Available online at www.sciencedirect.com

ScienceDirect

Current Opinion in Behavioral Sciences

Searching for principles of brain computation

Wolfgang Maass

Experimental methods in neuroscience, such as calcium-imaging and recordings with multi-electrode arrays, are advancing at a rapid pace. They produce insight into the simultaneous activity of large numbers of neurons, and into plasticity processes in the brains of awake and behaving animals. These new data constrain models for neural computation and network plasticity that underlie perception, cognition, behavior, and learning. I will discuss in this short article four such constraints: inherent recurrent network activity and heterogeneous dynamic properties of neurons and synapses, stereotypical spatio-temporal activity patterns in networks of neurons, high trial-to-trial variability of network responses, and functional stability in spite of permanently ongoing changes in the network. I am proposing that these constraints provide hints to underlying principles of brain computation and learning.

Address
Graz University of Technology, Institute for Theoretical Computer Science, Inffeldgasse 16b/I, A-8010 Graz, Austria

Corresponding author: Maass, Wolfgang (maass@igi.tugraz.at)

Current Opinion in Behavioral Sciences 2016, **11**:81–92

This review comes from a themed issue on **Computational modeling**

Edited by **Peter Dayan** and **Daniel Durstewitz**

http://dx.doi.org/10.1016/j.cobeha.2016.06.003

Constraint/principle 1: neural circuits are highly recurrent networks consisting of different types of neurons and synapses with diverse dynamic properties

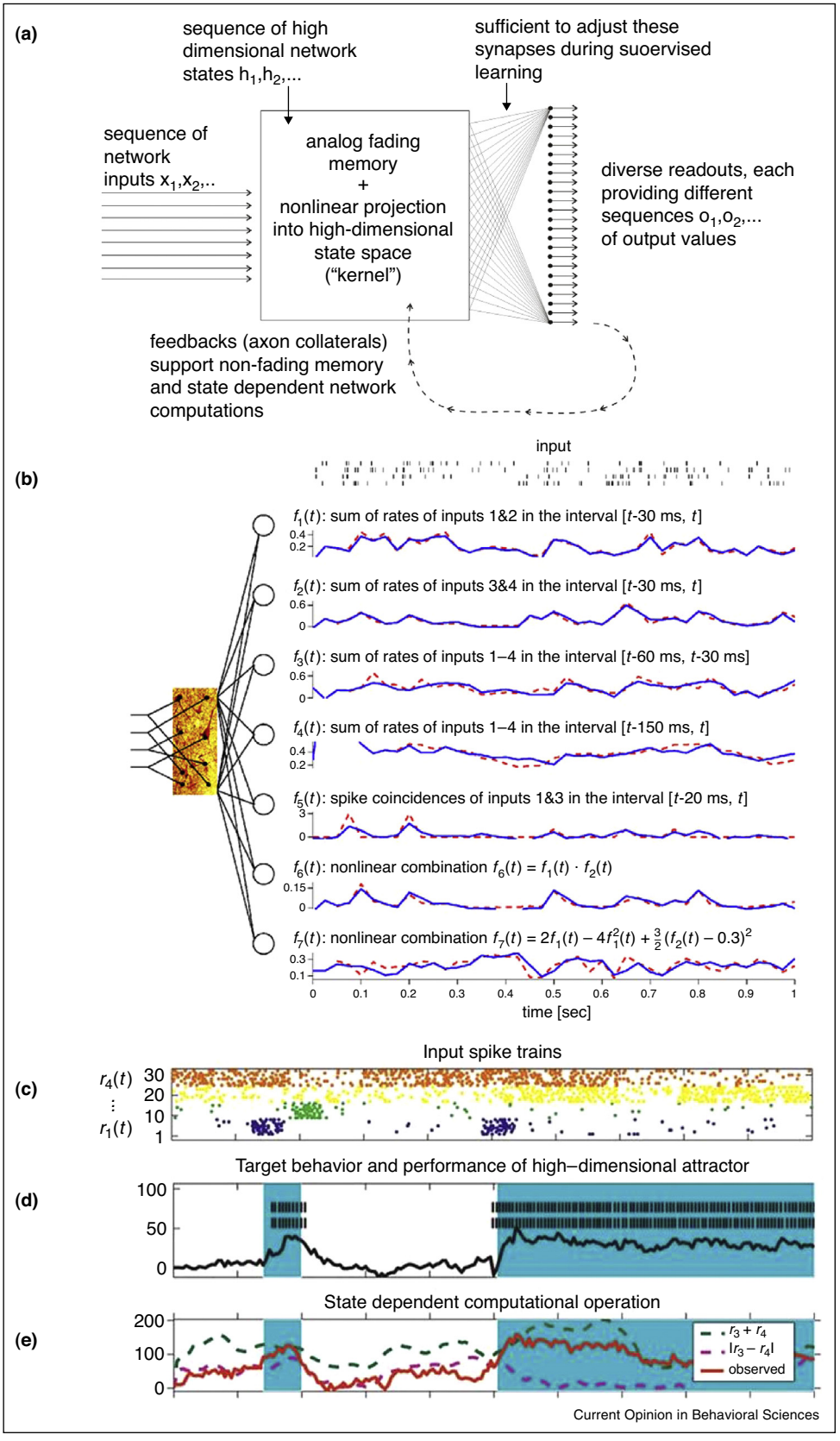
Most computations in our current generation of digital computers have a feedforward organization, where specific network modules carry out specific subcomputations and transmit their results to the next module. One advantage of this computational organization is that it is easy to understand and control. Also deep learning networks favor feedforward computation (and backwards propagation of errors or predictions) because this organization best supports currently known algorithms for network learning. But nature has apparently discovered a way of fully using recurrent neural networks for reliable computation and learning that is based on different principles. Already evolutionary very old nervous systems, such as those of hydra [1], *C. elegans* [2•], and zebrafish [3] are highly recurrent and exhibit a complex global network dynamics, similar as brain networks in more advanced species.

These activity patterns differ in several aspects from those that we encounter in our digital computers. Hence recent reviews [4•,5•] have emphasized the need to understand the basic principles of brain computations in recurrent networks of neurons.

Numerous theoretical and modeling studies have analyzed the dynamics of simple recurrent networks of homogeneous types of neurons and synapses, see for example, [6–8]. But neural networks in the brain consist of different types of neurons [9] that are connected by different types of synapses with heterogeneous shortterm and longterm dynamics [10–12]. These features of biological networks of neurons make a theoretical analysis difficult, and they constrain computational models. In particular, biological networks of neurons are not well-suited for emulating generic computations of Boolean circuits or artificial neural networks. For example, it would be difficult to implement tensor product variable binding [13] in a neural network model which takes into account that there are excitatory and inhibitory neurons with different dynamic properties, that neurons do not emit analog values but spikes at low firing rates, and that synapses are subject to noise and short-term dynamics (i.e., mixtures of paired pulse facilitation and depression). The short-term dynamics of synapses lets the amplitudes of postsynaptic potentials decrease or increase for a sequence of spikes in dependence of the pattern of preceding spikes. This history-dependence obstructs a stable transmission of spikes and firing rates, which we would need for emulating a Boolean circuit or artificial neural network. The obvious question is of course whether the experimentally found diversity of units, mechanisms, and time-constants in brain networks is detrimental for all types of computations, or whether it could enhance specific computational operations that nature has discovered.

One computational model for which a diversity of computational units and time constants is not detrimental, and in fact benefitial, is the liquid computing paradigm [14•,15•,16]. It is sometimes subsumed together with the somewhat similar echo-state model of [17•] under the name reservoir computing [see chapter 20 of [8]]. A common feature of both types of reservoir computing models is that they conceptually divide neural network computations into two stages (see Figure 1a), a fixed

Figure 1


(a)
sequence of high dimensional network states $h_1,h_2,...$
sufficient to adjust these synapses during suoervised learning
sequence of network inputs $x_1,x_2,..$
analog fading memory + nonlinear projection into high-dimensional state space ("kernel")
diverse readouts, each providing different sequences $o_1,o_2,...$ of output values
feedbacks (axon collaterals) support non-fading memory and state dependent network computations
(b)
input
$f_1(t)$: sum of rates of inputs 1&2 in the interval [t-30 ms, t]
$f_2(t)$: sum of rates of inputs 3&4 in the interval [t-30 ms, t]
$f_3(t)$: sum of rates of inputs 1–4 in the interval [t-60 ms, t-30 ms]
$f_4(t)$: sum of rates of inputs 1–4 in the interval [t-150 ms, t]
$f_5(t)$: spike coincidences of inputs 1&3 in the interval [t-20 ms, t]
$f_6(t)$: nonlinear combination $f_6(t) = f_1(t) \cdot f_2(t)$
$f_7(t)$: nonlinear combination $f_7(t) = 2f_1(t) - 4f_1^2(t) + \frac{3}{2}(f_2(t) - 0.3)^2$
time [sec]
(c)
Input spike trains
$r_4(t)$
$r_1(t)$
(d)
Target behavior and performance of high–dimensional attractor
(e)
State dependent computational operation
$r_3 + r_4$
$|r_3 - r_4|$
observed
Current Opinion in Behavioral Sciences

generic nonlinear preprocessing stage and a subsequent stage of linear readouts that are trained for specific computational tasks. A structural difference between the liquid computing model and the echo-state model is that the latter assumes that there is no noise in the network, and that analog outputs of computational units can be transmitted with arbitrary precision to other units. In contrast, the liquid computing model is geared toward biological neural networks, where noise, diversity of units, and temporal aspects related to spikes play a prominent role.

A diversity of units and time constants in a recurrent neural network causes no problem if one analyzes its computational contribution from the perspective of a projection-neuron or readout neuron (such as pyramidal cells on layers 2/3 and 5 [9]) that receives synaptic inputs from thousands of neurons within the recurrent neural network, and extracts information for other networks. From this perspective it is not required that neurons in the recurrent neural network complete specific subcomputations. It suffices if diverse neurons and subcircuits within the recurrent network produce a large number of potentially useful features and nonlinear combinations of such features, out if which a projection neuron can select and combine through a weighted sum — or a more complex dendritic integration — useful information for its target networks. In this way even a seemingly chaotic dynamics of a recurrent local network can make a useful computational contribution [14•,15•,16,18–20].

This perspective raises the question how a recurrent neural network could optimally support through generic computational preprocessing subsequent readout neurons. Some theoretical foundation (see [15•,18,21•,22], and Figure 1a for details) arises through a link to one of the most successful learning approaches in machine learning: Support Vector Machines (SVMs; [23]). A SVM also consists of two stages: a generic nonlinear preprocessing stage (called kernel) and linear readouts. The kernel projects external input vectors x_1, x_2, ... nonlinearly onto vectors h_1, h_2, ... in a much higher dimensional space. One can view a large nonlinear recurrent neural network as an implementation of such a kernel, where the network response h_i to an input x_i corresponds to the kernel output. This network response h_i can be defined for example as the high-dimensional vector that records for each neuron in the network its recent firing activity, say within the last 30 ms (as in Figure 3b). This network response h_i provides then the synaptic input to any readout neurons. It represents the 'visible' part of the network state, while other 'hidden' dimensions of the true network state, such as the current internal state of dynamic synapses is not visible for readout neurons [16]. If the map from x_i to h_i is nonlinear, the network can increase the expressive capability of subsequent linear readouts. For example, if the network states h_1, h_2, ... would contain all products of components of the network inputs x_1, x_2, ..., a linear readout from these network states attains the same expressive capability as a quadratic readout function in terms of the original network inputs x_1, x_2, ... The quality of the kernel operation of a neural network can be measured by the dimension of the linear vector space that is spanned by the ensemble of network states h_1, h_2, ... which result for some finite ensemble of different network inputs x_1, x_2, ... This dimension is equal to the rank of the matrix with columns h_1, h_2, ... This approach to measure the computational power of a neural circuit through a dimensionality analysis was introduced in [18,21•], and later applied to experimental data in [22], see [24] for a review. It provides an alternative to approaches based on the analysis of neural codes and tuning curves of individual neurons for specific simple stimuli. It provides instead a paradigm for analyzing neural codes for complex natural stimuli x_i on the network level — from the perspective of neural readouts. The kernel property of a neural network would be theoretically optimal if it would map any ensemble of different external inputs x_1, x_2, ... onto linearly independent network states $h_1, h_2, \ldots$. A linear readout can assign through a proper choice of its weights any desired output values o_1, o_2, ... to linearly independent network states h_1, h_2, ... If the vectors $h_1, h_2, \ldots$ are not linearly independent, then the rank of the matrix with these columns tells us how much of these theoretically ideal expressive capabilities of linear readouts remain. A more subtle analysis is needed to integrate noise-tolerance into this network level analysis

(Figure 1 Legend) Computational paradigms resulting from principle 1. **(a)** Generic computational model. **(b)** Demonstration that a randomly connected network of 135 spiking neurons with diverse short-term plasticity of synaptic connections supports multiplexing: Different linear readouts can be trained to produce simultaneously different online computations, in this example on two time-varying firing rates $f_1(t)$ and $f_2(t)$ represented by the spiking activity of the first 2 and the last 2 input neurons shown at the top [83]. Outputs of the linear readouts are plotted as blue curves, target outputs as dashed red curves. **(c)–(e)** Demonstration of additional computational properties that arise with feedback from trained readouts: nonfading memory and context-depending switching of computations. **(c)** 4 spike input streams with timevarying Poisson firing rates $r_1(t), \ldots, r_4(t)$. (d) Two spiking readouts with feedback (their spike outputs are shown in black) were trained to remember which of the first two input streams had last exhibited a burst (time points where $r_1(t)$ had the most recent burst are marked in blue). **(e)** Another readout was trained to output spikes with rates $r_3(t) + r_4(t)$ or $|r_3(t) - r_4(t)|$ (both shown as dashed curves) in dependence of the binary state represented in (d). The orange curve shows the resulting output of this readout neuron, that approximates $r_3(t) + r_4(t)$ during one network state (indicated by blue) and otherwise $|r_3(t) + r_4(t)|$.

Source: Figure 1b is reprinted from 'Theory of the Computational Function of Microcircuit Dynamics,' in 'Microcircuits: The Interface between Neurons and Global Brain Function,' edited by Sten Grillner and Ann M. Garybiel, published by ''The MIT Press, pp. 371–390, 2006', with kind permission from The MIT Press. Figure 1d,e is reprinted from Ref. [37] 'Computational aspects of feedback in neural circuits', in 'PLOS Computational Biology, 3(1):e165, 2007', with kind permission from The PLOS Journals.

of neural coding [18,21•,22], since a readout needs to be able to assign target outputs in a trial-invariant manner. Hence one needs to distinguish linear independence of network states h_i caused by saliently different inputs x_i from accidental linear independence caused by noise. But if the network is sufficiently large and nonlinear, it tends to endow a simple linear readout even in the presence of noise with the computational and learning capability of a more complex nonlinear readout. In addition, the resulting two stage network has a very desirable learning dynamics if only the weights of a linear readout are adjusted: there are no local minima in its error function — hence gradient descent arrives in the end at the global optimum.

One other benefit of such a two-stage computational model is its multiplexing efficiency: the same first stage (the kernel) can be shared by an unlimited number of subsequent linear projection neurons (indicated on the right in Figure 1a), that learn to extract different computational results for their specific target networks (see a simple demo in Figure 1b). This feature does not depend on specific aspects of the model, but is shared with any model wich proposes that generic cortical microcircuits generate a menu of features that supports a variety of downstream computations. Such multiplexing of computations through parallel readouts from a common recurrent network provides an alternative to models based on a precise ice-cube-like spatial organization of sub-computations in a cortical column, see for example, [25] for a discussion. Recent experimental data [26] suggest that different projection neurons do in fact extract from the same local microcircuit quite different results.

So far I have only addressed static computations on batch input vectors x_i. A further computational benefit of having diverse units in a neural network, especially units with a wide spread of time constants, becomes apparent if one takes into account that many brain computations have to integrate information from several preceding time windows. An important class of such computations are computations on time series with a fading memory. These are computations where the output at time t may also depend on inputs that have arrived before time t in the recent past. Surprisingly, even the arguably most complex nonlinear transformations that map input time series onto output time series with a fading memory, Volterra series, can be implemented through simple memory-less readouts from any ensemble of filters that have a sufficiently wide spread of time constants. If one views synapses with their inherent short-term dynamics as filters, then it suffices if the network contains synapses with diverse shortterm dynamics. More precisely the following separation property of the ensemble of filters is relevant (see Theorem 1 in [14•] and [27] for further details): Does at least one of the filters produce at the current time t different output values for two input time series that differed at some point in the recent past? It has recently been shown that both cultured neural circuits [28] and ganglion cells in the retina [29] have a good separation property of this kind. A good separation property entails that output o_3 of a linear readout at time step 3 may depend nonlinearly not only on the current network input x_3, but also on preceding network inputs x_1 and x_2. The liquid computing model (see Figure 1a) postulates that the separation property is, in addition to the previously discussed kernel property, a basic computational property of generic neural circuits. This prediction of the model was subsequently verified through recordings from visual [30] and auditory [31] primary cortex. In principle, even a working memory can be composed according to this analyzis from local units or modules of a recurrent neural network that have different time constants [32–34].

An interesting question is which details of biological neural circuits are essential for maximizing their kernel-property and separation property (see [15•,35] for some first results). Recurrent connections, diversity of neuron types, and diversity of synapse types all appear to contribute to the kernel-property and separation property. But not all of these features appear to be necessary for that. An alternative view of the experimentally found complexity of neural circuits is that tightly structured connectivity, homogeneity of neurons, and homogeneity of synapses are essential properties of human-designed computational circuits, but are task-irrelevant dimensions for biological neural circuits, because readout neurons with adaptive capabilities can compensate for inhomogeneities and deficiencies of the circuits which provide inputs to them. In other words many details of neural circuits can be viewed as task-irrelevant dimensions. A more specific functional role of diverse types of synaptic dynamics for stabilizing network activity was proposed in [36].

The importance of readouts becomes even larger if one no longer assumes that their output only affects downstream networks. Mathematical results [37•] imply that the capability of the liquid computing model is substantially enhanced if linear readout neurons — that are trained for specific tasks — are allowed to project their output also back into the local network (see dashed loop at the bottom of Figure 1a). In fact, most projection neurons from a generic cortical microcircuit do have axon collaterals, which carry out such back projections. The essential structural difference to the model without feedback is that now the training also affects the dynamics of the recurrent network itself. Under ideal conditions without noise this model with feedback acquires the computational power of a universal Turing machine [37•]. But computer simulations (see Figure 1c,d) show that the feedback also adds in the presence of noise important computational capabilities to the liquid computing model: It now can remember salient inputs in its internal state

for unlimited time (see Figure 1c,d). Furthermore it can switch its computational function in dependence of its internal state, and hence also in dependence of external cues (Figure 1e). Experimental data [38] have subsequently shown that cortical networks of neurons do in fact have these theoretically predicted enhanced computational capabilities.

Training of linear readouts with feedback requires well-tuned learning methods because the resulting closed loop tends to amplify the impact of changes of synaptic weights to readout neurons. Two successful methods for supervised learning [19,37•] employ and refine related methods for echo state networks [39]. These learning methods work well in simulations, but do not aspire to be biologically realistic. Biologically more realistic learning methods based on reinforcement learning were proposed in [20,40].

Constraint/principle 2: neural network activity is dominated by variations of assembly activations

From the perspective of some theories of neural coding and computations it would be desirable that different neurons in a local network can encode independently of each other specific features of a sensory stimulus. However virtually all simultaneous recordings from many neurons within a local patch of cortex suggest that the joint activity patterns of nearby neurons are restricted to variations of a rather small repertoire of spatio-temporal firing patterns. One usually refers to these patterns as assemblies, assembly sequences, or packets of activity [41]. It was shown in [42•] that patches of auditory cortex typically respond with variations of just one or two different joint activity patterns to a repertoire of over 60 auditory stimuli, and to continuously morphed stimuli. Also patches of V1 in rodents appear to respond to natural movies with variations of just a few joint activity patterns [43•]. Furthermore a small repertoire of activity patterns tends to occur also spontaneously [44•]. The fact that a small repertoire of joint activity patterns also occurs in slice [45] supports the conjecture that these patterns are consequences of network architecture and parameters that result from an interplay of the genetic code and plasticity processes. In particular, learned behaviors have been shown to become encoded by similar stereotypical joint activity pattern in the higher cortical areas PFC (prefrontal cortex) [46•] and PPC (posterior parietal cortex) [47•].

However it has remained open how neural networks compute with these stereotypical joint activity patterns. In order to test this in models, one first has to find ways of inducing their emergence. Ref. [48] showed that stereotypical patterns emerge through STDP (spike-timing dependent plasticity) in recurrent networks with very little noise even in the absence of external inputs. More recently such patterns have also been induced through STDP in such a way that they encode the class to which an input pattern belongs [49•]. Whereas this model used simplified lateral inhibition, Figure 2a,b shows that similar pattern emerge through STDP in networks with explicitly modeled inhibitory neurons [50]. Furthermore it has been shown that input-dependent assemblies also emerge in models that employ in addition synaptic plasticity for inhibitory synapses [51]. One computational benefit that is suggested by these models is that assembly coding facilitates the learning task of readout neurons: They are able to learn very fast — even without supervision (see Figure 2c,d) — to report which assembly is currently active, and hence to which class an input pattern belongs.

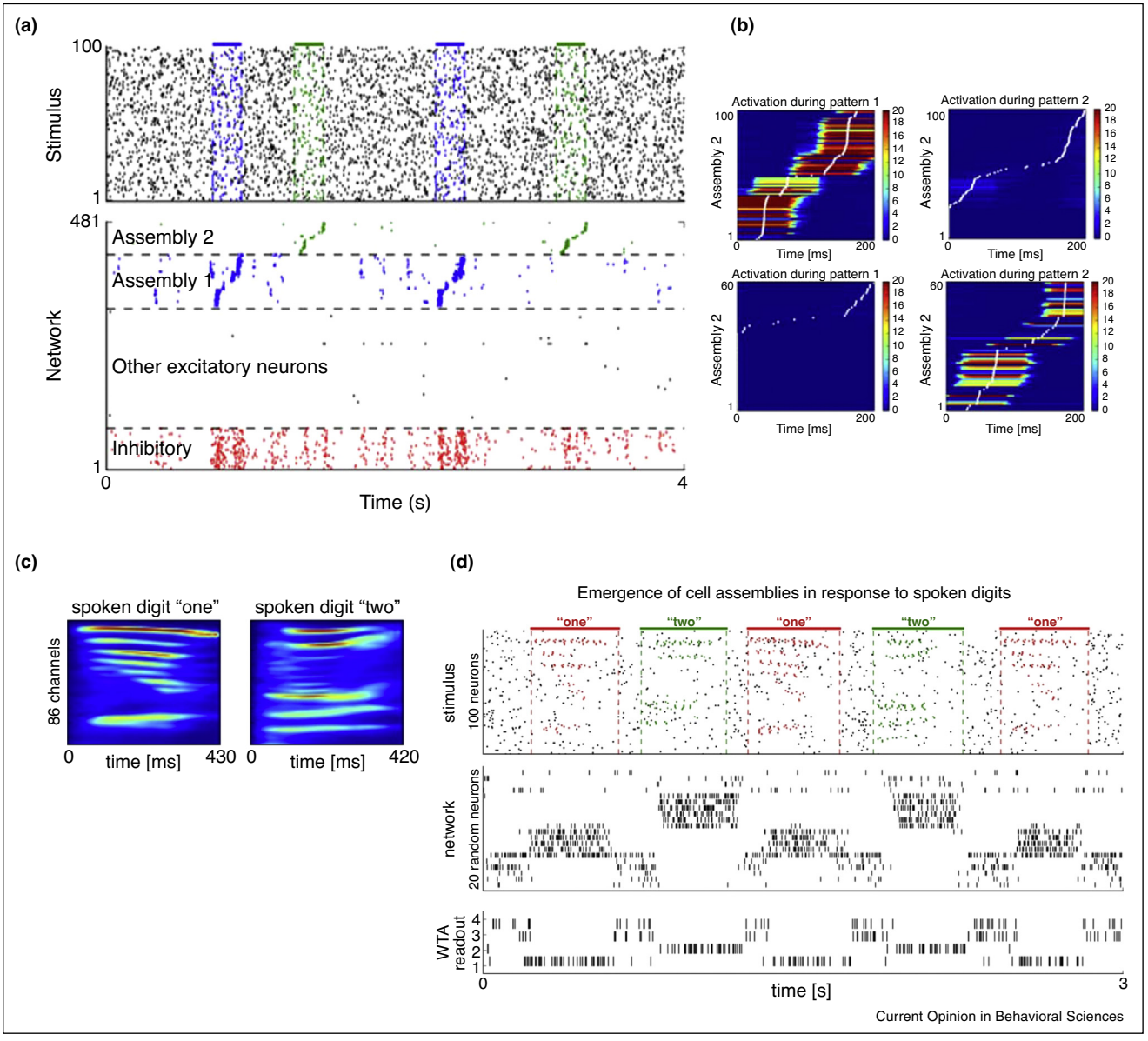
Assemblies and assembly sequences had already been postulated by [52] to be tokens of network dynamics that create links between the fast time scale of spikes and the slower time scale of cognition and behavior. Ref. [53•] proposed to view assemblies as word-like codes for salient objects, concepts, among others that are combined in the brain through a yet unknown type of 'neural syntax'.

Constraint/principle 3: networks of neurons in the brain are spontaneously active and exhibit high trial-to-trial variability

Virtually all neural recordings show that network responses vary substantially from trial to trial. This is not surprising, since channel kinetics in dendrites and synaptic transmission are reported to be highly stochastic [54,55]. These data force us to add a substantial amount of variability or noise to the set of constraints for neural network computations. Again, a key question is whether this constraint can also be viewed as a principle that provides a clue for understanding the organization of brain computations. Usually noise is just seen as a nuisance in a computational system [56].

Hints for a possible benefitial role of large trial-to-trial variability for brain computations is provided by experimental data which suggest that ambiguous sensory stimuli are represented in brain networks through flickering between different network states, that each represent one possible interpretation of the ambiguous stimulus [57,58•]. Also the values of possible choices appear to be represented in monkey orbitofrontal cortex (OFC) before decision making through flickering between corresponding network states [59•] on a small time scale like in Figure 3b. These new data from simultaneous recordings from many neurons with high temporal precision suggest that 'subjective decision-making involves the OFC network transitioning through multiple states, dynamically representing the value of both chosen and unchosen options' [59•]. A well-known approach for probabilistic inference (Markov chain Monte Carlo or MCMC sampling, see [23,59•]) suggests to interpret

Figure 2



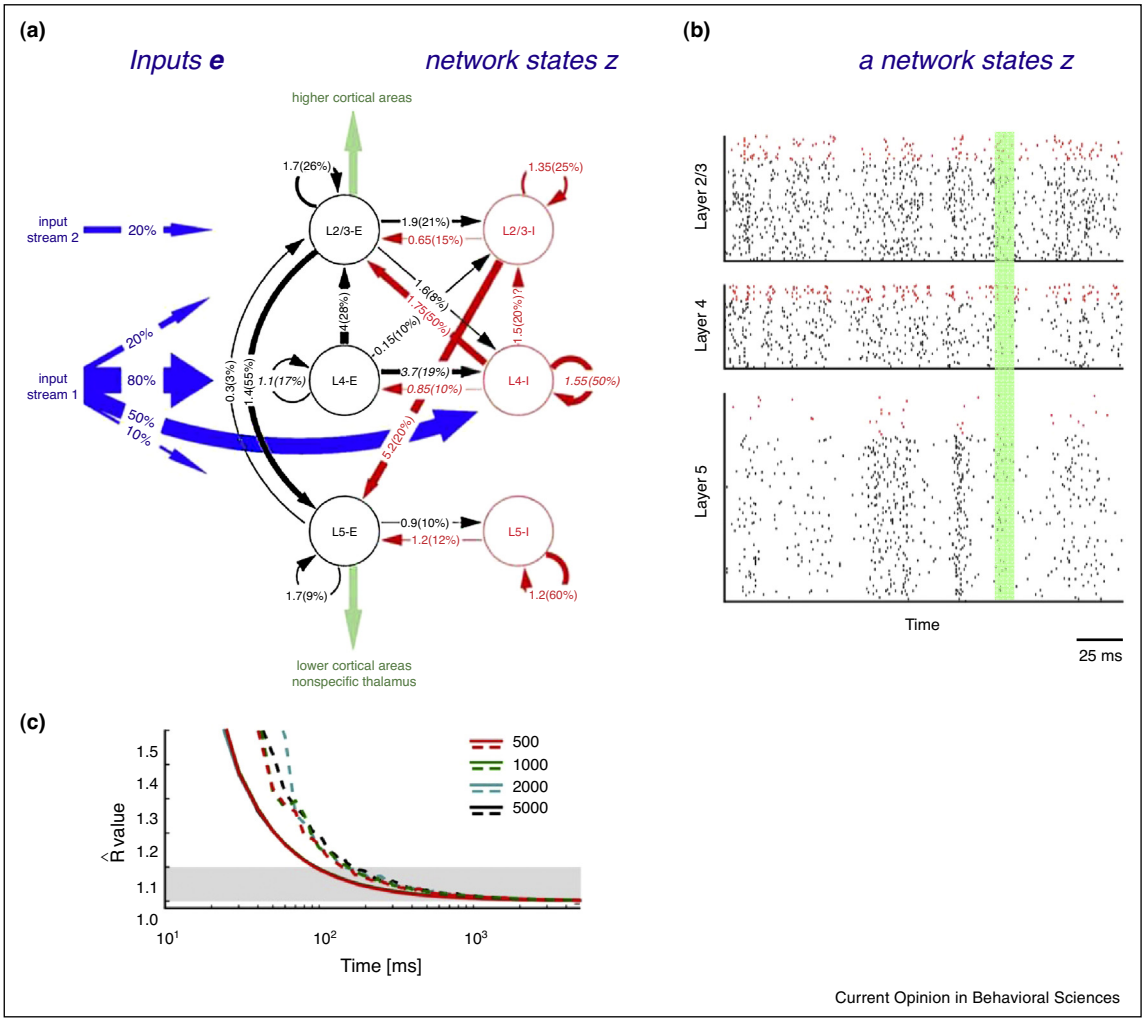
Emergence and computational use of assembly codes. **(a)** Emergence of input-specific assemblies (or more precisely, assembly sequences) through STDP in response to repeating external input patterns (there are blue and green spike patterns which are superimposed by noise spikes shown in black). This occurs even if the input patterns (frozen Poisson patterns) have exactly the same rates and statistics as the noise input between patterns. The assembly sequences shown in (a) and (b) emerge after about 100 occurrences of each input pattern in a generic recurrent neural network [50]. **(b)** The mean firing time of each excitatory neurons is marked by a white dot, with a histogram of all firing times represented through color coding in the same row (analogously as in [44•]). **(c)** Sample utterings of two spoken digits that were transformed into spike inputs shown in the top row of d) in a network simulation from Ref. [49•]. The middle row of **(d)** shows the two emergent assemblies for the two spoken words 'one' and 'two'. The bottom row of (d) shows the firing response of 4 linear readout neurons in a WTA (winner take all) circuit. This WTA readout learns without any supervision to report the occurrence of one of the two assembly sequences, and hence spoken digit classification, through the firing of neurons 1 and 2.

Source: Panels c,d of Figure 2 are reprinted from Ref. [49•] 'Emergence of dynamic memory traces in cortical microcircuit models through STDP' published by 'The Journal of Neuroscience, 33(28):11515–11529', 2013, with kind permission from The Journal of Neuroscience.

these experimental data as probabilistic inference through stochastic computation, more precisely through sampling from some internally stored probability distribution of network states.

Insight into the nature of such internally stored probability distribution can in principle be gained by analyzing the statistics of network states, defined for example by a binary vector with a '1' for every neuron that fires within

Figure 3



Model for probabilistic inference through sampling in a generic cortical microcircuit model with stochastically spiking neurons [65•]. **(a)** Synaptic weights and other parameters encode a unique stationary distribution $p(\mathbf{z})$ of network states **z**. **(b)** The network state **z** at time t can be defined for example as a binary vector [60•] that records which neuron fires in a small time window around t (shaded in green). The stochastic dynamics of the network for some external input **e** can be interpreted as sampling from the conditional distribution $p(\mathbf{z}|\mathbf{e})$. **(c)** Instead of traditional measures for computation time, the time needed to converge from an initial state to the stationary distribution of network states (not to any particular state!) becomes relevant. A standard heuristic estimate (Gelman–Rubin analysis) suggests that this convergence is quite fast — around 100 ms — and independent of the network size (color coded for network sizes between 500 and 5000 neurons) for the data-based model from (a). Likely reasons for independence from network size are small synaptic weights, weight normalization, and a large amount of stochasticity in the model. The Gelman–Rubin analysis suggests that convergence to the stationary distribution has taken place by the time when the curves (solid lines: mean; dashed lines: worst case) enter the gray zone below 1.1.

Source: Figure 3 is reprinted from 'Stochastic computations in cortical microcircuit models' published by 'PLOS Computational Biology, 9(11):e1003311, 2013', with kind permission from The PLOS Journals.

some small time bin [60•] (see Figure 3b). The term 'neural sampling' had been coined in Ref. [61•] for the resulting theory of probabilistic inference through sampling in stochastically firing recurrent networks of neurons. Each neuron v_i represents in this model a binary random variable z_i through spikes: a spike sets the value of this random variable to 1 for some short period of time. It was shown in Ref. [61•] that if synaptic weights are symmetric, a network of simple models for spiking neurons can represent the same probability distribution as a Boltzmann machine with the same architecture, although it uses a different sampling strategy. This is interesting because a Boltzmann machine is one of the most studied neural network models in machine learning for probabilistic inference and learning, and it is known that it can learn and represent any multi-variate distribution over binary random variables with at most 2nd order dependencies. In addition it was shown that a suitable architecture enables

networks of spiking neurons with asymmetric weights to go beyond that: a spiking network can represent [62] and learn [63] any distribution over discrete variables, even with higher order dependencies as they occur, for example, in the explaining-away effect of visual perception [64].

Also data-based models for generic cortical microcircuits (Figure 3a) with stochastically firing neurons can carry out probabilistic inference through sampling. For example they can estimate through sampling posterior marginals such as $p(z_1|\mathbf{e}) = \sum_{a_2,\ldots,a_m} p(z_1, a_2, \ldots, a_m|\mathbf{e})$, where $\mathbf{e}$ is some external input [65•]. The current external input $\mathbf{e}$ could represent for example sensory evidence and internal goals. The variables a_i for $i > 1$ run in this formula over all possible values of random variables z_i that are irrelevant for the current probabilistic inference task. The sum indicates that these variables are marginalized out, which is in general a computationally very demanding (in fact: NP-hard) operation. The binary variable z_1 could represent, for example, the choice between two decisions, so that an estimation of the posterior marginal $p(z_1|\mathbf{e})$ supports Bayes-optimal decision making. The key point is that this computationally very difficult posterior marginal can be estimated quite easily through sampling: It is represented by the firing rate of the neuron v_1 that corresponds to the binary random variable z_1 [65•]. Also sampling-based representations of time-varying probabilities — where each random variable is represented through several spiking neurons — have been examined [66,67]. At the current time point it is not yet clear to what extent brains make use of the option to carry out probabilistic inference through sampling. To answer this question one needs further experimental insight into the relation between flickering internal states of brain networks on one hand and perception and behavior on the other hand. Refs. [57,59•] have demonstrated that this is in principle feasible.

Stochasticity of spiking neurons conveys another computational benefit to a network: it enables the network to solve problems — for example, constraint satisfaction problems — in a heuristic manner [56,65•,68]. Here each network state (defined like in Figure 3b) represents a possible solution to a problem, and the frequency of being in this network state encodes the quality (fitness) of the solution. This computational model is consistent with the data from [59•], where easier choices were associated with fewer switches between the neural representation of the two options. The computation time for solving a task depends in such a sampling-based model on the time that the network needs until it produces, starting at some given initial state, samples from the stationary distribution of network states (see Figure 3c), which is defined by the architecture and parameters of the network [65•]. A substantial level of noise in the network and not too large synaptic weights support in general fast convergence [65•].

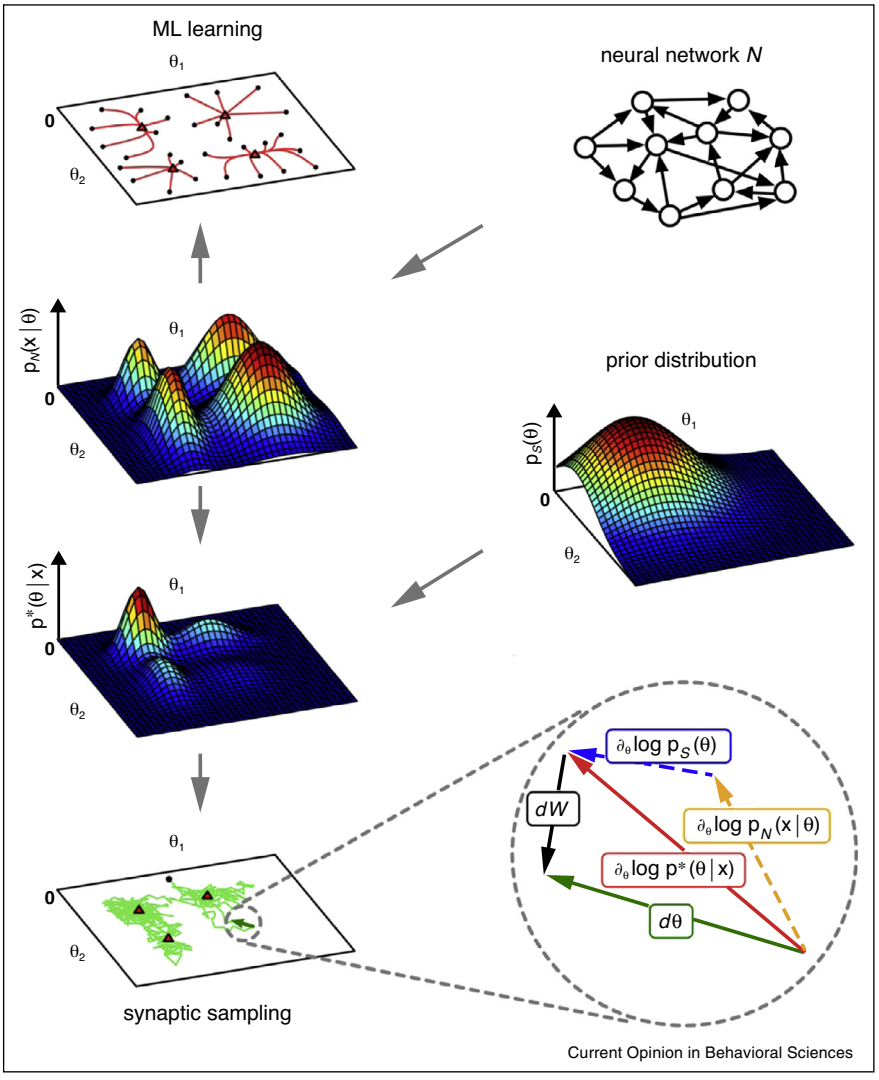
The hypothesis that the human brain encodes substantial amounts of knowledge in the form of probabilities and probability distributions had previously been proposed in cognitive science [69–71]. Probabilistic computations have started to play a prominent role in many models in neuroscience, for example in models for multisensory integration [72] and confidence [73].

Constraint/principle 4: networks of neurons in the brain provide stable computational function in spite of ongoing rewiring and network perturbations

Experimental data show that network connectivity [74•,75,76], neurotransmitters [77] and neural codes [78•] are subject to continuously ongoing changes, even in the adult brain. This constraint suggests to consider the hypothesis that the brain samples not only network states on the fast time scale of spiking activity as discussed under principle 3, but simultaneously also different network configurations on the slower time scale of network plasticity and spine dynamics (i.e., hours and days). This slower sampling of network configurations has been called synaptic sampling [79•]. The synaptic sampling model suggests that brain networks do not converge to a desirable network configuration and stay there, but rather sample continuously — but at different speeds ('temperatures') — from a posterior distribution of network configurations (Figure 4).

Learning a posterior distribution of network configurations, rather than a specific network configuration, has been proposed to be a more attractive goal for network plasticity — for example, because of better generalization capability [80]. The question how a biological network of neurons could represent and learn such a posterior distribution was described in [72] as a key open problem. Ref. [79•] proposes that this posterior distribution is represented by a stationary distribution of network configurations in the Markov chain that is defined by the stochastic dynamics of rewiring, STDP, and noise in synaptic weights. The Fokker–Planck equation provides a transparent link between local stochastic rules for synaptic plasticity and spine dynamics, and the resulting stationary distribution $p^*(\boldsymbol{\theta})$ of network configurations $\boldsymbol{\theta}$. Learning is viewed from this perspective as convergence to a lower dimensional manifold of network configurations that provides good compromises between computational function and structural constraints. Structural constraints take the form of a prior in this model (see Figure 4). One interesting benefit of this conceptual alternative to maximum likelihood learning is that the network immediately and automatically compensates for internal or external changes that modify the posterior distribution of network configurations (see Figure 5 of [79•]). But the underlying stochastic theory suggests that network configurations are likely to change continuously in functionally irrelevant dimensions — even in the absence of major perturbations.

Figure 4



Two different options for the organization of network learning. Assume that some recurrent neural network $\mathcal{N}$ is given (top right), together with a generative model $p_{\mathcal{N}}(\boldsymbol{x}|\boldsymbol{\theta})$ (second plot from the top in the left column) for a given ensemble $\boldsymbol{x}$ of network inputs. Maximum likelihood learning moves the parameter vector $\boldsymbol{\theta}$ of the network from a given initial state (black dot) to a local maximum of $p_{\mathcal{N}}(\boldsymbol{x}|\boldsymbol{\theta})$ (red triangles in top panel of left column). In contrast, the Bayesian synaptic sampling approach takes in addition a prior $p_S(\boldsymbol{\theta})$ (middle panel in right column) into account — that could encode for example sparsity constraints — and aims at sampling network parameters θ from the posterior distribution $p^*(\boldsymbol{\theta}|\boldsymbol{x}) \propto \sim p_S(\boldsymbol{\theta}) p_{\mathcal{N}}(\boldsymbol{x}|\boldsymbol{\theta})$ (left column, 3rd panel from the top). This can be achieved through a synaptic plasticity rule that takes the form of a stochastic differential equation with a drift term $\partial_{\boldsymbol{\theta}} \log p^*(\boldsymbol{\theta}|\boldsymbol{x})$ (red arrow in panel at the right bottom) that results from derivations of the log of the prior (blue arrow) and likelihood (yellow arrow), together with a stochastic diffusion term $d\mathcal{W}$ (black arrow). The Fokker–Planck equation implies that $p^*(\boldsymbol{\theta}|\boldsymbol{x})$ is the unique stationary distribution of this stochastic parameter dynamics ('synaptic sampling'). A sample trajectory of the parameter vector $\boldsymbol{\theta}$ is plotted in green in the bottom left panel. Because of its stochastic component $d\mathcal{W}$ this learning approach can easily integrate stochastic spine dynamics with STDP, see (Kappel *et al*., 2015) [79•] for details. The high-dimensional space of network parameters $\boldsymbol{\theta}$ is replaced in this figure for illustration purposes by a 2D space.

Source: Figure 4 is reprinted from Ref. [79•] 'Network plasticity as Bayesian inference' published by 'PLOS Computational Biology, 11(11):e1004485, 2015', with kind permission from The PLOS Journals.

A rethinking of the way in which network organization and plasticity is genetically encoded and implemented in the brain has been suggested by [81•]. This challenge was motivated by the observation that the same neural circuit attains at different times and in different individuals the same performance with quite different parameter settings. The synaptic sampling perspective suggests an explanation for this observation: Each measurement of network parameters and synaptic connectivity provides a snapshot from an ongoing stochastic process.

Conclusions

The four constraints/principles for models of brain computation and learning that I have discussed are compatible with each other. They have to be compatible, since experimental data tell us that they are all present in brain networks. But obviously there are tradeoffs between these principles. For example, more stereotypical network responses (principle 2) reduce the fading memory and kernel function (principle 1), see Figure 12 in [49•]. Hence I propose that the expression of each principle is regulated by the brain for each area and developmental stage in a task dependent manner.

Altogether I have argued that the currently available experimental data provide useful guidance for understanding how cognition and behavior is implemented and regulated by networks of neurons in the brain. Marr and Poggio had proposed in [82] had proposed to distinguish three levels of models for brain computations:

- the computational (behavioral) level,
- the algorithmic level,
- the biological implementation level.

Whereas substantial work had focused on the interconnection of these three levels from the top down, more detailed data on the biological implementation level provide now also a basis for creating bottom-up connections. We have seen that each of the four constraints from the biological implementation level has significant implications for models on the algorithmic and computational level.

Conflict of interest

Nothing declared.

Acknowledgements

I would like to thank David Kappel, Robert Legenstein, and Zhaofei Yu for helpful comments, and Jonathan Wallis for sharing unpublished experimental data. This review was written under partial support the by the European Union project #604102 (Human Brain Project).

References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

- • of special interest
- •• of outstanding interest

1. Dupre C, Yuste R: **Calcium imaging reveals multiple conduction systems in hydra**. *Annual Meeting of the Society for Neuroscience*. 2015. Abstract 94.21.

2. • Kato S, Kaplan HS, Schrödel TS, Skora S, Lindsay TH, Yemini E, Lockery S, Zimmer M: **Global brain dynamics embed the motor command sequence of *Caenorhabditis elegans***. *Cell* 2015, **163**:656-669.
Related behavior to the dynamics of the recurrent neural network that represents the brain of *C. elegans*.

3. Portugues R, Feierstein CE, Engert F, Orger MB: **Whole-brain activity maps reveal stereotyped, distributed networks for visuomotor behavior**. *Neuron* 2014, **81**:1328-1343.

4. • Singer W: **Cortical dynamics revisited**. *Trends Cogn Sci* 2013, **17**:616-626.
Summarized experimental data, theories and open questions regarding network dynamics.

5. • Yuste R: **On testing neural network models**. *Nat Rev Neurosci* 2015, **16** 767-767.
Summarized experimental data, theories and open questions regarding network dynamics.

6. van Vreeswijk CA, Sompolinsky H: **Chaos in neuronal networks with balanced excitatory and inhibitory activity**. *Science* 1996, **274**:1724-1726.

7. Vogels TP, Rajan K, Abbott LF: **Neural network dynamics**. *Ann Rev Neurosci* 2005, **28**:357-376.

8. Gerstner W, Kistler WM, Naud R, Paninski L: *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press; 2014.

9. Harris KD, Shepherd GMG: **The neocortical circuit: themes and variations**. *Nat Neurosci* 2015, **18**:170-181.

10. Gupta A, Wang Y, Markram H: **Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex**. *Science* 2000, **287**:273-278.

11. Larsen RS, Sjöström PJ: **Synapse-type-specific plasticity in local circuits**. *Curr Opin Neurobiol* 2015, **35**:127-135.

12. Gjorgjieva J, Drion G, Marder E: **Computational implications of biophysical diversity and multiple timescales in neurons and synapses for circuit performance**. *Curr Opin Neurobiol* 2016, **37**:44-52.

13. Smolensky P: **Tensor product variable binding and the representation of symbolic structures in connectionist systems**. *Artif Intell* 1990, **46**:159-216.

14. • Maass W, Natschlaeger T, Markram H: **Real-time computing without stable states: a new framework for neural computation based on perturbations**. *Neural Computation* 2002, **14**:2531-2560.
Introduced the liquid computing model, separation property arising from diverse units, generic network computations for diverse readouts.

15. • Maass W, Natschlaeger T, Markram H: **Fading memory and kernel properties of generic cortical microcircuit models**. *J Physiol, Paris* 2004, **98**:315-330.
Analyzed fading memory and kernel property (nonlinear preprocessing) in generic networks of spiking neurons

16. Buonomano D, Maass W: **State-dependent computations: spatiotemporal processing in cortical networks**. *Nat Rev Neurosci* 2009, **10**:113-125.

17. • Jaeger H: **The ''echo state'' approach to analysing and training recurrent neural networks**. *GMD Report 148 — GMD — German National Research Institute for Computer Science*. 2001.
Introduced the echo state network model.

18. Legenstein R, Maass W: **Edge of chaos and prediction of computational performance for neural circuit models**. *Neural Netw* 2007, **20**:323-334.

19. Sussillo D, Abbott LD: **Generating coherent patterns of activity from chaotic neural networks**. *Neuron* 2009, **63**:544-557.

20. Hoerzer GM, Legenstein R, Maass W: **Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning**. *Cerebral Cortex* 2014, **24**:677-690.

21. • Maass W, Legenstein R, Bertschinger N: **Methods for estimating the computational power and generalization capability of neural microcircuits**. In *Advances in Neural Information Processing Systems*, vol 17. Edited by Saul LK, Weiss Y, Bottou L. MIT Press; 2005:865-872.
Introduced dimensionality analysis of network states for analyzing computational capabilities of a complex recurrent network.

22. Rigotti M, Barak O, Warden MR, Wang X-J, Daw NDD, Miller EK, Fusi S: **The importance of mixed selectivity in complex cognitive tasks**. *Nature* 2013, **497**:585-590.

23. Bishop CM: *Pattern Recognition and Machine Learning*. Springer; 2006.

24. Fusi S, Miller EK, Rigotti M: **Why neurons mix: high dimensionality for higher cognition**. *Curr Opin Neurobiol* 2016, **37**:66-74.

25. Olshausen BA, Field DJ: **How close are we to understanding V1?** *Neural Comput* 2005, **17**:1665-1699.

26. Chen JL, Carta S, Soldado-Magraner J, Schneider BL, Helmchen F: **Behaviour-dependent recruitment of long-range projection neurons in somatosensory cortex**. *Nature* 2013, **499**:336-340.

27. Maass W, Markram H: **On the computational power of circuits of spiking neurons**. *J Comp System Sci* 2004, **69**:593-616.

28. Dranias MR, Ju H, Rajaram E, Van Dongen AM: **Short-term memory in networks of dissociated cortical neurons**. *J Neurosci* 2013, **33**:1940-1953.

29. Marre O, Botella-Soler V, Simmons KD, Mora T, Tkačik G, Berry MJ: **High accuracy decoding of dynamical motion from a large retinal population**. *PLoS Comput Biol* 2015, **11**:e1004304.

30. Nikolic D, Haeusler S, Singer W, Maass W: **Distributed fading memory for stimulus properties in the primary visual cortex**. *PLoS Biol* 2009, **7**:1-19.

31. Klampfl S, David SV, Yin P, Shamma SA, Maass W: **A quantitative analysis of information about past and present stimuli encoded by spikes of A1 neurons**. *J Neurophysiol* 2012, **108**:1366-1380.

32. Goldman MS: **Memory without feedback in a neural network**. *Neuron* 2009, **61**:621-634.

33. Bernacchia A, Seo H, Lee D, Wang XJ: **A reservoir of time constants for memory traces in cortical neurons**. *Nat Neurosci* 2011, **14**:366-372.

34. Stokes MG: **'Activity-silent' working memory in prefrontal cortex: a dynamic coding framework**. *Trends Cogn Sci* 2015, **19**:394-405.

35. Haeusler S, Maass W: **A statistical analysis of information processing properties of lamina-specific cortical microcircuit models**. *Cereb Cortex* 2007, **17**:149-162.

36. Sussillo D, Toyoizumi T, Maass W: **Self-tuning of neural circuits through short-term synaptic plasticity**. *J Neurophysiol* 2007, **97**:4079-4095.

37. • Maass W, Joshi P, Sontag ED: **Computational aspects of feedback in neural circuits**. *PLoS Comput Biol* 2007, **3**:e165.
Introduced method and theoretical analysis for training readouts with feedback into a network to maintain persistent internal states, and to carry out context-dependent computations.

38. Mante V, Sussillo D, Shenoy KV, Newsome WT: **Context-dependent computation by recurrent dynamics in prefrontal cortex**. *Nature* 2013, **503**:78-84.

39. Jaeger H, Haas H: **Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication**. *Science* 2004, **304**:78-80.

40. Thalmeier D, Uhlmann M, Kappen HJ, Memmesheimer RM. *Learning Universal Computations with Spikes*. 2015; arXiv preprint arXiv:150507866.

41. Luczak A, McNaughton BL, Harris KD: **Packet-based communication in the cortex**. *Nat Rev Neurosci* 2015, **16**: 745-755.

42. • Bathellier B, Ushakova L, Rumpel S: **Discrete neocortical dynamics predict behavioral categorization of sounds**. *Neuron* 2012, **76**:435-449.
Demonstrated that large numbers of diverse auditory inputs cause activation of a variation of one of a very small number of assemblies in the primary auditory cortex of rodents.

43. • Miller JEK, Ayzenshtat I, Carrillo-Reid L, Yuste R: **Visual stimuli recruit intrinsically generated cortical ensembles**. *Proc Natl Acad Sci* 2014, **111**:E4053-E4061.
Demonstrated that neural activity in the primary visual cortex of rodents is dominated by variations of a small number of spatio-temporal patterns.

44. • Luczak A, Barthó P, Harris KD: **Spontaneous events outline the realm of possible sensory responses in neocortical populations**. *Neuron* 2009, **62**:413-425.
Proposed a model for network activity where only a small fraction of possible spatio-temporal patterns occurs spontaneously, and in response to sensory stimuli.

45. Sadovsky AJ, MacLean JN: **Mouse visual neocortex supports multiple stereotyped patterns of microcircuit activity**. *J Neurosci* 2014, **34**:7769-7777.

46. • Fujisawa S, Amarasingham A, Harrison MT, Buzsaki G: **Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex**. *Nat Neurosci* 2008, **11**:823-833.
Demonstrated that learned behaviors are encoded by assembly sequences in the prefrontal cortex of rodents.

47. • Harvey CD, Coen P, Tank DW: **Choice-specific sequences in parietal cortex during a virtual-navigation decision task**. *Nature* 2012, **484**:62-68.
Demonstrated that learned behaviors are encoded by assembly sequences in the parietal cortex of rodents.

48. Izhikevich EM, Gally JA, Edelman GM: **Spike-timing dynamics of neuronal groups**. *Cereb Cortex* 2004, **14**:933-944.

49. • Klampfl S, Maass W: **Emergence of dynamic memory traces in cortical microcircuit models through STDP**. *J Neurosci* 2013, **33**:11515-11529.
Introduced a model for generating stimulus-dependent assembly sequences through STDP in networks of spiking neurons, and studied computational consequences.

50. Pokorny C, Maass W: *Emergence of Assembly Sequences through STDP in Data-based Cortical Microcircuit Models*. 2016:. (in preparation).

51. Litwin-Kumar A, Doiron B: **Formation and maintenance of neuronal assemblies through synaptic plasticity**. *Nat Commun* 2014:5.

52. Hebb DO: *The Organization of Behavior: A Neuropsychological Approach*. John Wiley & Sons; 1949.

53. • Buzsaki G: **Neural syntax: cell assemblies, synapsembles, and readers**. *Neuron* 2010, **68**:362-385.
Reviewed results on behavioral and cognitive correlates of assembly activations, and examined how cognitive computations with assemblies could be organized in the brain.

54. Branco T, Staras K: **The probability of neurotransmitter release: variability and feedback control at single synapes**. *Nat Rev Neurosci* 2009, **10**:373-383.

55. Kavalali ET: **The mechanisms and functions of spontaneous neurotransmitter release**. *Nat Rev Neurosci* 2015, **16**:5-16.

56. Maass W: **Noise as a resource for computation and learning in networks of spiking neurons**. *Special Issue of the Proc of the IEEE on 'Engineering Intelligent Electronic Systems based on Computational Neuroscience'* 2014, **102**:860-880.

57. Leopold DA, Logothetis NK: **Multistable phenomena: changing views in perception**. *Trends Cogn Sci* 1999, **3**:254-264.

58. • Jezek K, Henriksen EJ, Treves A, Moser EI, Moser M: **Theta-paced flickering between place-cell maps in the hippocampus**. *Nature* 2011, **478**:246-249.
Demonstrated that ambiguous information about the environment causes relatively fast flickering between corresponding network states in the rodent hippocampus.

59. • Rich EL, Wallis JD: **Decoding subjective decisions from orbitofrontal cortex**. *Nat Neurosci* 2016. (in press).
Demonstrated that choosing between different options is related to fast flickering between corresponding network states in monkey OFC.

60. • Berkes P, Orban G, Lengyel M, Fiser J: **Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment**. *Science* 2011, **331**:83-87.
Introduced the paradigm to analyze the statistical distribution of joint network states defined by simultaneous activity of neurons within a small time window.

61. • Buesing L, Bill J, Nessler B, Maass W: **Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons**. *PLoS Comput Biol* 2011, **7**:e1002211.
Introduced a principled model for stochastic computation in networks of spiking neurons ('neural sampling').

62. Pecevski D, Buesing L, Maass W: **Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons**. *PLoS Comput Biol* 2011, **7**:e1002294.

63. Pecevski D, Maass W: **Learning probabilistic inference through STDP**. *eNeuro* 2016, in press.

64. Knill DC, Kersten D: **Apparent surface curvature affects lightness perception**. *Nature* 1991, **351**:228-230.

65. • Habenschuss S, Jonke Z, Maass W: **Stochastic computations in cortical microcircuit models**. *PLoS Comput Biol* 2013, **9**:e1003311.
Analyzed conditions under which networks of spiking neurons consisting of complex and diverse units can carry out probabilistic inference and solve constraint satisfaction problems through stochastic computation (sampling).

66. Savin C, Deneve S: **Spatio-temporal representations of uncertainty in spiking neural networks**. *Adv Neural Inform Process Syst* 2014:2024-2032.

67. Legenstein R, Maass W: **Ensembles of spiking neurons with noise support optimal probabilistic inference in a dynamically changing environment**. *PLoS Comput Biol* 2014, **10**:e1003859.

68. Aarts E, Korst J: *Simulated Annealing and Boltzmann Machines*. 1988.

69. Tversky A, Kahneman D: **Judgment under uncertainty: heuristics and biases**. *Science* 1974, **185**:1124-1131.

70. Gigerenzer G, Hoffrage U, Kleinbölting H: **Probabilistic mental models: a Brunswikian theory of confidence**. *Psychol Rev* 1991, **98**:506.

71. Tenenbaum JB, Kemp C, Griffiths TL, Goodman ND: **How to grow a mind: statistics, structure, and abstraction**. *Science* 2011, **331**:1279-1285.

72. Pouget A, Beck JM, Ma WJ, Latham PE: **Probabilistic brains: knowns and unknowns**. *Nat Neurosci* 2013, **16**:1170-1178.

73. Meyniel F, Sigman M, Mainen ZF: **Confidence as Bayesian probability: from neural origins to behavior**. *Neuron* 2015, **88**:78-92.

74. • Holtmaat A, Svoboda K: **Experience-dependent structural synaptic plasticity in the mammalian brain**. *Nat Rev Neurosci* 2009, **10**:647-658.
Demonstrated continuously ongoing rewiring of networks of neurons in the brain through spine dynamics.

75. Stettler DD, Yamahachi H, Li W, Denk W, Gilbert CD: **Axons and synaptic boutons are highly dynamic in adult visual cortex**. *Neuron* 2006, **49**:877-887.

76. Loewenstein Y, Yanover U, Rumpel S: **Predicting the dynamics of network connectivity in the neocortex**. *J Neurosci* 2015, **35**:12535-12544.

77. Spitzer NC: **Neurotransmitter switching? No surprise**. *Neuron* 2015, **86**:1131-1144.

78. • Ziv YLD, Burns LD, Cocker ED, Hamel EO, Ghosh KK, Kitch LJ, Gamal AE, Schnitzer MJ: **Long-term dynamics of CA1 hippocampal place codes**. *Nat Neurosci* 2013, **16**:264-266.
Demonstrated inherent drift of neural codes on the time scale of weeks.

79. • Kappel D, Habenschuss S, Legenstein R, Maass W: **Network plasticity as Bayesian inference**. *PLoS Comput Biol* 2015, **11**:e1004485.
Introduced a theoretical framework for analyzing the dynamics of network configurations under stochastic spine dynamics and synaptic plasticity ('synaptic sampling').

80. MacKay DJ: **Bayesian interpolation**. *Neural Comput* 1992, **4**:415-447.

81. • Marder E, Goaillard JM: **Variability, compensation and homeostasis in neuron and network function**. *Nat Rev Neurosci* 2006, **7**:563-574.
Demonstrated that networks of neurons achieve stable computational function with many different parameter settings, and are able to move between them in response to external perturbations.

82. Marr D, Poggio T: *From Understanding Computation to Understanding Neural Circuitry*. MIT Tech Report; 1976.

83. Maass W, Markram H: **Theory of the computational function of microcircuit dynamics**. *The Interface between Neurons and Global Brain Function*. MIT Press; 2006:. pp 371–390.