Searching for Principles of Brain Computation

Wolfgang Maass
Institut für Grundlagen der Informationsverarbeitung
Technische Universität Graz, Austria

Structure of my talk

1. Problems that we are facing, and how to overcome them
2. Four principles (constraints) of brain computation
3. Visions for future work, and open problems

1. Problems that we are facing, and how to overcome them

How can theoretical neuroscience become more of a „science“ ?

- Paradigm for a really successful theoretical science: Theoretical physics

- Characteristic features of theoretical physics

--ongoing debates between opposing camps

--strong interest in new experimental data

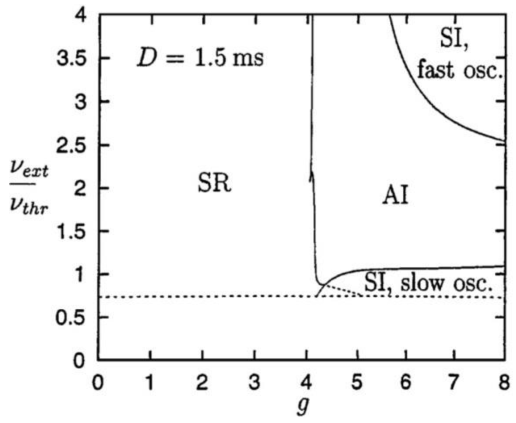
--theory aims to be falsifiable

--falsification of theoretical predictions has impact on theory

A quick case study of theory and experimental data in computational neuroscience:

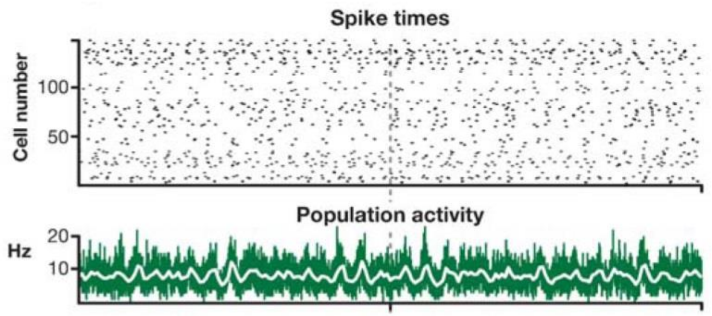
What firing regimes of neural circuits are most suitable for computations?

The AI (asynchronous irregular) firing regime was proposed to be suitable for neural computation



Brunel 2000, Journal of Computational Neuroscience,, 2000.

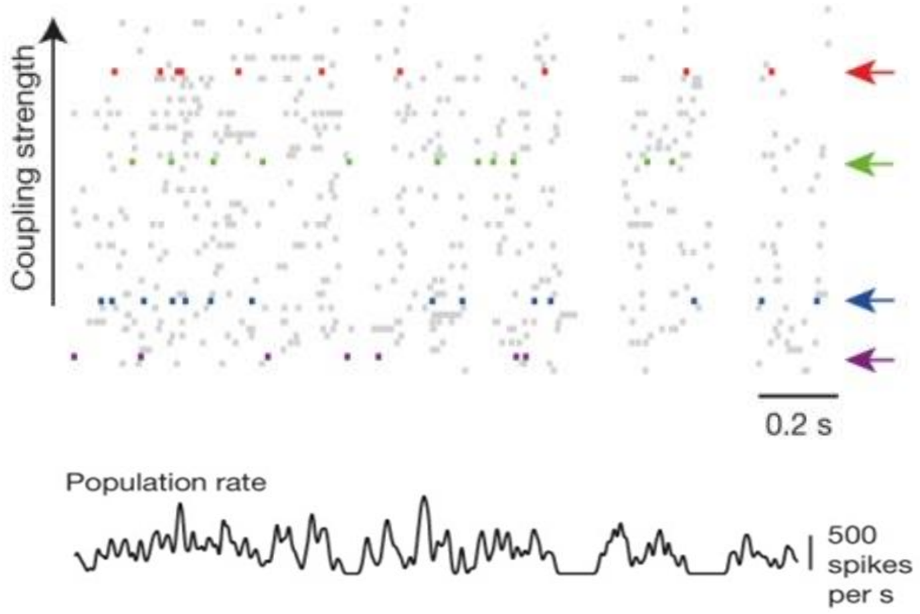
And a large number of models and theory studies investigated how such AI firing regime can be produced



Vogels et al. 2005, Annu. Rev. Neurosci. , 2005.

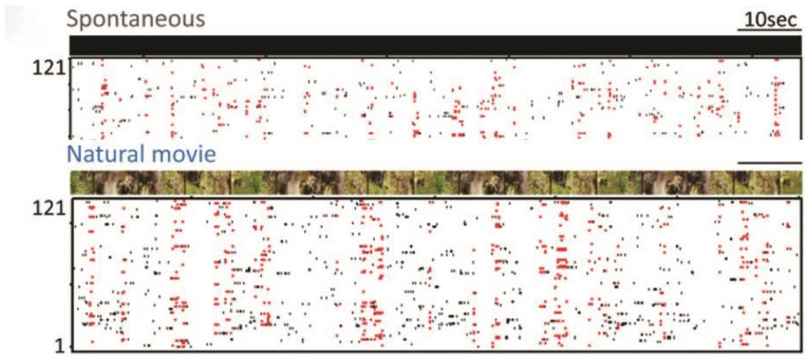
But: Virtually all simultaneous recordings from many neurons suggest that neural circuits operate in a regime where spatio-temporal firing patterns dominate

spikes recorded through multi-electrodes



Okun et al. 2015, Nature 2015

firing rates on a larger time scale (Ca-imaging



Miller et al. , PNAS 2014.

Is theoretical neuroscience addressing this discrepancy between theoretically driven models and experimental data?

What types of models/analyses are needed?

(Marr and Poggio, 1976) suggested to analyze neural computation on 3 different levels:

- *computational (behavioural) level* (what needs to be computed?)
- *algorithmic level* (how is that computed?)
- *implementation level* (how can biological neural networks implement that?)

In (Poggio, The levels of understanding framework revisited, 2012)
he suggested to add two further levels of analysis at the top:

- *evolution*
- *learning and development*

In addition he suggested that in view of rich data in computational neuroscience one should now focus on connections between the levels, and also to proceed bottom-up.

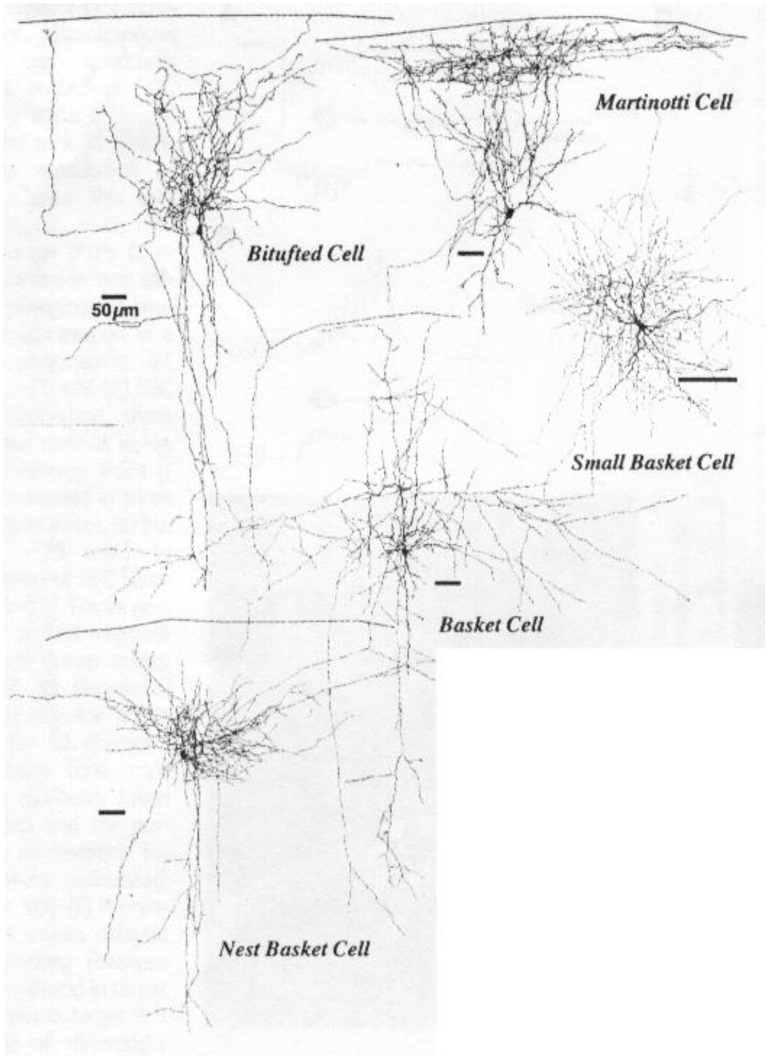
2. Four Principles (Constraints) of Brain Computation

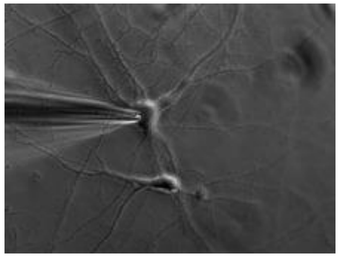
Principle 1: Neural circuits are highly recurrent networks of neurons and synapses with diverse dynamic properties

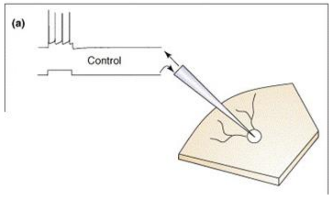
Note that already the evolutionary oldest neural circuits (e.g. in hydra, C-elegans) were highly *recurrent* networks, whereas we as theoreticians usually prefer to think in terms of feedforward networks.

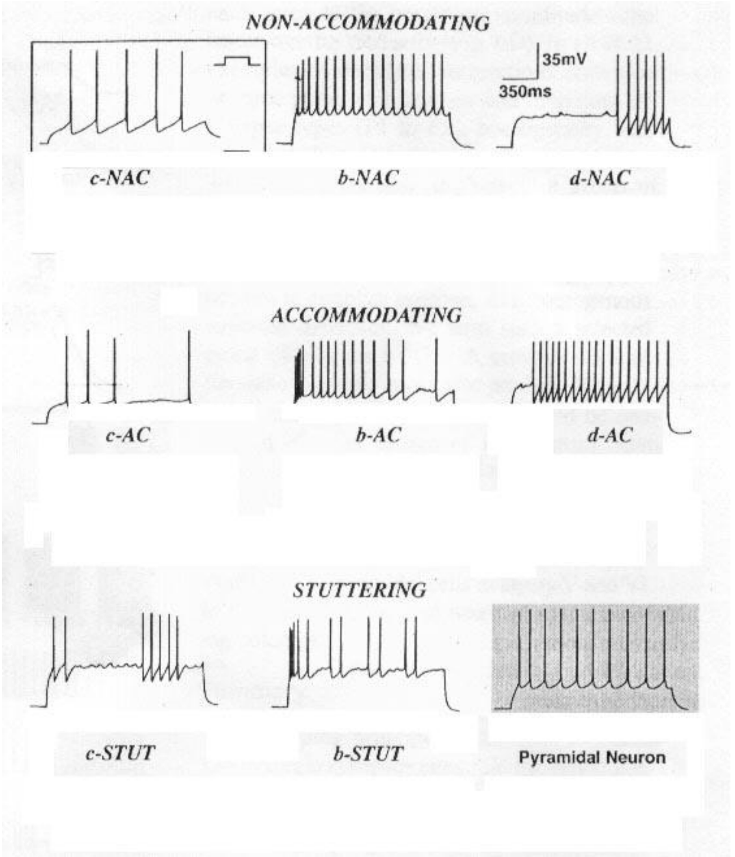
There are many different types of neurons that exhibit diverse temporal dynamics:

The same input (here a step current) causes different responses in different types of neurons









Short term dynamics of synapses

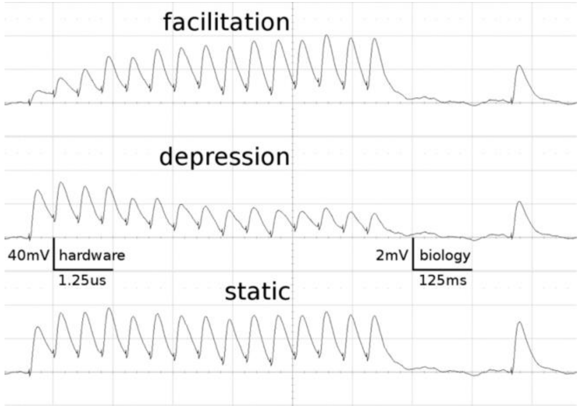
Every synapse has a complex inherent temporal dynamics

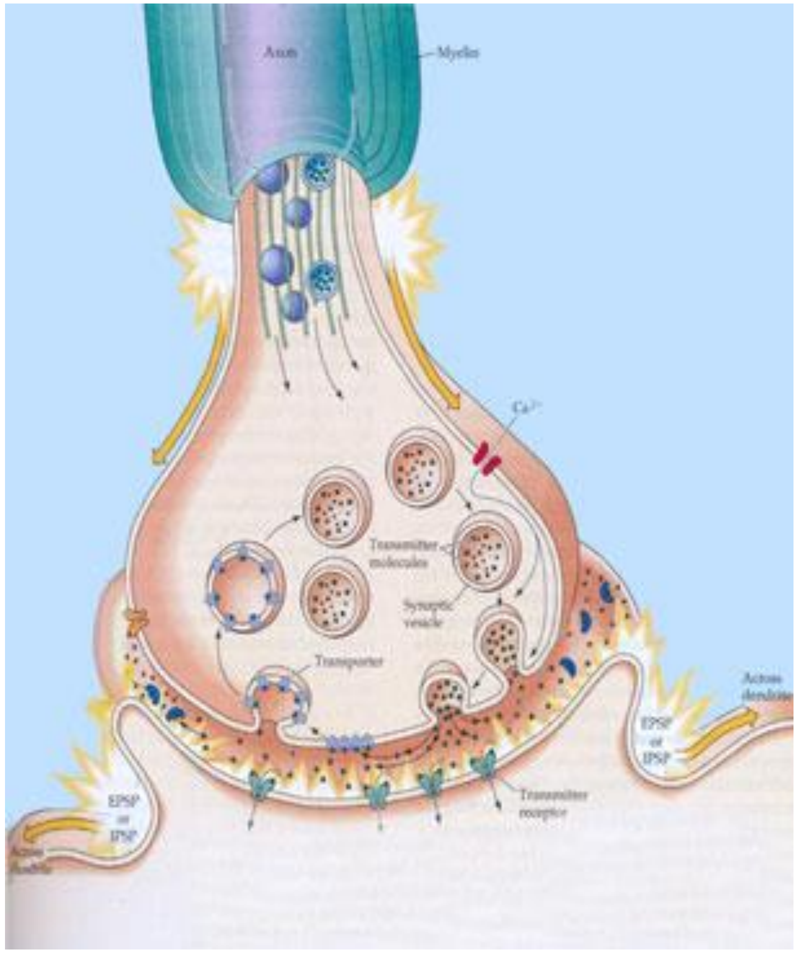
(and can NOT be modeled by a single parameter w like in artificial neural networks).

Model for a dynamic synapse with parameters w, U (release probability, D(time constant for depression), F (time constant for facilitation) according to [Markram, Wang, Tsodyks, PNAS 1998]:

The amplitude A_k of the postsynaptic potential for the k^{th} spike in a spike train with inter-spike intervals $\Delta_1, \Delta_2,\ldots,\Delta_{k-1}$ is modeled by the equations

$$A_k = w \cdot u_k \cdot R_k$$
$$u_k = U + u_{k-1}(1-U)\exp(-\Delta_{k-1}/F)$$
$$R_k = 1 + (R_{k-1} - u_{k-1}R_{k-1} - 1)\exp(-\Delta_{k-1}/D)$$



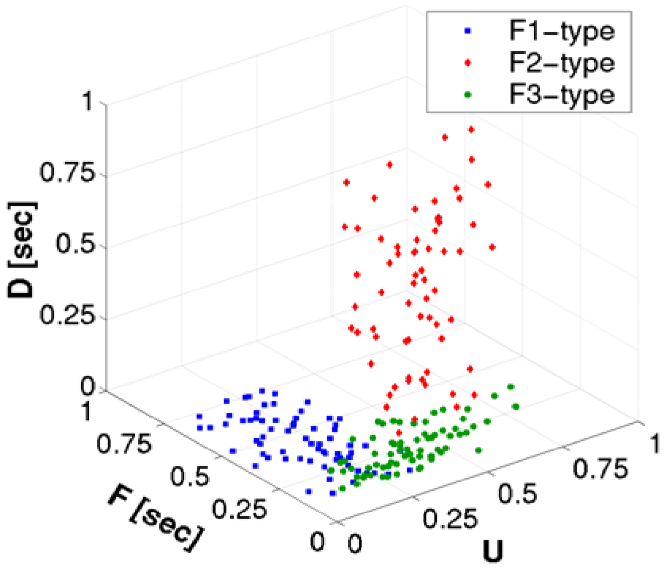


The parameters U, D, F are different for different synapses

Empirically found distributions are reported in

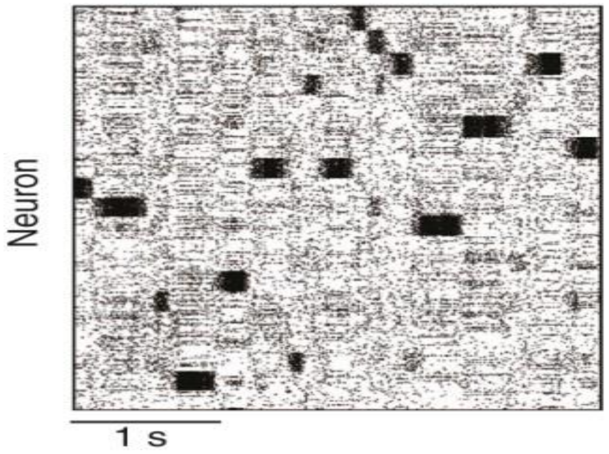
H. Markram, Y. Wang, and M. Tsodyks, Differential signaling via the same axon of neocortical pyramidal neurons, *PNAS* 95, 5323 – 5328, 1998.

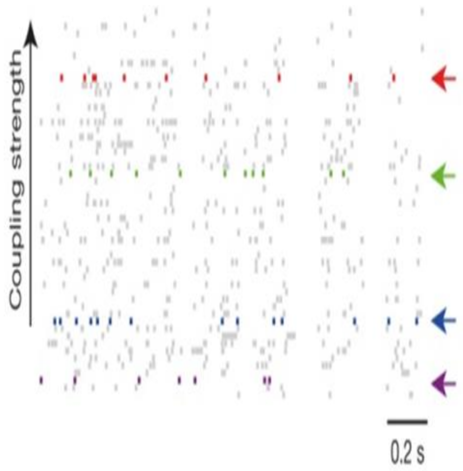
A. Gupta, Y. Wang, and H. Markram, Organizing principles for a diversity of GABAergic interneurons and synapses in the neocortex, *Science* 287, 273 – 278, 2000.



I will return later to the experimentally found relatively low values of the release probability U for the first spike.

Consequence: **Network activity patterns in theory-driven models tends to differ strongly from experimentally observed ones**



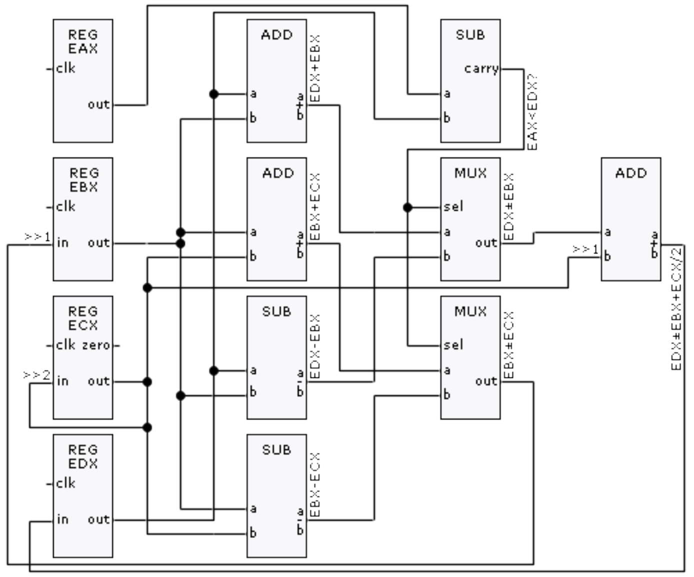


model

data

A. Litwin-Kumar and B. Doiron. Nature Communications, , 2014.

These data suggest that neural computation in the brain has a different organization than computations in digital circuits, artificial neural networks. networks of neuroids, etc



In fact; I would lbe willing to bet that one *cannot* simulate computations of digital circuits, artificial neural networks, or neuroid networks with reasonably realistic models for recurrent networks of biological neurons and synapses.

Note that elimination of noise by averaging over several parallel copies of a circuit would require „parameter sharing“, which is questionable in biological networks

Principle 2: Neural computation needs to serve diverse „neural users“

(which extract samples of high-D network states, and are adaptive)

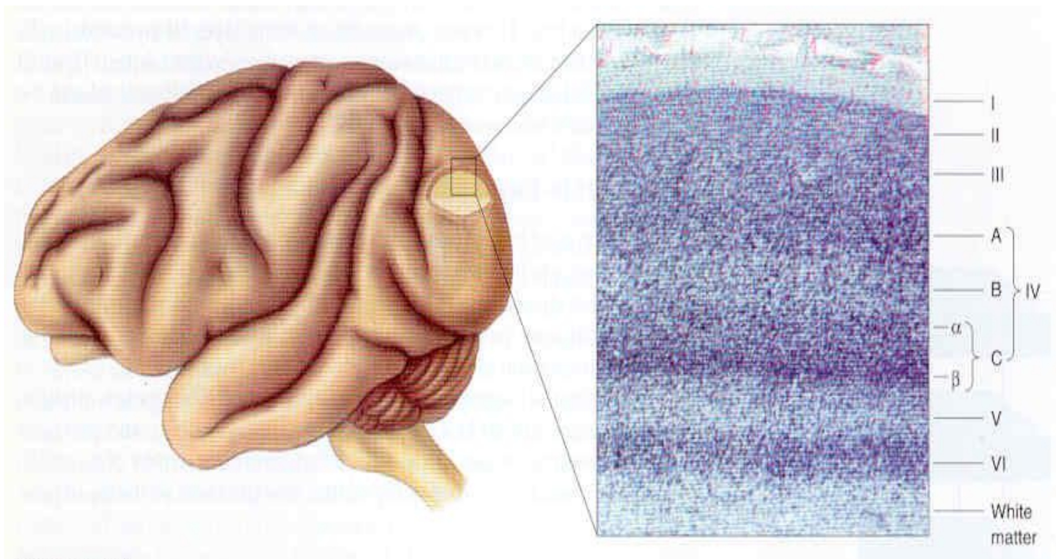


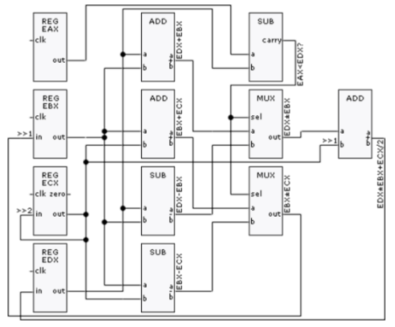
Figure 10.12
The cytoarchitecture of the striate cortex. The tissue has been Nissl stained to show cell bodies, which appear as dots. (Source: Adapted from Hubel, 1988; p. 97.)

Neural users are numerous different downstream neural systems, to which projection neurons on superficial and deep layers project.

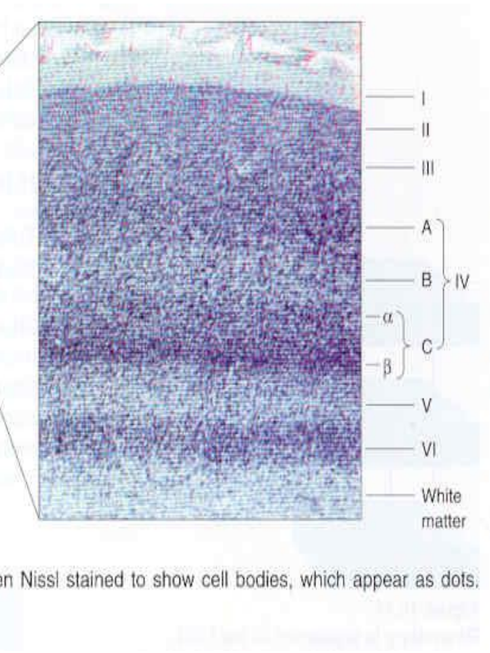
These projection neurons extract high-D samples from the network activity.

Their synapses are subject to longterm plasticity.

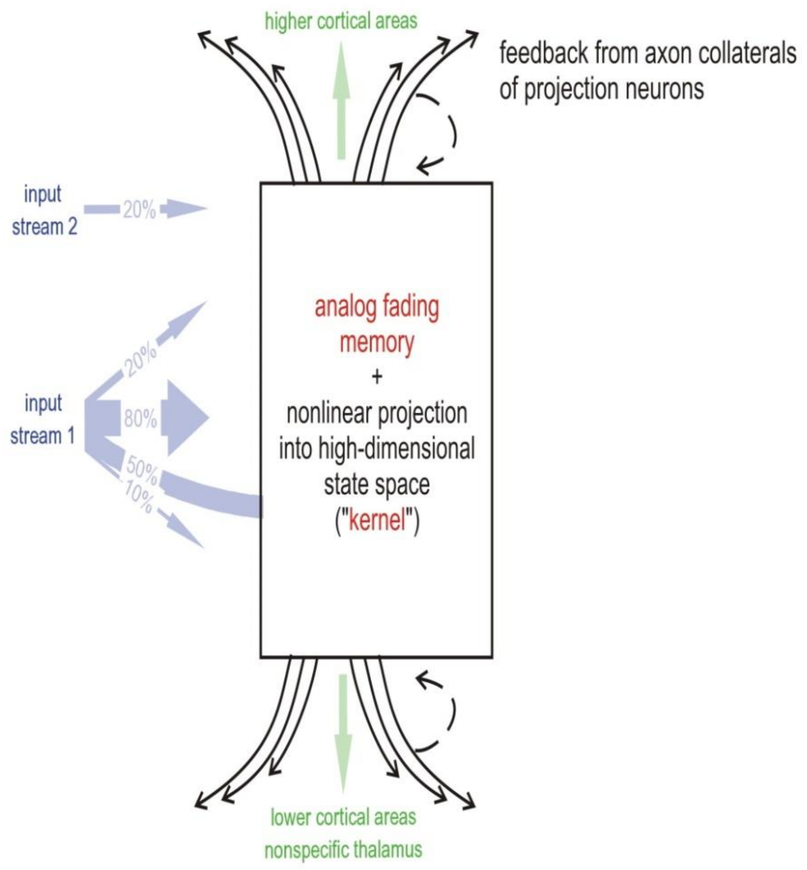
Consequence: When thinking about computations in a cortical column, we should analyze its sequence of high-D „**network states**“.



What computational operations within a column are suggested by this perspective?



en Nissl stained to show cell bodies, which appear as dots.



Two candidates:

- Integration of incoming information over time
- Nonlinear projection of this information into the high-D space of network states

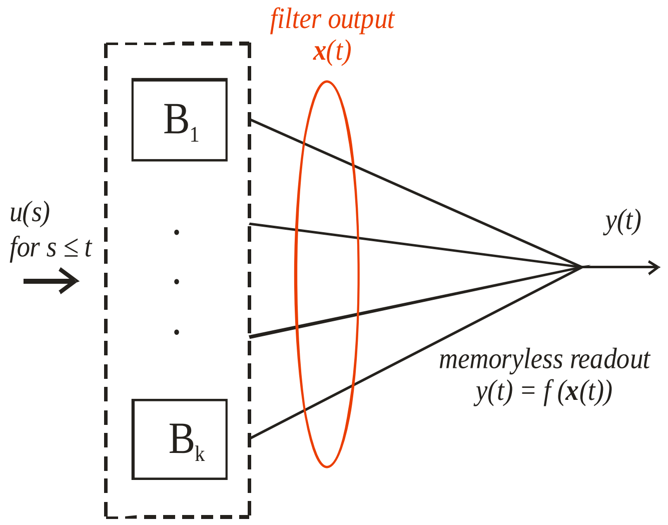
Diversity of neurons and synapses could support temporal integration of information over time

Theorem (Maass, Natschläger, Markram, 2002) ,based on (Boyd and Chua, 1985):

Any time-invariant filter with fading memory can be approximated with any degree of precision by this simple computational model

- ***if*** there is a rich enough pool **B** of basis filters (time invariant, with fading memory) from which the basis filters $B_1, \ldots, B_k$ in the filterbank can be chosen (**B** needs to have the pointwise separation property) *and*

- ***if*** any continuous bounded function can be approximated by some readout

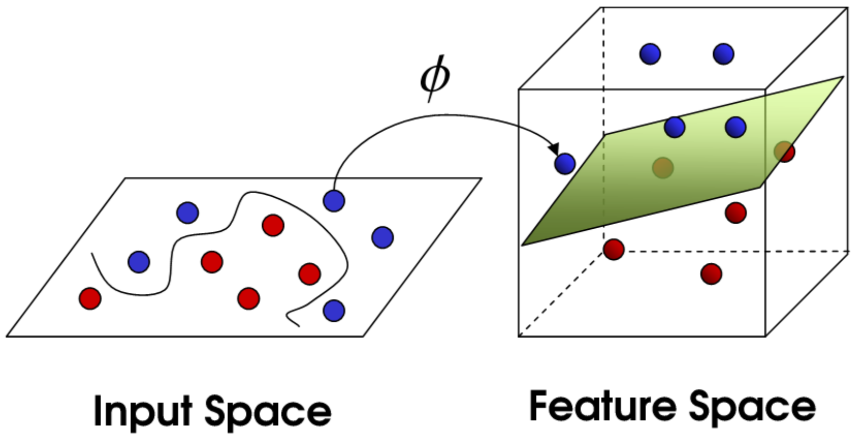


Def: A class **B** of basis filters has the pointwise separation property if there exists for any two input functions $u(\bullet), v(\bullet)$ with $u(s) \neq v(s)$ for some $s \le t$ a basis filter $B \in$ **B** with $(Bu)(t) \neq (Bv)(t)$.

Open problem: Can theory provide further insight into the functional role of diverse computational units in a recurrent network?

Boosting the computational power of linear readouts (projection neurons) through generic nonlinear projections into high-D spaces

This principle is well-known from Machine Learning (kernels of Support Vector Machines):



Note that no concrete nonlinear operations, such as multiplication, are needed for that:

It suffices if different inputs to the kernel (or cortical column) are mapped onto linearly independent output vectors.

A simple demo for this style of network computation

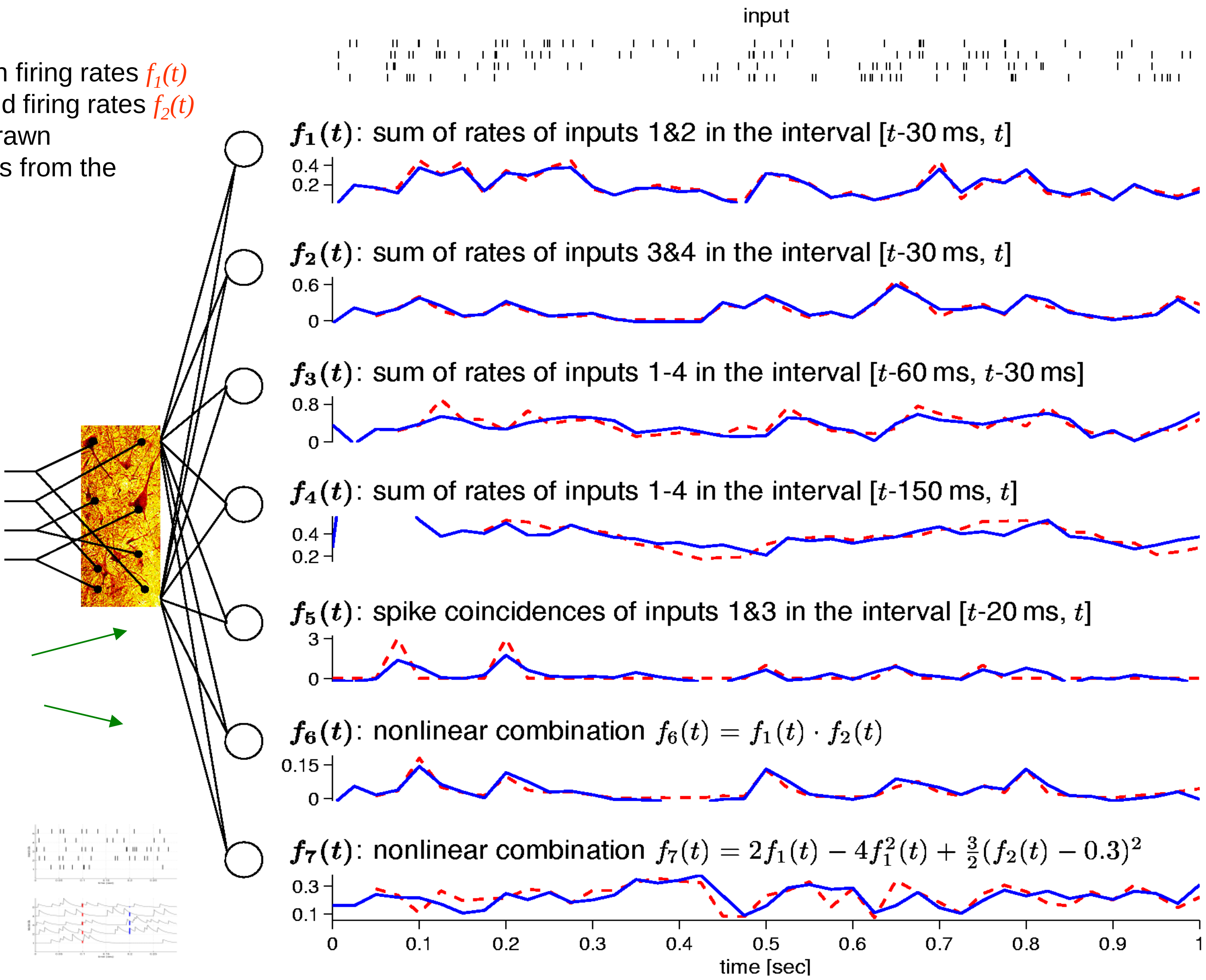
(Maass, Natschlaeger, Markram, 2004)

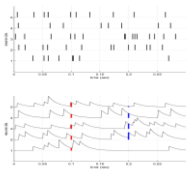
Network input:

4 Poisson spike trains with firing rates $f_1(t)$ for spike trains 1 and 2 and firing rates $f_2(t)$ for spike trains 3 and 4, drawn independently every 30 ms from the interval [0, 80] Hz

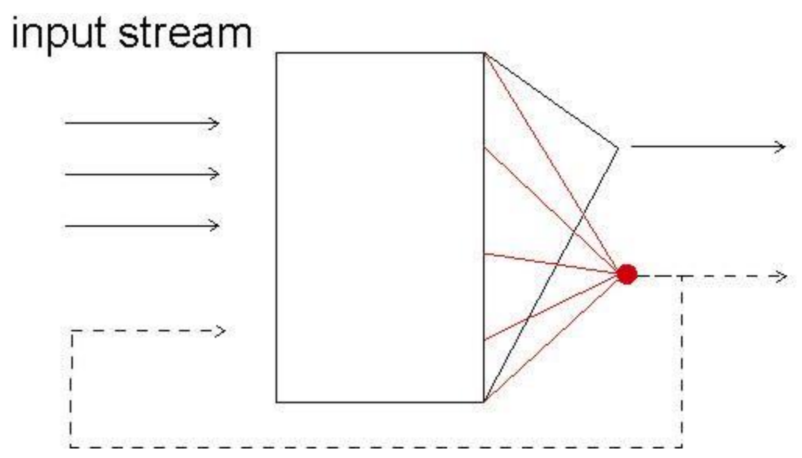
Randomly connected network of 135 spiking neurons with dynamic synapses:

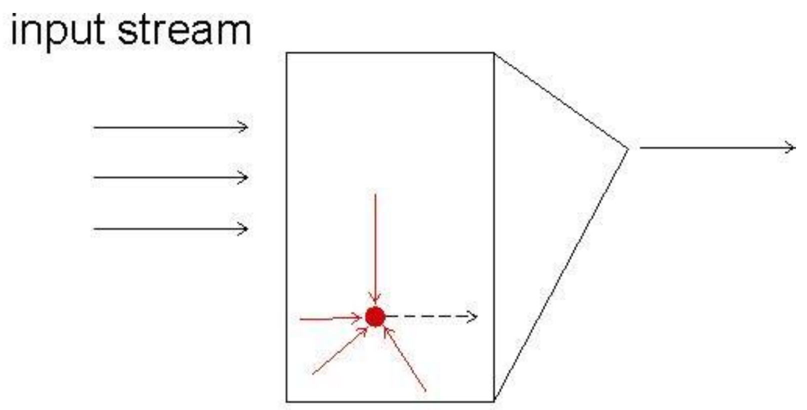
7 linear readouts, trained for 7 different tasks by linear regression (blue traces) receive EPSPs from the 135 network neurons





The theoretical computational power of the model makes a qualitative jump if one allows feedback from trained readouts





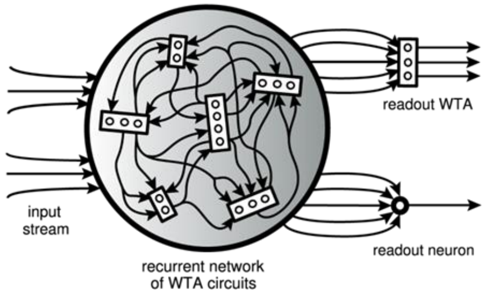
Training a readout neuron with feedback is equivalent to training a neuron **within** the neural circuit.

Theorem: If one allows feedback from readout neurons back into the circuit, and if a readout neuron can learn to compute any continuous function, then this model becomes universal for analog (and digital) computation on input streams.

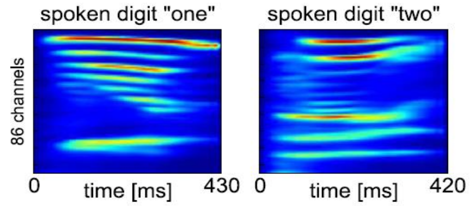
[Maass, Joshi, Sontag, PLOS Comp. Biol. 2007

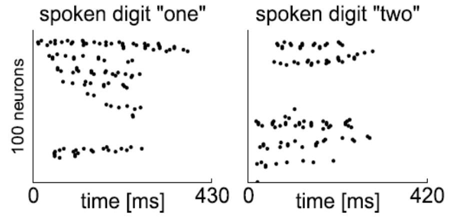
Additional effect if one applies STDP to all (or many) synaptic connections between excitatory neurons:

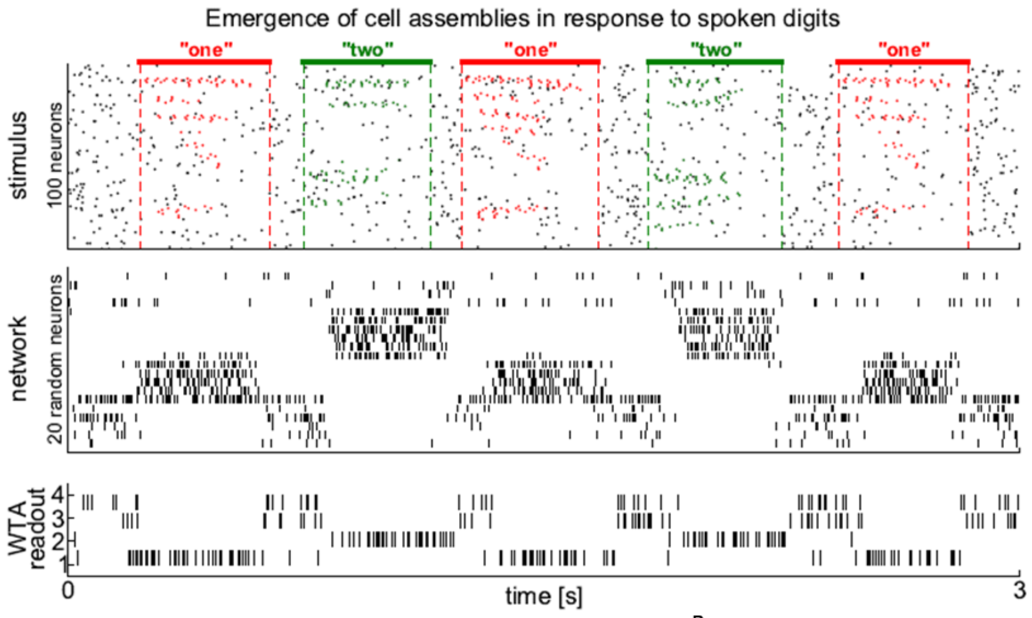
The network responds with stereotypical spatio-temporal patterns to repeating input patterns



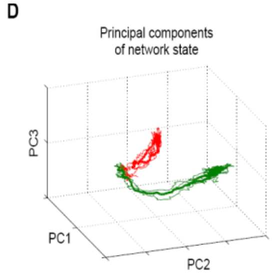
Input patterns







For digit classification It is no longer necessary to train a readout through supervised learning: A downstream WTA circuit learns autonomously to classify the two spoken digits without supervision



S. Klampfl and W. Maass. Emergence of dynamic memory traces in cortical microcircuit models through STDP. J. of Neuroscience, 2013

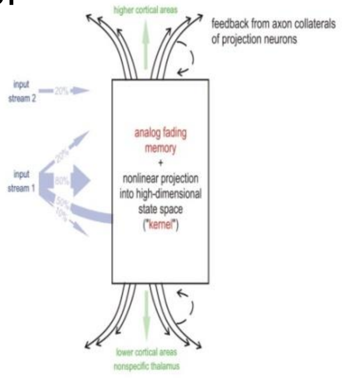
G. Griesbacher, W. Maass, in preparation

Experimental data support the resulting computational model (generic preprocessing for diverse readouts)

1. Temporal integration of information and nonlinear projection into high D

D. Nikolic, S. Haeusler, W. Singer, and W. Maass. Distributed fading memory for stimulus properties in the primary visual cortex. PLoS Biology, 2009

S. Klampfl, S. V. David, P. Yin, S. A. Shamma, and W. Maass. A quantitative analysis of information about past and present stimuli encoded by spikes of A1 neurons. J. of Neurophys., 2012



2. Multiplexing of "neural codes" in the network for different tasks:

Rigotti, M., Barak, O., Warden, M. R., Wang, X. J., Daw, N. D., Miller, E. K., & Fusi, S. The importance of mixed selectivity in complex cognitive tasks. *Nature*, *2013*

Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T.. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 2013

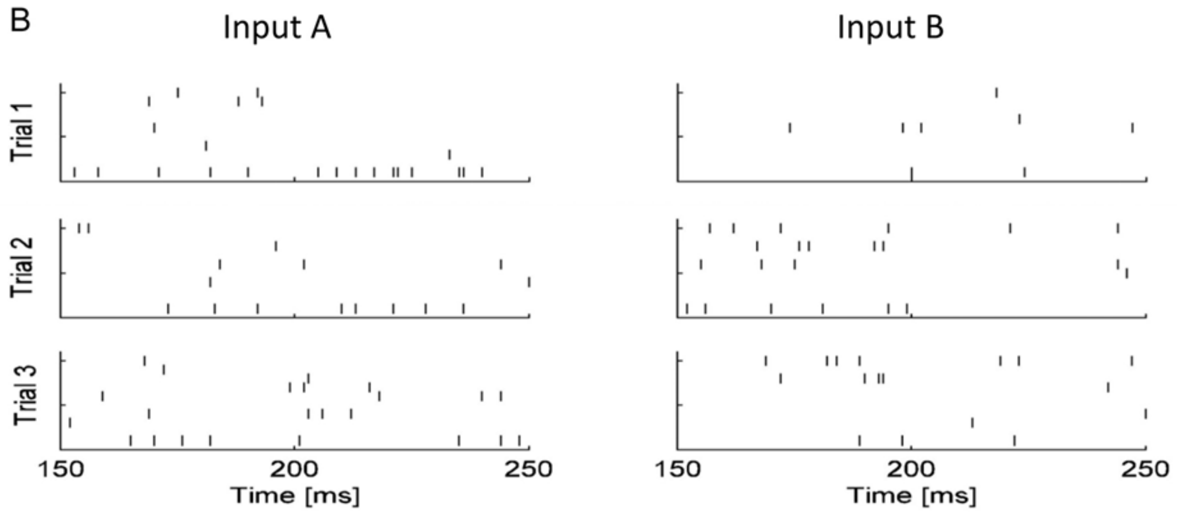
3. Diversity of neural readouts from the same cortical column:

Chen, J. L., Carta, S., Soldado-Magraner, J., Schneider, B. L., & Helmchen, F.. Behaviour-dependent recruitment of long-range projection neurons in somatosensory cortex. *Nature* 2013.

Principle 3: Brain computations are subject to high trial-to-trial variability ("noise")

Substantial rial-to-trial variability is observed in the brain at virtually all spatial and temporal scales.

Example: Variability of spike responses in area V1 of cat: each column shows 3 trials with the same stimulus



[Nikolic, Haeusler, Singer, Maass, PLoS Biol. 2009]

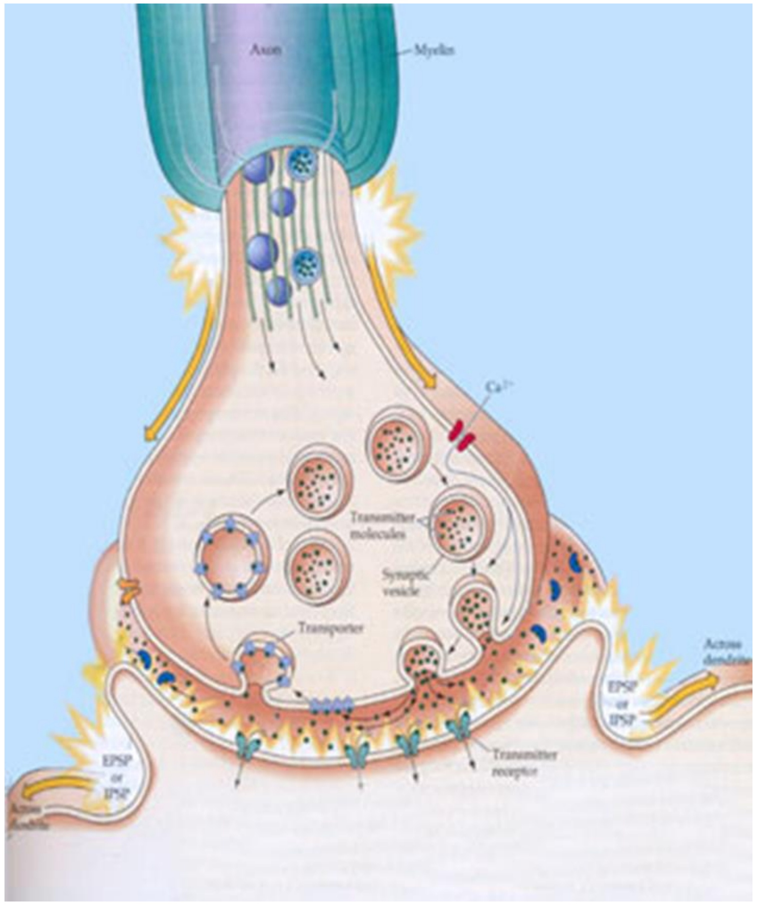
A major source of variability in neural circuits: probabilistic vesicle release at synapses

Common estimates of release probability of a vesicle in response to a presynaptic spike are around **0.5** (for neocortex), see e.g.

(Branco, Staras, Nat. Rev. in Neurosci, 2009)

In addition vesicles are frequently released without a presynaptic spike

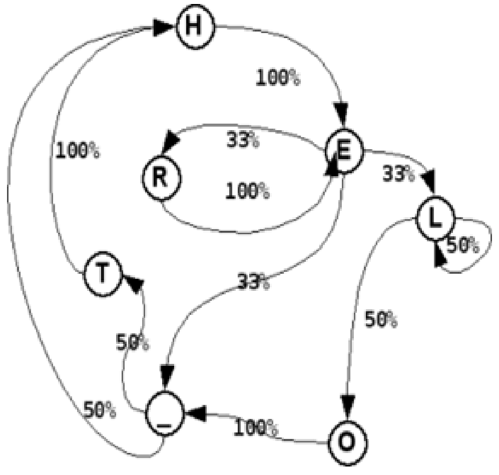
(Kavalali, Nat. Rev. in Neurosci., 2015)



How can one compute with stochastic neural systems?

It would be difficult to emulate deterministic computational models without biologically unrealistic averaging over duplicate copies of the circuit.

Markov chains (MCs) are stochastic systems that are commonly used in computer science and machine learning (simulated in software)



Key property of MCs (used e.g. for Google page rank): Under some mild assumptions they have a unique **stationary distribution** p of network states, to which they converge from any initial state.

I will discuss two types of computational applications of MCs for networks of spiking neurons:

- solving constraint satisfaction problems
- probabilistic inference

A common type of MC: Boltzmann machines (BMs)
Useful in theory and applications, but biologically unrealistic

- This type of MC is commonly used in machine learning (e.g. for „deep learning“) and for solving constraint satisfaction problems
- BMs are stochastic artificial neural networks, whose units output 1 or 0, with stochastic switches according to some global schedule:

  When unit i is allowed to switch, it assumes

  $$x_i = 1 \; with \; probability \; \sigma\left(\frac{1}{T}\left(\sum_i w_{ij} x_j + b_i\right)\right), \quad \text{else} \quad x_i = 0$$

  for the common sigmoidal activation function $\sigma(x) = 1/(1+e^{-x})$

- The state of a BM with N units is a bit vector of length N

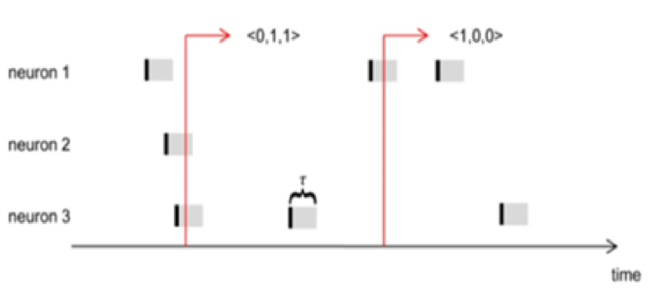
.

- Every Boltzmann distribution (i.e., distribution over binary random variables with at most 2nd order dependencies) is the stationary distribution p of some BM.

- The stochastic dynamics of BMs is equivalent to Gibbs sampling (which is frequently used for probabilistic inference in ML: „MCMC sampling“)

Theoretical results I

- For every BM with N units there is a network of N spiking neurons (SNN) that has the same stationary distribution p of network states , where one uses a standard way of converting spikes to bits:



- Spiking neuron model: Instantaneous firing probability $\rho_k(t) = \frac{1}{\tau} exp(u_k(t))$ for a standard definition of the membrane potential $u_k(t) = \sum_{i=1}^{n} w_{ki}\tilde{y}_i(t) + w_{k0}$

- But the corresponding SNNs have a different stochastic dynamics, since BMs are reversable MCs, SNN are non-reversable MCs

- Consequence for theory: One needs to replace the „detailed balance" condition for BMs by the „neural computability condition" for SNNs in order to construct SNNs that have a given stationary distribution p

(Büsing, Bill, Nessler, Maass. PLOS Comp. Biol. 2011)

.

Theoretical results II

- For the case with **symmetric** weights one can characterize the stationary distribution p_C of a SNN C (like for a BM) through its energy function:

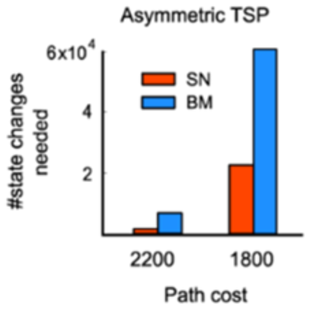
$$p_C(\mathbf{z}) = \frac{1}{Z} \cdot exp(-E\{\mathbf{z}\}/T)$$

with $E\{\mathbf{z}\} = -\sum_{i<j} w_{ij} z_i z_j - \sum_i b_i z_i$

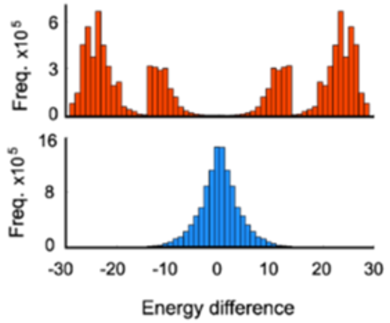
- This provides a new method for constructing SNNs that can solve specific computational tasks:

  One first constructs an energy function that assigns lowest energy to good solutions of a computational problem (e.g., TSP, SAT, SUDOKU, ...)

- The resulting SNN finds often solutions faster (i.e., with fewer state changes) than a BM with the same energy function: and temperature. Example for the TSP:



Reason: spiking neurons overcome faster energy barriers



- One can also engage network motifs with asymmetric weights.

(Zonke, Habenschuss, Maass. Arxiv 2015)

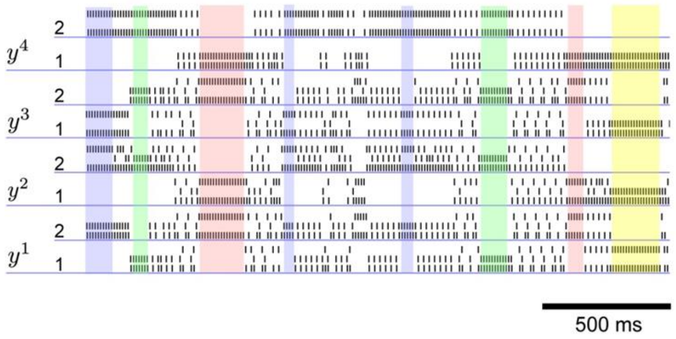
Theoretical results III

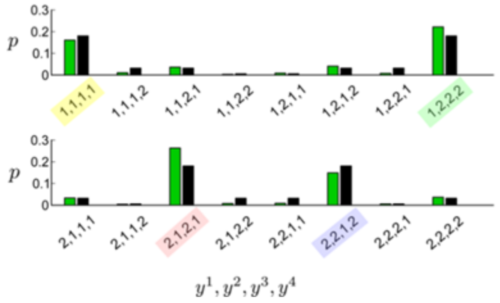
- Using auxiliary spiking neurons (and asymmetric weights) SNNs can learn through STDP any distribution p over discrete random variables, also with higher order dependencies

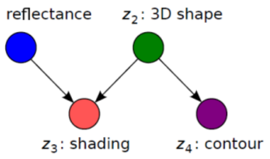
- More precisely, a suitable SNN can build through STDP an internal models for such given distribution p, just by processing examples that are drawn from p

- In this way, SNNs can acquire through learning really complex knowledge
- They can extract information from this knowledge base through probabilistic inference (through sampling)

Example: Learning probabilistic inference with „explaining away“ for a visual cognition task (Knill, Kersten, Nature 1991)







(Pecevski, Maass, 2015 (under review)

Challenge for future work: Move models for stochastic computation closer to biological data

- The previous sketched paradigms work best with idealized modesl for stochastic neurons and synapses; additional biological features tend to degrade performance

- In addition, it is not likely that salient random variables are represented by single neurons in the brain. This also requires changes in the theory.

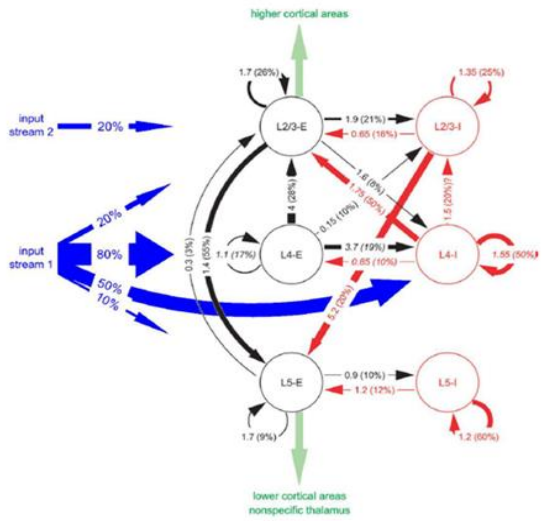
One theoretical result on stochastic computation that holds also for biologically detailed models

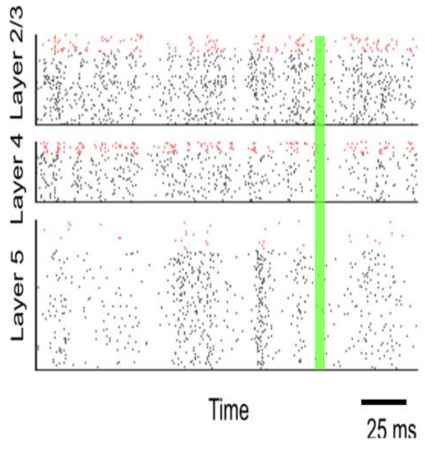
Even complex data based models of networks of neurons have a stationary distribution of network states z --and of spatio-temporal patterns

(Habenschuss, Jonke, Maass, PLOS CB 2013)

.

Inputs ***e*** *network states z* *a network state z*



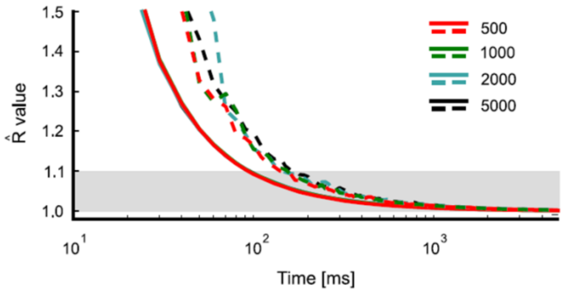


One possible advantage of biological network design: Convergence to stationary distribution is surprisingly fast for data- based microcircuit models (shown are curves are from Gelman-Rubin analysis).

Open problem: Why`?

This microcircuit can estimate for example (via MCMC sampling) posterior marginals, conditioned on external input e:

$$p(z_1|\mathbf{e}) = \sum_{v_2,\ldots,v_m} p(z_1, v_2, \ldots, v_m|\mathbf{e})$$



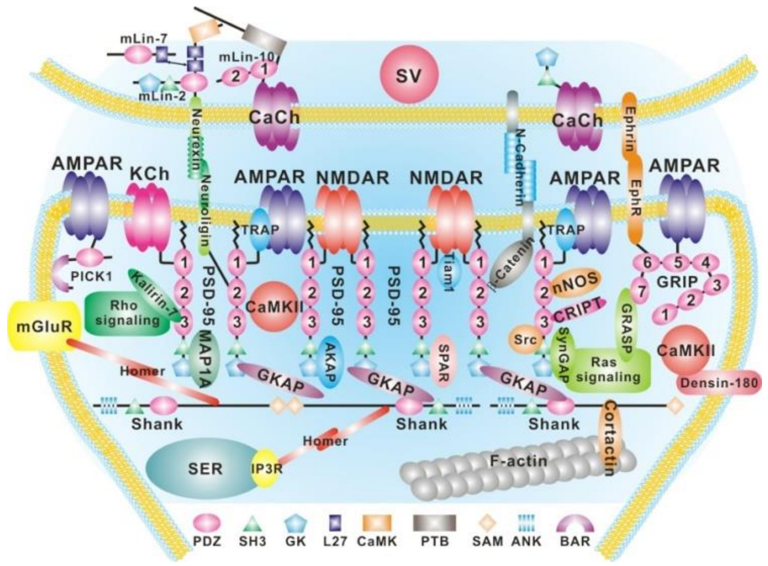
Principle 4: Brain networks are subject to permanently ongoing rewiring and parameter changes

This imposes constraints on models for learning, and provides hints for the organization for network plasticity

- One of the most puzzling fact about neural circuits is that they change all the time, even in the absence of overt learning

- How can such system have stable computational performance ?

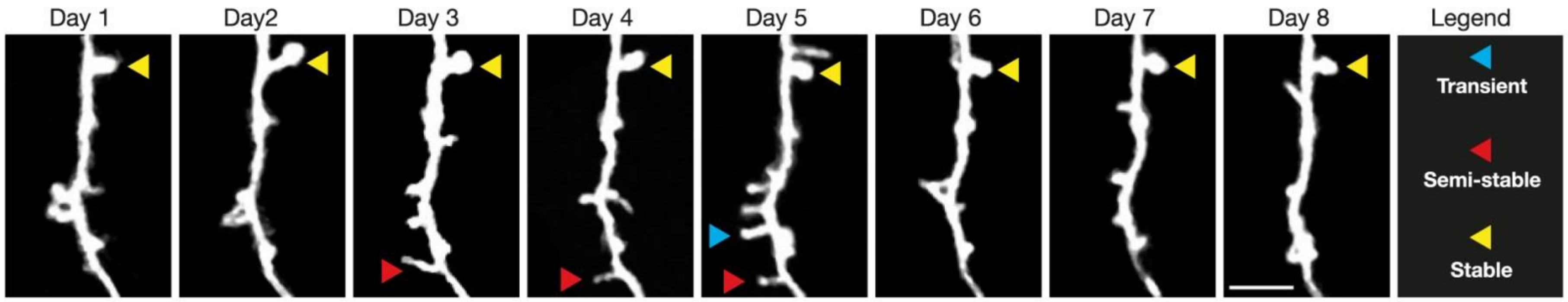
Some experimental data that demonstrate permanently ongoing network rewiring and parameter changes



A postsynaptic density consists of over 1000 different types of proteins, many in small numbers.

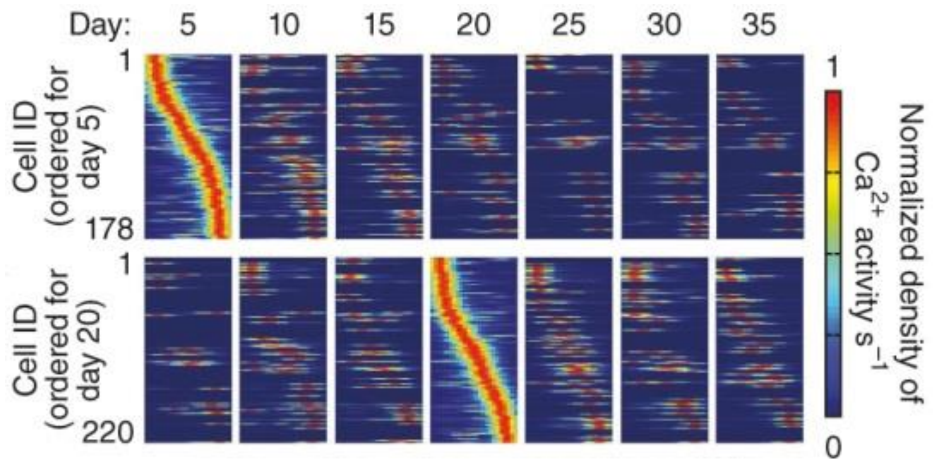
Since these molecules have a lifetime of only weeks or months, their number is subject to permanent stochastic fluctuations.
Receptors etc. are subject to Brownian motion within the membrane.

Furthermore axons sprout and dendritic spines come and go on a time scale of days (even in adult cortex, perhaps even in the absence of neural activity)



Data from Svoboda Lab

Longterm recordings show that *neural codes drift* on the time-scale of weeks and months



Ziv, Y., Burns, L. D., Cocker, E. D., Hamel, E. O., Ghosh, K. K., Kitch, L. J., ... & Schnitzer, M. J.. Long-term dynamics of CA1 hippocampal place codes. Nature Neuroscience, 2013

See also:

Rokni, U., Richardson, A. G., Bizzi, E., & Seung. Motor learning with unstable neural representations. *Neuron*, *2007*

and forthcoming new data.

Mathematical framework for capturing these phenomena: *„Synaptic Sampling“*

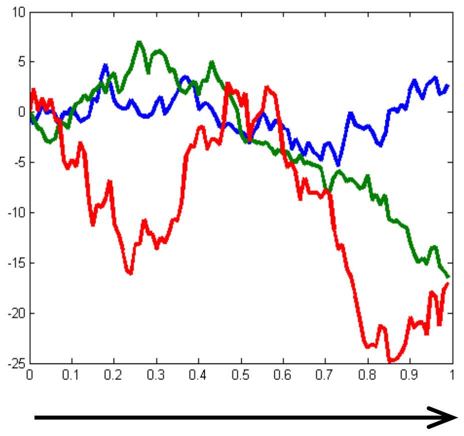
We model the evolution of network parameters through Stochastic Differential Equations (SDEs):

$$d\theta_i = \underbrace{\left(b\frac{\partial}{\partial\theta_i}\log p^*(\boldsymbol{\theta})\right) dt}_{\text{drift}} + \underbrace{\sqrt{2Tb}\cdot d\mathcal{W}_i}_{\text{diffusion}}$$

The diffusion term $d\mathcal{W}_i$ in the SDE denotes an infinitesimal step of a random walk („Brownian motion),

whose temporal evolution from time ***s*** to time ***t*** satisfies

$\boldsymbol{\mathcal{W}_i^t - \mathcal{W}_i^s \sim \mathrm{NORMAL}(0, t-s)}$.



time t

p^* $(\boldsymbol{\theta})$ can be any given target distribution of the parameter vector.

Mathematical framework for capturing these phenomena: „*Synaptic Sampling*“

The resulting **evolution of the probability density** of the parameter vector $\boldsymbol{\theta}$ is given by a deterministic PDE (*Fokker-Planck equation*):

$$\frac{\partial}{\partial t} p_{FP}(\boldsymbol{\theta}, t) = \sum_i -\frac{\partial}{\partial \theta_i}\left(\left(b \frac{\partial}{\partial \theta_i} \log p^*(\theta_i | \mathbf{x}, \boldsymbol{\theta}_{\backslash i})\right) p_{FP}(\boldsymbol{\theta}, t)\right) + \frac{\partial^2}{\partial \theta_i^2}\left(Tb\, p_{FP}(\boldsymbol{\theta}, t)\right)$$

By setting the left-hand side to 0, this FP-equation makes the resulting **stationary distribution** $\frac{1}{Z} p^*(\boldsymbol{\theta})^{\frac{1}{T}}$ for the vector $\boldsymbol{\theta}$ of all network parameters θ_i explicit.

Implication: One can **program into stochastic plasticity rules**

$$d\theta_i = \left(b \frac{\partial}{\partial \theta_i} \log p^*(\boldsymbol{\theta})\right) dt + \sqrt{2Tb} \cdot d\mathcal{W}_i$$

the desired target distribution $\frac{1}{Z} p^*(\boldsymbol{\theta})^{\frac{1}{T}}$ of the parameter vector.

This provides a principled of way designing and understanding local plasticity rules in neural networks.

In particular, synaptic sampling can implement sampling from a *posterior* distribution of network parameters

synaptic sampling
with prior $p_S(\boldsymbol{\theta})$

unsupervised learning (generative models)

$$p^*(\boldsymbol{\theta}|\boldsymbol{x}) \propto p_S(\boldsymbol{\theta})\, p_{\mathcal{N}}(\boldsymbol{x}|\boldsymbol{\theta})$$

where

- $\boldsymbol{x}$ are repeatedly occurring network inputs
- $p_{\mathcal{N}}(\boldsymbol{x}|\boldsymbol{\theta})$ is the generative model provided by a neural network $\mathcal{N}$ with parameters $\boldsymbol{\theta}$

reinforcement learning

$$p^*(\boldsymbol{\theta}) \propto p_S(\boldsymbol{\theta}) \cdot p_{\mathcal{N}}(\mathrm{R}=1|\boldsymbol{\theta})$$

where R signals reward

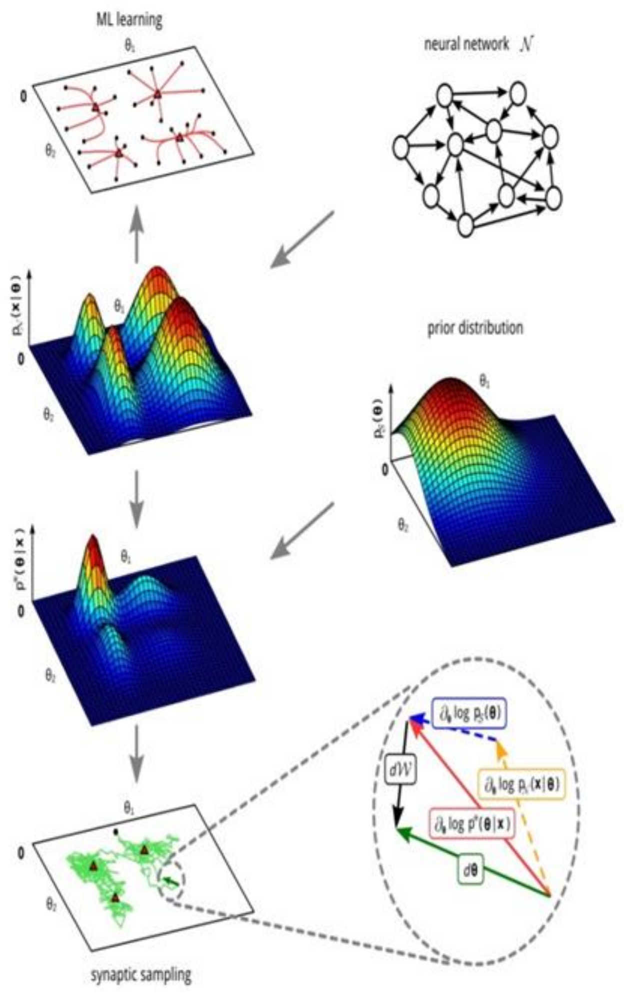
This integrates policy gradient RL with probabilistic inference.

Kappel, Habenschuss, Legenstein, Maass;
Network plasticity as Bayesian inference,
PLoS Comp Biol, *in press, and NIPS 2015*
(draft in Arxiv)

Kappel, Habenschuss, Legenstein, Maass;
Reward-based network plasticity as Bayesian inference,
RLDM 2015

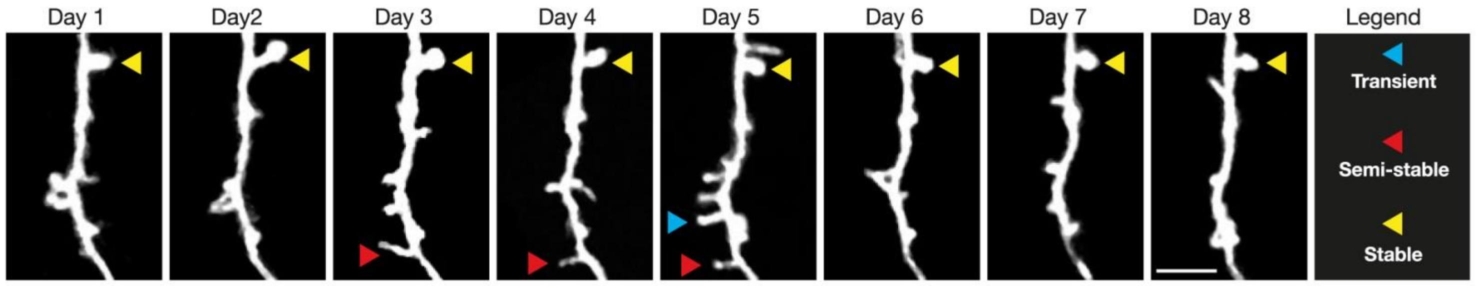
How does this change our understanding of network plasticity ?

- Priors enable the network to combine experience dependent learning with structural rules in a theoretically optimal way (**"learning as Bayesian inference"**)
- **Better generalization capability** through learning of a posterior (predicted by MacKay, 1992)
- **Structural plasticity** (rewiring) can easily be **integrated** into this learning framework
- Learning does not fix the **parameters** $\boldsymbol{\theta}$ of the network at some optimal position (as in max. likellihood learning), Rather, parameters (and neural codes) **keep moving** within some low-dimensional manifold where both prior and network performance are high
- Network perturbations and lesions are no big deal, since parameters do not converge to particular values (**automatic self-repair**)

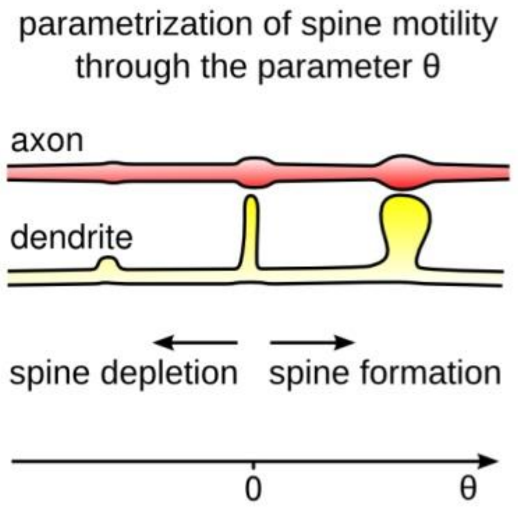

ML learning
neural network $\mathcal{N}$
prior distribution
$p_S(\theta)$
$p_\mathcal{N}(x|\theta)$
$p^*(\theta|x)$
θ_1
θ_2
0
$\partial_\theta \log p_S(\theta)$
$\partial_\theta \log p_\mathcal{N}(x|\theta)$
$\partial_\theta \log p^*(\theta|x)$
dW
$d\theta$
synaptic sampling

Demos of that in (Kappel, Habenschuss, Legenstein, Maass;
Network plasticity as Bayesian inference, PLoS Comp Biol, *in press,*
(draft in Arxiv)

Spine dynamics and synaptic plasticity can easily be integrated into a SDE for a parameter that regulates both

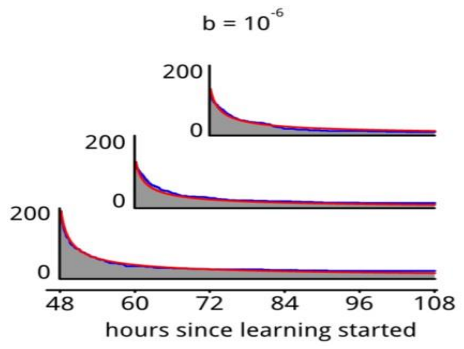


Ansatz: A single parameter θ_i controls the spine volume and – once a synaptic connection has been formed – the weight of this synaptic connection.



Not only **STDP,** but also **experimentally observed power-law survival curves for synaptic connections** are reproduced by this combined rule:

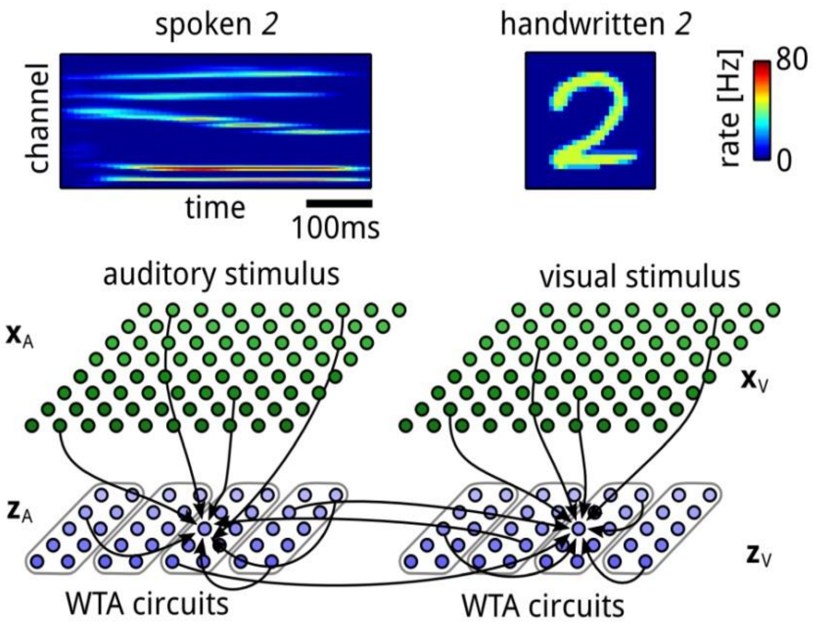
Experimental data from (Löwenstein, Kuras, Rumpl, J. of Neuroscience, 2015)



Synaptic sampling yields automatic self-repair

Example: Self-repair of a generative model: Two generative models „visual cortex“ z_V , and „auditory cortex“ z_A both modelled as recurrent networks of spiking WTA circuits.

Both receive during learning handwritten and spoken versions of the same digit („1“ or „2“), transformed into firing rates



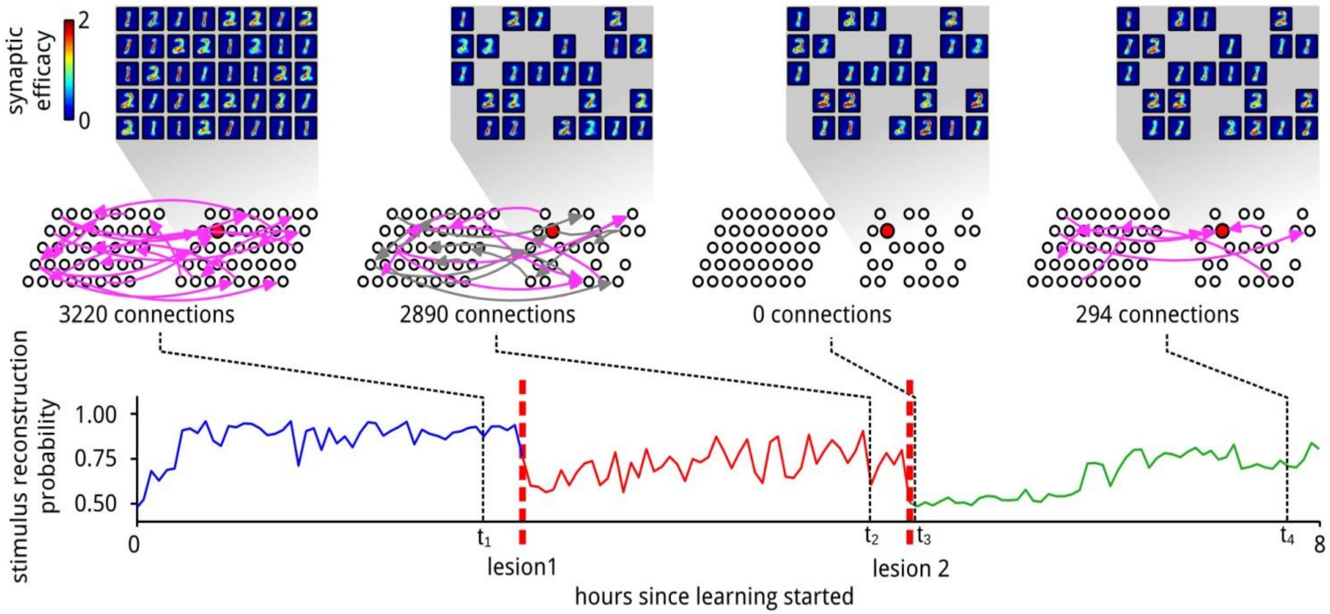
All synaptic connections between inputs and hidden neurons, and among hidden neurons are allowed to grow

The previously described synaptic sampling rule is applied to all of these potential synaptic connections

Test of self-repair capability through synaptic sampling

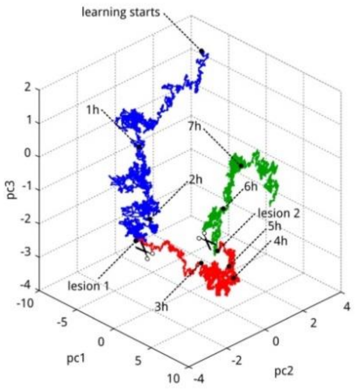
We removed in 2 successive lesions

1. all neurons from the „visual cortex“ z_V that had created in their weights a generative model for digit „2“
2. all synaptic connections between the „visual cortex“ z_V and the „auditory cortex“ z_A (and these were not allowed to regrow)



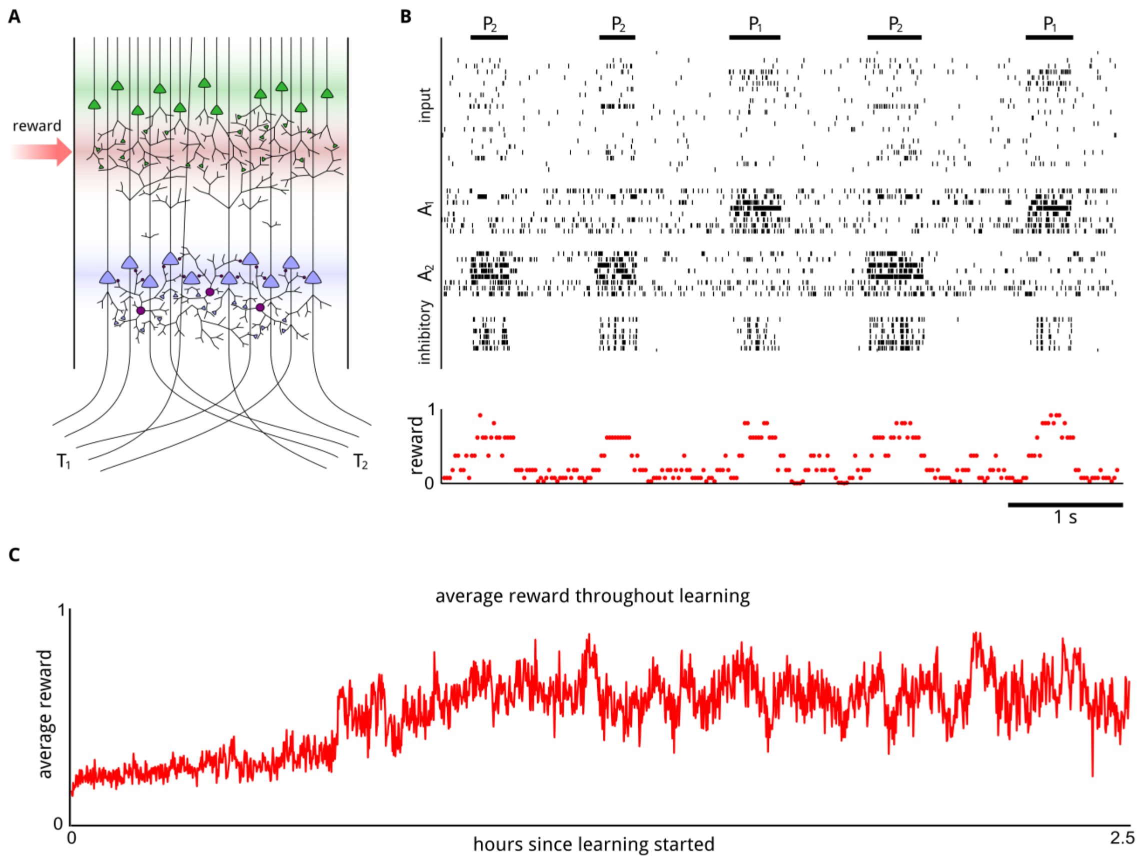
in gray: connections from preceding phase

Result: The network performance (measured by information about current digit in visual cortex when only auditory input was provide) recovered after each lesion .



First 3 principal components of a subset of the parameters $\boldsymbol{\theta}$:

Simple demo of reward-based synaptic sampling



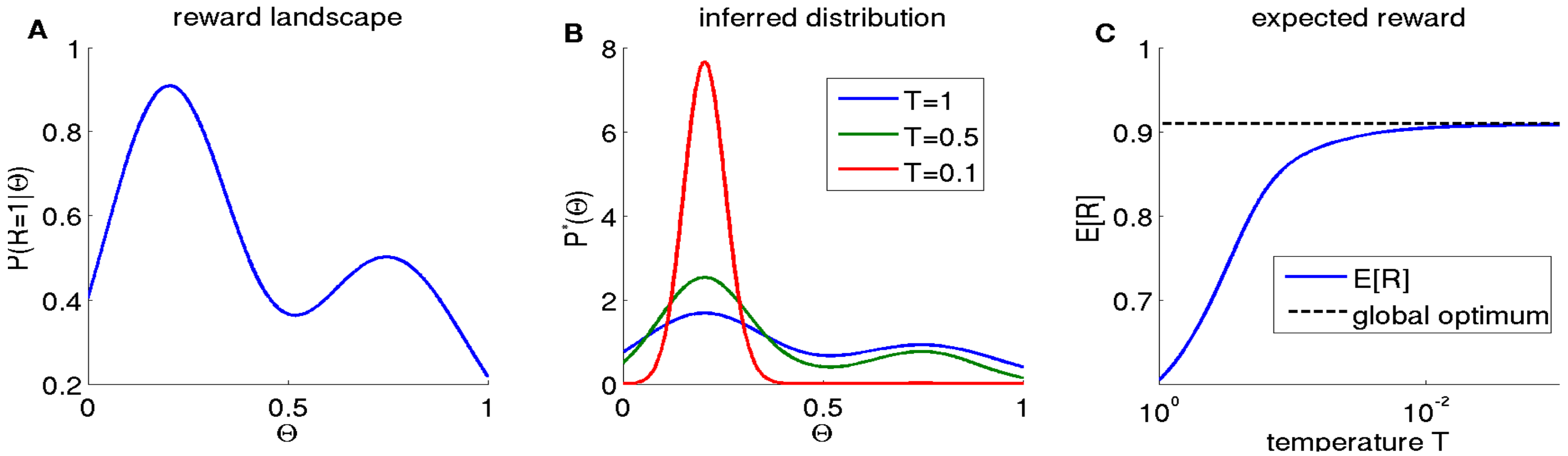
Network is rewarded if the assembly that projects to target T_i fires more for input that resembles pattern P_i

(Kappel et al, in preparation)

Reward-based synaptic sampling can approximate global network optimization (simulated annealing)

The expected reward depends on the temperature *T* *(which regulates the amplitude of spontaneous stochastic changes of network parameters)* as follows (for a flat prior):

$$E[R] = \frac{1}{Z} \int p_{\mathcal{N}}(\mathrm{R} = 1\,|\boldsymbol{\theta})\ \ p_{\mathcal{N}}(\mathrm{R} = 1\,|\boldsymbol{\theta})^{\frac{1}{T}}\, \mathrm{d}\boldsymbol{\theta}$$



Hence a cooling schedule for the stochastic dynamics of network parameters is in principle able to find **globally optimal solutions** (like in simulated annealing).

Consolidation of synaptic weights and connections could be viewed as special case of such cooling.

3. Visions for future work, and open problems

I have argued that constraints provided by experimental data suggest specific principles of brain computation:

1. Neural circuits are highly recurrent networks of neurons and synapses with diverse dynamic properties

I had discussed one Theorem that suggests a functional role for this diversity in a feedforward setting; more theoretical analysis is needed that addresses diversity, especially in recurrent networks

2. Neural computation needs to serve neural users (which receive high-D inputs, and are adaptive)

I am suggesting that we view neural computation as preprocessing for learning

3. Brain computations are subject to high trial-to-trial variability

Hence we should consider models for stochastic brain computation

4. Brain networks are subject to permanently ongoing rewiring and parameter changes

Hence we should consider models for learning that do not aim at convergence to a local optimum, such as max. likelihood learning

Interesting aspect regarding Marr/Poggio levels of analysis

- The 4 constraints of biological implementation on which I had focused all suggest specific approaches on the computational and algorithmic level, which are different from commonly chosen ones by theoretical neuroscientists.

- Hence it seems to make little sense to start top-down with clever computational or algorithmic approaches (but little knowledge of experimental data), and hope that these magically meet constraints of biological implementions

- Additional desirable property of the 4 principles that I have discussed is, lthat they have a certain level of generality, i.e., are applicable to a fairly large variety of models.

- In contrast, many socalled „abstract“ models for biological neural networks (even models in textbooks) are inconsistent with experimental data; i.e., they do **not** cover more detailed models as special cases.

Major open problem areas in the theory of neural computation

- We need better concepts, models, and tools for understanding computational properties of **high-dimensional stochastic dynamical systems** (consisting of diverse units**)**
- How can we address the **complexity of data on synapses** (including the molecular level), their hidden variables and their dynamics, complex dependencies on neuromodulators and network activity history?
- Understand the role of **stereotypical spatio-temporal activity patterns of neurons** for neural computations? (are these „words“ or even „sentences“ of neural codes?)
- Understand how **stochastic computations can arrive fast at good solutions** (not necessarily „arbitrary“ initial states)
- We are also missing theoretical tools for dealing with **diverse dynamic network components in stochastic computations.**
- How should the **interaction of neocortex with other brain areas** (including thalamus) be reflected in our computational analysis?

How can we improve the „science“-aspect of theoretical neuroscience?

- make discrepancies between different theoretical approaches explicit (rather than being polite/tactical)
- invest more efforts into falsifiable theory
- encourage a „risk analysis“ of models for neural systems, rather than covering up pros and cons of a model by claiming that it is „biologically plausible“
- encourage radically new ideas and theories (we may not even have understood the basics of neural computation!)
- recruit also clever minds from mathematics and theoretical computer science (but direct them towards stochastic dynamical systems or other questions motivated by biological data)
- set up a list of benchmark tasks (linked to experimental data that provide clues how brains solve such tasks)

I will discuss the work of the following former and current members of our team in Graz (Opening for Postdoc/Assist. Prof. !)

Bernhard Nessler
(FIAS, Frankfurt)

Michael Pfeiffer
(ETH Zurich)

Lars Büsing
(Columbia Univ)

Stefan Habenschuss
(software industry)

Dejan Pecevski
(software industry)

Zeno Jonke
(software industry)

David Kappel

Robert Legenstein