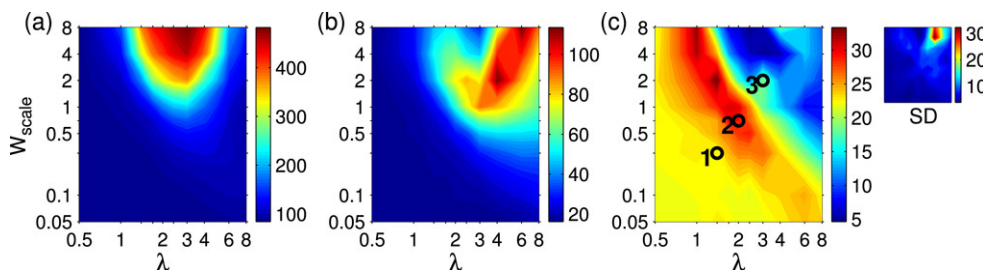


Fig. 6. Analysis of the sets of neurons which get activated. (a) For a given input pattern, only a small fraction of the 540 circuit neurons are activated (i.e., emit at least one spike). Shown is the mean number of neurons which are activated by an input pattern. For each parameter setting, the mean over 20 circuits and 125 patterns per circuit is shown. The input patterns were drawn from the same distribution as the templates for the task in Fig. 1. (b) For different input patterns, sets of activated neurons are diverse. Shown is the mean Hamming distance between two of the 125 activation vectors arising from a set of circuit inputs consisting of 125 jittered versions of a spike template. For each parameter setting, the mean over 20 circuits is shown. (c) Difference between the mean Hamming distance for an input set consisting of 125 randomly drawn spike templates and the mean Hamming distance for an input set consisting of 125 jittered versions of a spike template. This difference predicts computational performance (see Fig. 1(b)) quite well. For each parameter setting, the mean over 20 circuits is shown. The standard deviation is shown in the inset on the upper right.

for a typical spike template. Note that different subsets of circuit neurons get activated for different input templates, since the rank of the state matrix in this regime is approximately 200, see Fig. 5(a). We tested to what extent the diversity of activation vectors determines our proposed measures. To measure the diversity of activation vectors over a set of m inputs u_1, \dots, u_m , we calculated the mean Hamming distance between two of the activation vectors $\mathbf{x}_{u_1}^{\text{act}}, \dots, \mathbf{x}_{u_m}^{\text{act}}$ (i.e., we calculated the mean of $|\mathbf{x}_{u_i}^{\text{act}} - \mathbf{x}_{u_j}^{\text{act}}|_{L_1}$ for $i, j = 1, \dots, m, i < j$, where $|\cdot|_{L_1}$ denotes the L_1 norm). Fig. 6(b) shows that the mean Hamming distance between activation vectors for a set of input patterns consisting of $m = 125$ jittered versions of a single randomly drawn spike template only partially predicts the generalization capability of the circuit shown in Fig. 4(a) and the corresponding prediction by the estimated VC-dimension shown in Fig. 4(b). The same is true if one compares the kernel-quality of the circuit models with the mean Hamming distance of activation vectors for 125 randomly drawn spike templates (data not shown). The difference of the Hamming distances for these two input sets heuristically informs us about how strongly the signal (i.e., different spike templates) is represented in the spatial circuit code (i.e., which neurons are activated) compared to how strongly the noise (i.e., different jittered versions of a single template) is represented in the spatial circuit code. It turns out that this difference fits quite well to the outcome

of our theoretically derived measures, compare Fig. 6(c) with Fig. 5(c).

Since noise in the circuit (e.g., a noise current injected into the circuit neurons) has only a minor effect on the set of activated neurons, this explains why the rank measures are not strongly influenced by noise.

7. Predicting computational performance on the basis of circuit states with limited precision

In the earlier simulations, the readout unit was assumed to have access to the actual analog circuit states (which are given by the low-pass filtered output spikes of the circuit neurons). In a biological neural system however, readout elements may have access only to circuit states of limited precision since signals are corrupted by noise. Therefore, we repeated our analysis for the case where each circuit state is only given with some fixed finite precision. More precisely, the range of values of the state vector components for each circuit was first normalized to the range $[0, 1]$. Then, each circuit state $x_i(t)$ at time t was mapped onto the closest value in $\{0, 0.1, 0.2, \dots, 1\}$.

⁷ The rank of the matrix consisting of 500 circuit states $\mathbf{x}_u(t)$ for $t = 200$ ms was computed for 500 spike patterns over 200 ms as described in Section 4, see Fig. 1(a). For each circuit, the average over 5 different sets of spike patterns was used.

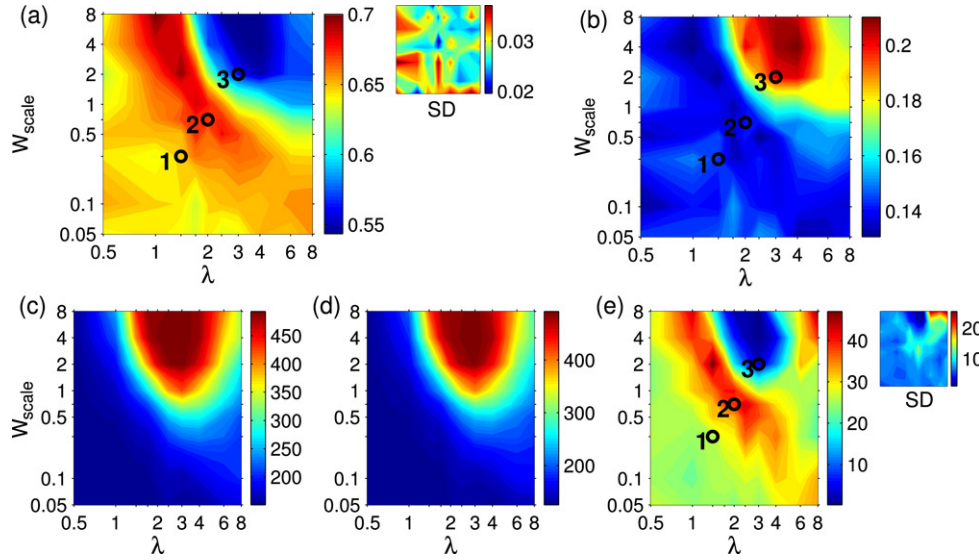


Fig. 7. Performance and values of the proposed measures for different types of neural microcircuit models with linear readouts for classification of spike patterns on the basis of circuit states with limited precision. In each case the actual analog value of each component of the circuit state was rounded to one of 11 possible values (see text). (a) Fraction of examples (for 500 test examples) for which the output of the readout agreed with the target classification (the readout was trained by linear regression with 2000 training examples; see Fig. 1(b) for more details). Note the similarity to Fig. 1(b). The standard deviation of the result is shown in the inset on the upper right. (b) Generalization capability of the same neural microcircuit models for a particular learning task, quantified by the difference of test error and training error (error defined as the fraction of examples that are misclassified). Compare with Fig. 4(b). (c) Kernel-quality for spike patterns of 90 different circuit types. (d) Generalization capability for spike patterns. (e) Difference of both measures (the standard deviation is shown in the inset on the upper right). Compare with Fig. 5(c).

The performance of such readouts on the spike pattern classification task described in Section 2 is shown in Fig. 7(a). As compared to Fig. 1(b), there is a slight increase in performance in the upper right corner of the performance landscape which corresponds to circuits of high activity. This increase is mostly caused by an increase in the generalization capability of circuits in this regime as shown in Fig. 7(b) (compare to Fig. 5(b)).

The proposed measures indeed predict an increase of the generalization capabilities of highly active circuits (see Fig. 7(d)) which is only partly compensated by a decrease in the kernel-quality of these circuits (see Fig. 7(c)). The difference of both measures emphasizes this, however it overestimates the influence of increased generalization capability on computational performance for such circuits with linear readouts (see Fig. 7(e)).

Altogether we conclude that both the actual computational performance of circuits with linear readouts and the proposed predictors are slightly changed when one moves from infinite to finite precision of circuit states, but that the proposed predictors of computational performance do quite well in either case.

8. Evaluating the computational performance of neural microcircuit models in UP- and DOWN-states

Data from numerous intracellular recordings suggest that neural circuits in vivo switch between two different dynamic regimes that are commonly referred to as UP- and DOWN-states. UP-states are characterized by a bombardment with synaptic inputs from recurrent activity in the circuit, resulting in a membrane potential whose average value is significantly closer to the firing threshold, but also has larger variance.

Furthermore, synaptic bombardment in UP-states leads to an increase in membrane conductance. We have simulated these different dynamic regimes in circuits of size $3 \times 3 \times 15$ by varying the membrane resistance R_m , the background current $I_{\text{background}}$, and the noise current I_{noise} in parallel (see Section 2).

We have tested the computational performance of circuits in 13 different dynamic regimes. The membrane resistance R_m was varied between 3.3 and 2.2 M Ω (note that the DOWN-state is characterized by a larger membrane resistance). The ratio of these resistances is in accordance with empirical data, see Shu et al. (2003).⁸ For each value of R_m , a corresponding background and noise current was set. The background current was set such that $I_{\text{background}} \cdot R_m$ (the steady state membrane voltage produced by this current) varied between 11.5 (for $R_m = 3.3$ M Ω) and 14.5 mV (for $R_m = 2.2$ M Ω). The noise current was set such that $I_{\text{noise}} \cdot R_m$ varied between 0.6 (for $R_m = 3.3$ M Ω) and 6.6 mV (for $R_m = 2.2$ M Ω).

Two conceptually different computational tasks were performed on these parameter sets and performance was predicted by the rank measure. For the first task, inputs consisted of 200 ms low frequency (5 Hz) spike trains in 30 input channels. The kernel-quality of a circuit was estimated by measuring the rank of the state matrix at $t = 200$ ms for 135 independently drawn spike patterns. The VC-dimension was estimated for a set S_{univ} of inputs consisting of 135

⁸ In Shu et al. (2003), the input resistance of cortical pyramidal neurons in vitro was 33 M Ω in the mean. In the UP-state, the membrane conductance increased by 12.8 nS in mean, leading to a membrane resistance of roughly 23 M Ω .

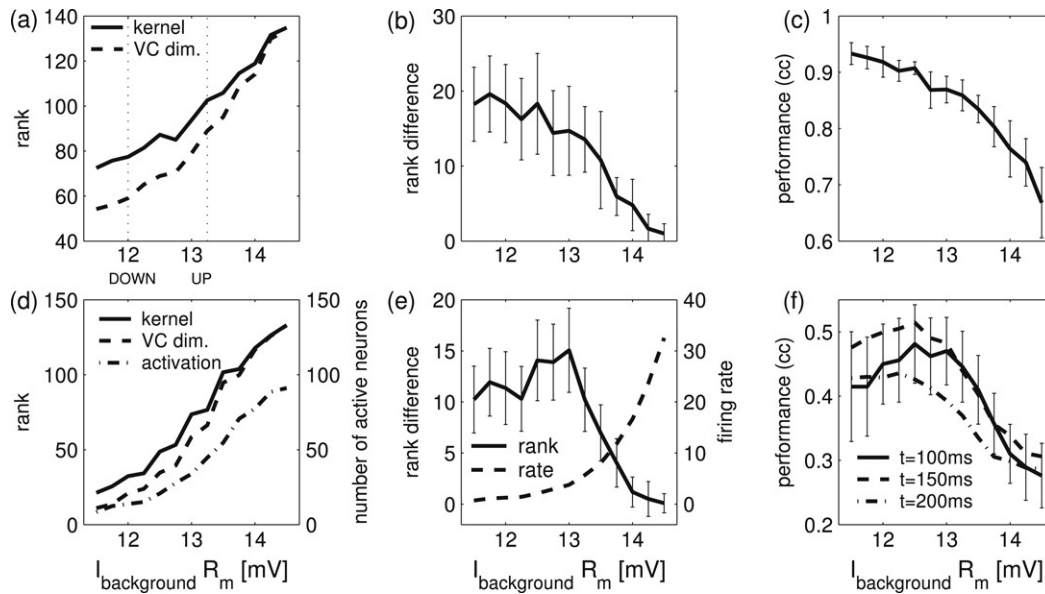


Fig. 8. Analysis of the computational performance of simulated neural microcircuits (together with linear readouts) in different dynamic regimes. (a) Estimates of the kernel-quality for 135 spike patterns (solid line; average over 20 circuits). Estimate of the VC-dimension for a set S_{univ} of inputs consisting of 135 jittered versions of one spike pattern (dotted line; average over 20 circuits). (b) Difference of measures from panel (a) (as in Fig. 5(c); mean \pm SD). (c) Evaluation of computational performance (correlation coefficient; for 500 test examples) of a linear readout (trained by linear regression with 1000 training examples). 20 randomly drawn circuits and 20 target classification functions randomly drawn from the set of 2^{30} possible classification functions were tested for each of the 13 different circuit types, and resulting correlation coefficients were averaged. (d) Estimates of the kernel-quality for input streams u with 13^2 different combinations of 13 firing rates (solid line; average over 20 circuits). Estimate of the VC-dimension for a set S_{univ} of inputs consisting of 13^2 different spike trains u that represent one combination of firing rates. (dotted line; average over 20 circuits). Also shown is the average number of neurons that get activated (i.e. fire at all) for a typical input (dash-dotted line; scale on right hand side). (e) Difference of measures from panel (d) (solid line). Average firing rate in the same circuits for the same inputs (dashed line). Note that the firing rates are poorly correlated both with the difference of the two measures, and with the computational performance shown in (f). (f) Evaluation of computational performance (correlation coefficient; for 400 test examples) of a linear readout (trained by linear regression with 2000 training examples). 20 randomly drawn circuits were tested for each of the 13 different circuit types, and resulting correlation coefficients were averaged. Results are shown for classification at three different time points after stimulus onset: 100 ms (solid line), 150 ms (dashed line), and 200 ms (dash-dotted line). This shows that at least for this task performances for these time points were comparable. Their performance peak (as a function of the amount of background noise) was also predicted quite well by our two measures as shown in panel (e).

jittered versions of one spike pattern. In order to eliminate effects of particular spike patterns, the average rank over 20 different spike patterns was evaluated for each circuit. Results are shown in Fig. 8(a), (b). For each circuit, a linear readout was trained to perform one (randomly chosen) out of 2^{30} possible classification tasks on noisy variations u of 30 fixed spike patterns as circuit inputs u . The target performance was to output at time $t = 200$ ms the class (0 or 1) of the spike pattern from which the preceding circuit input had been generated (for some arbitrary partition of the 30 fixed spike patterns into two classes). Performance results are shown in Fig. 8(c). Details on this task are given in Appendix A.3.

The second task was a classification task on firing rates. Two rates r_1 and r_2 were drawn independently from the set $\{20, 23, \dots, 56\}$ Hz. Inputs u consisted of four Poisson spike trains of length 100 ms, two with rate r_1 and two with rate r_2 . The kernel-quality of a circuit was estimated by measuring the rank of the state matrix at $t = 100$ ms for the 13^2 different combinations of firing rates. The VC-dimension was estimated for a set S_{univ} of inputs consisting of 13^2 different spike trains u that represent one combination of firing rates. In order to eliminate effects of particular rate combinations, the average rank over 20 different rate pairs was evaluated for each circuit. Results are shown in Fig. 8(d), (e). The rather low number of activated neurons (see Fig. 8(d), dash-dotted line) explains

why both the kernel-measure and the VC-dimension assume values well below the total number of neurons in the circuit (135) despite the presence of strong noise current. Note that different sets of neurons get activated for different inputs, hence the rank of the state matrix is larger than the average number of activated neurons (see also the discussion on this issue in Section 6). We then evaluated the computational performance of neural microcircuit models on this rate task. Out of the 13^2 possible combinations of firing rates, 10 were chosen randomly. These ten pairs of firing rates were randomly partitioned into two classes. The task was to learn this arbitrary classification of pairs of chosen firing rates (details on the task are given in Appendix A.3). This task was nontrivial insofar as the classification did not just rely on thresholds for firing rates. Performance results are shown in Fig. 8(f).

The results suggest that linear readouts together with circuits at the left end of this parameter map (corresponding to DOWN-states) are predicted to have better computational performance for computations on sparse input. This agrees quite well with direct evaluations of computational performance. Hence the proposed quantitative measures may provide a theoretical foundation for understanding the computational function of different states of neural activity.

An evaluation of the performance of the same predictors for computational performance for a different two-dimensional

parametrization of circuits between UP- and DOWN-states can be found in [Maass et al. \(2005\)](#).

9. Discussion

We have proposed a new method for understanding why one neural microcircuit C is performing better than another neural microcircuit C' (for some large family of computational tasks that only have to agree with regard to those features of the circuit input, e.g. rates or spike patterns, on which the target outputs may depend). More precisely, we have introduced two measures (a kernel-measure for the nonlinear processing capability, and a measure for the generalization capability) whose sum predicts quite well those dynamical ranges where cortical microcircuit models have the largest computational capability. This method can also be used to understand the computational function of further biological details that may be added in future microcircuit models. But this method is in principle applicable not just to circuit models, but also to neural microcircuits in vivo and in vitro. Here it can be used to analyze (for example by optical imaging) for which family of computational tasks a particular microcircuit in a particular dynamic regime is optimal, or at least well-suited.

The main assumption of the method is that (approximately) linear readouts from neural microcircuits have the task to produce the actual outputs of specific computations. Also for the classification of spike patterns, a particular time was assumed at which the readout was supposed to extract the result. It is not clear how such timing problems are solved in the brain. On the other hand, we have shown in [Fig. 8\(f\)](#) that the performance of the readout does not depend very much on the chosen time point.

We are not aware of specific theoretically founded rules for choosing the sizes of the ensembles of inputs for which the kernel-measure and the VC-dimension are to be estimated. Obviously both have to be chosen sufficiently large so that they produce a significant gradient over the parameter map under consideration (taking into account that their maximal possible value is bounded by the circuit size). To achieve theoretical guarantees for the performance of the proposed predictor of the generalization capability of a neural microcircuit one should apply it to a relatively large ensemble S_{univ} of circuit inputs (and the dimension n of circuit states should be even larger). But the computer simulations of 102 types of neural microcircuit models that were discussed in this article suggest that practically quite good predictions can already be achieved for a much smaller ensemble of circuit inputs.

The paradigms laid out in this article can potentially also be used to design high performance neural microcircuits. One can speculate that a design which is optimized for the trade off between separation of inputs and generalization is implemented for example in primary visual areas of many animals. In cat area 17, complex cells are highly selective for features such as orientation, but relatively unaffected by translation of the stimulus in the receptive field of the cells. This behavior is well-suited for tasks such as object recognition where orientation is important, but translation of an object can be regarded as noise.

The analysis of how circuits should be designed in general to achieve optimality with respect to the proposed measures is one direction of future work. Another direction is the investigation of learning rules which are based on the measures presented in this article to self-organize a circuit for a given computational task. Learning rules which bring the dynamics of circuits to the edge of chaos have already been proposed for circuits consisting of threshold gates, see [Legenstein and Maass \(2007\)](#) for a short review.

Altogether we hope that this article provides a small step towards the development of a computational theory that can not only be applied to constructed or strongly simplified models of neural circuits, but also to arbitrarily realistic computer models and to actual data from neural tissue.

Acknowledgements

We would like to thank Nils Bertschinger, Alain Destexhe, Wulfram Gerstner, and Klaus Schuch for helpful discussions. Written under partial support by the Austrian Science Fund FWF, project # P15386; FACETS, project # FP6-015879, of the European Union; and the PASCAL Network of Excellence.

Appendix. Simulation details

A.1. Spike template classification task

In this section, we describe the setup of the spike template classification task in detail. The performance of neural microcircuit models on this task is shown, e.g., in [Fig. 1\(b\)](#). The data for this plot was produced by the following steps:

1. We drew 80 spike templates. Each template consisted of four Poisson spike trains with a rate of 20 Hz and a duration of 200 ms. For training and testing later on, we did not use exactly these spike trains but took jittered versions of the templates. A jittered version of a template was produced by jittering each spike in each spike train by an amount drawn from a Gaussian distribution with zero mean and an SD of 10 ms. Spikes which were outside the time interval of zero and 200 ms after jittering were rejected. Two such spike templates and jittered versions of them are shown in [Fig. 1\(a\)](#).
2. We randomly drew 10 dichotomies on these spike templates. Each dichotomy classified 40 of the 80 spike templates as positive.
3. A neural microcircuit was drawn according to the λ and W_{scale} value in question. For each dichotomy, a neural readout was trained on the state vector of the microcircuit at time $t = 200$ ms with a jittered version of one spike template as input. 2000 training examples were used for training.
4. The performance of one trial was hence determined as the mean over the performances of the 10 readout units on 500 test examples.
5. The result for one W_{scale} and λ pair shown in [Fig. 1\(b\)](#) is the mean over 20 such runs (the SD of these 20 runs is shown in the inset).

A.2. Prediction of computational performance

We describe in detail how we estimated the kernel-quality and generalization capability of microcircuits for the task described above. These estimates are shown in Fig. 5. The following steps show how we estimated the kernel-quality, i.e., the ability of the circuit to discriminate different inputs:

1. We drew 500 spike templates of the same type as in the classification task (four 20 Hz Poisson spike trains over 200 ms). We used the same templates for all $W_{\text{scale}}-\lambda$ pairs of one trial to reduce the variance of the results.
2. A neural microcircuit was drawn according to the λ and W_{scale} value in question. The state vectors of the microcircuit at time $t = 200$ ms with one of the 500 spike templates as input were stored in the matrix M . The rank of the matrix M was estimated by singular value decomposition (Matlab function *rank*).
3. The result for one $W_{\text{scale}}-\lambda$ pair shown in Fig. 1(b) is the mean over 20 such runs (the SD of these 20 runs is shown in the inset).

The estimation of the generalization capability was done similarly. However, instead of using 500 different templates as inputs to the microcircuit, we drew only four different templates and used 500 jittered versions of these four templates (i.e., 125 jittered versions of each template) as inputs. This allowed us to test the response of the circuit on spike jitter.

A.3. Computations in UP- and DOWN-states

The evaluation of computational performance of neural microcircuits for the spike template classification task was done as described in Appendix A.1, with the following differences: Spike templates consisted of 30 spike trains. We considered 30 different spike templates. Jittered versions of spike templates were produced by Gaussian jitter with a SD of 4 ms. Neural readouts were trained on 1000 training examples and tested on 20 test examples.

The input to the microcircuit model for the classification task on firing rates consisted of four Poisson spike trains over 100 ms. Two of them had a rate r_1 , and two had a rate r_2 . Possible rates for r_1 and r_2 were $\{20, 23, \dots, 56\}$ Hz. From these 13^2 possible pairs of rates, 10 were chosen randomly. Furthermore, 30 dichotomies (each classifying 5 rate pairs as positive) on these 10 rate pairs were chosen randomly and kept fixed. A neural microcircuit was drawn according to the parameter values in question. For each dichotomy, a neural readout was trained with linear regression on 2000 training examples. The performance of one trial was determined as the mean over the performances of the 30 readout units on 500 test examples. The result for one parameter value shown in Fig. 8(f) is the mean over 20 such runs. The same simulations were performed for spike trains over 150 and 200 ms and classifications at time points 150 and 200 ms. This was done in order to compare the prediction with classifications at different time points (dashed and dash-dotted lines in Fig. 8(f)).

References

- Anderson, J., Lampl, I., Reichova, I., Carandini, M., & Ferster, D. (2000). Stimulus dependence of two-state fluctuations of membrane potential in cat visual cortex. *Nature Neuroscience*, 3(6), 617–621.
- Bartlett, P. L., & Maass, W. (2003). Vapnik–Chervonenkis dimension of neural nets. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (2nd ed.) (pp. 1188–1192). Cambridge: MIT Press.
- Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, 16(7), 1413–1436.
- Cherkassky, V., & Mulier, F. (1998). *Learning from data*. New York: Wiley.
- Derrida, B. (1987). Dynamical phase transition in non-symmetric spin glasses. *Journal of Physics A: Mathematical and General*, 20, 721–725.
- Destexhe, A., Rudolph, M., & Pare, D. (2003). The high-conductance state of neocortical neurons in vivo. *Nature Reviews Neuroscience*, 4(9), 739–751.
- Häusler, S., & Maass, W. (2007). A statistical analysis of information processing properties of lamina-specific cortical microcircuit models. *Cerebral Cortex* (Epub ahead of print).
- Kantz, H., & Schreiber, T. (1997). *Nonlinear time series analysis*. Cambridge: Cambridge University Press.
- Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. New York: Oxford University Press.
- Langton, C. G. (1990). Computation at the edge of chaos. *Physica D*, 42, 12–37.
- Legenstein, R., & Maass, W. (2007). What makes a dynamical system computationally powerful? In S. Haykin, J. C. Principe, T. Sejnowski, & J. McWhirter (Eds.), *New directions in statistical signal processing: From systems to brain* (pp. 127–154). Cambridge: MIT Press.
- Maass, W., Legenstein, R. A., & Bertschinger, N. (2005). Methods for estimating the computational power and generalization capability of neural microcircuits. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems: Vol. 17* (pp. 865–872). Cambridge: MIT Press.
- Maass, W., & Markram, H. (2004). On the computational power of recurrent circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4), 593–616.
- Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11), 2531–2560.
- Maass, W., Natschläger, T., & Markram, H. (2004). Fading memory and kernel properties of generic cortical microcircuit models. *Journal of Physiology – Paris*, 98(4–6), 315–330.
- Maass, W., & Orponen, P. (1998). On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10, 1071–1095.
- Maass, W., & Sontag, E. D. (1999). A precise characterization of the class of languages recognized by neural nets under gaussian and other common noise distributions. In M. S. Kearns, S. S. Solla, & D. A. Cohn (Eds.), *Advances in neural information processing systems: Vol. 11* (pp. 281–287). Cambridge: MIT Press.
- Markram, H., Wang, Y., & Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*, 95, 5323–5328.
- Mayor, J., & Gerstner, W. (2005). Signal buffering in random networks of spiking neurons: Microscopic versus macroscopic phenomena. *Physical Review E*, 72, 051906.
- Mitchell, M., Hraber, P. T., & Crutchfield, J. P. (1993). Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems*, 7, 89–130.
- Packard, N. (1988). Adaption towards the edge of chaos. In J. A. S. Kelso, A. J. Mandell, & M. F. Shlesinger (Eds.), *Dynamic patterns in complex systems* (pp. 293–301). Singapore: World Scientific.
- Shu, Y., Hasenstaub, A., Badoual, M., Bal, T., & McCormick, D. A. (2003). Barrages of synaptic activity control the gain and sensitivity of cortical neurons. *The Journal of Neuroscience*, 23(32), 10388–10401.
- Siegelmann, H., & Sontag, E. D. (1994). Analog computation via neural networks. *Theoretical Computer Science*, 131(2), 331–360.
- Vapnik, V. N. (1998). *Statistical learning theory*. New York: John Wiley.