A Model for Computation in Cortical Microcircuits

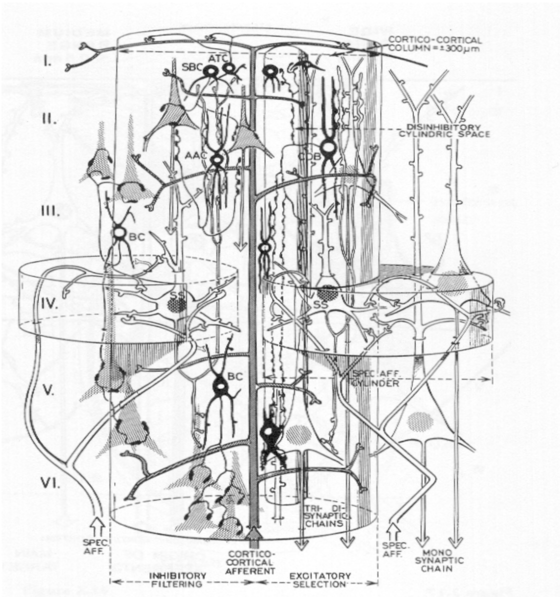
Wolfgang Maass

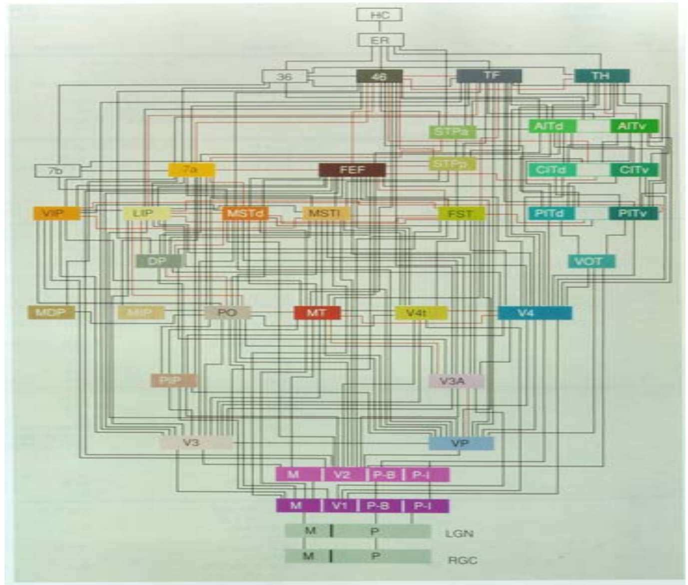
Institut für Grundlagen der Informationsverarbeitung
Graz University of Technology, Austria

Table of Content

1. Main hypothesis:

The computational task of a typical cortical microcircuit is to support (as a general preprocessor, e.g. via temporal integration and nonlinear operations) diverse computational goals of a variety of readout neurons.



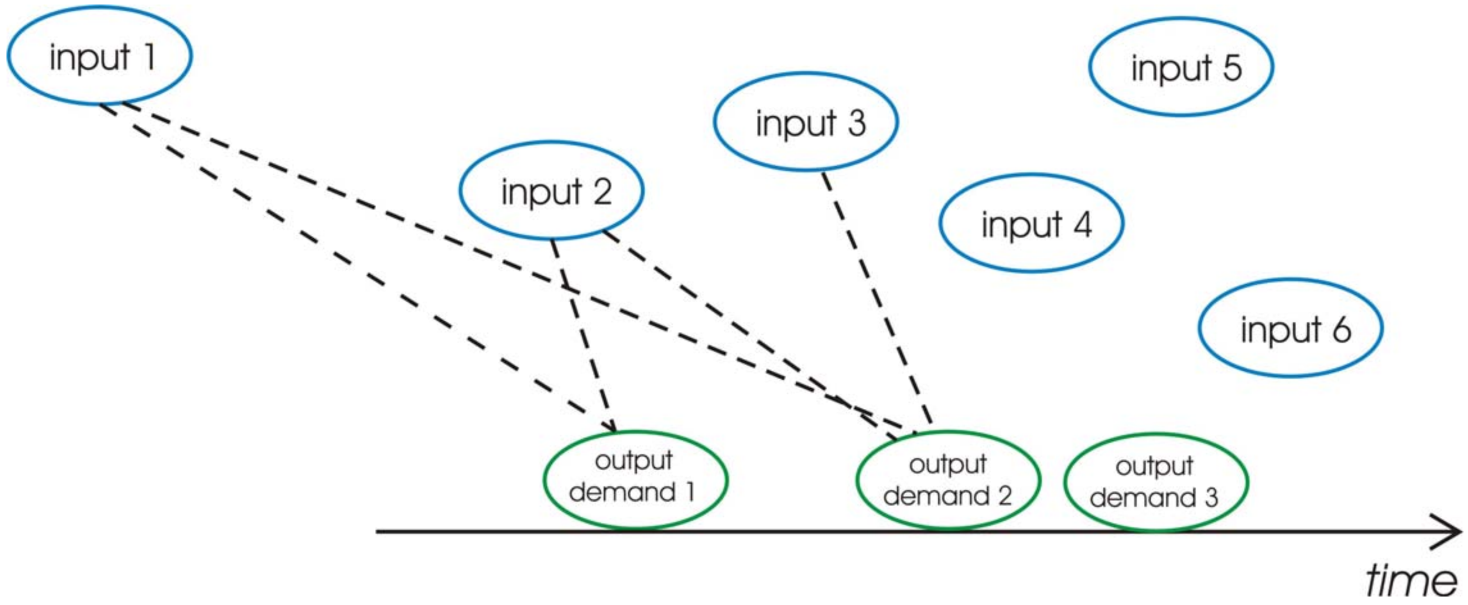


Traditional (and more common) hypothesis: The computational task of a cortical microcircuit is to carry out a specific computational operation (e.g. multiplication, extraction of orientation).

What types of computations should we consider ?

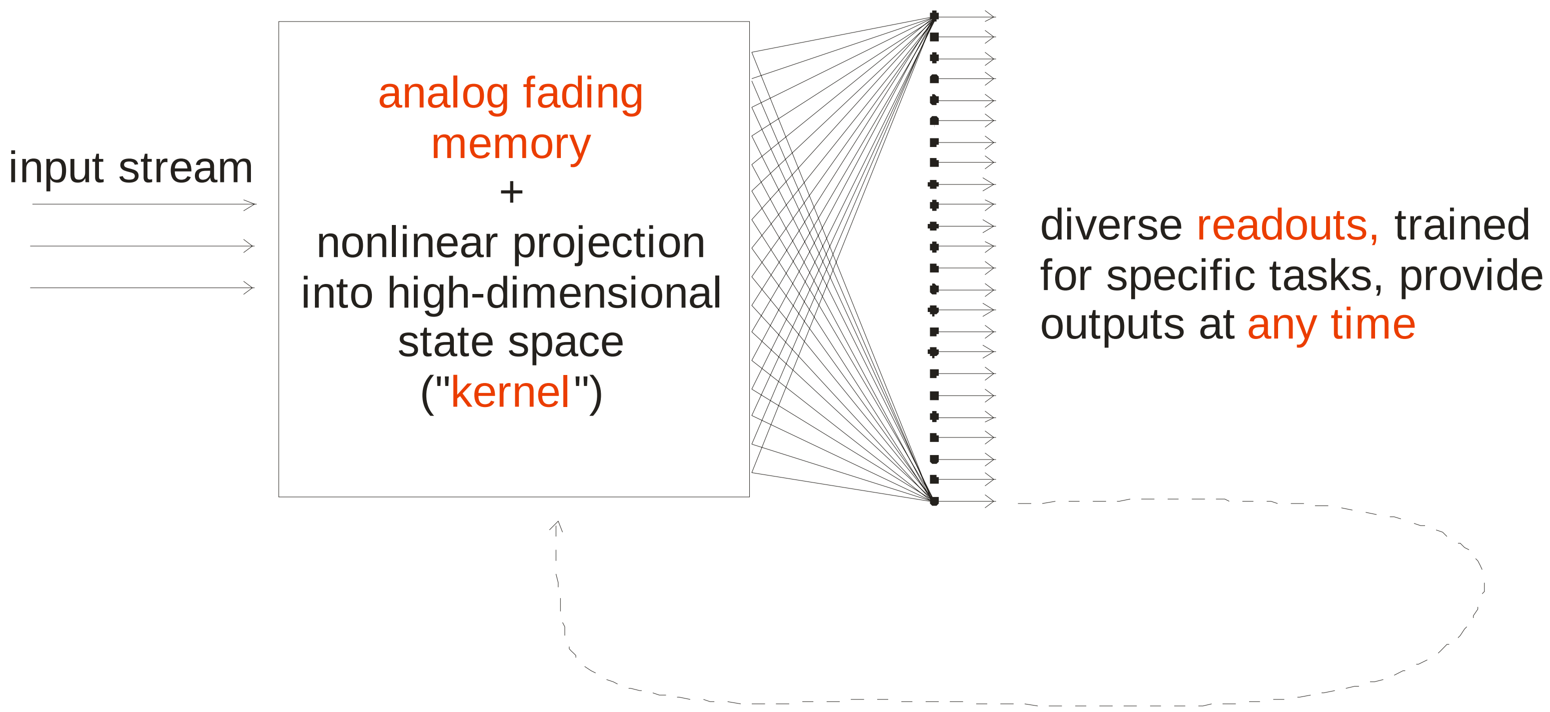
Classical models for computation (Turing machines, attractor neural networks) do not capture well the actual computational tasks that a biological organism has to perform in order to survive:

It receives continuously new pieces of information arrive, and demands for results of computations may arise at any time ("anytime algorithm", "real-time computing"):



Hence from a mathematical point of view, neural readouts have to implement filters (operators), i.e. they map input streams to output streams (rather than implementing a static function, such as multiplication).

Resulting computational model for a generic cortical microcircuit: [joint work with Natschlaeger and Markram] :



Insert: What is a kernel (in the terminology of machine learning) ?

A kernel provides many nonlinear combinations of input variables, in order to boost the expressive power of any subsequent linear readout.

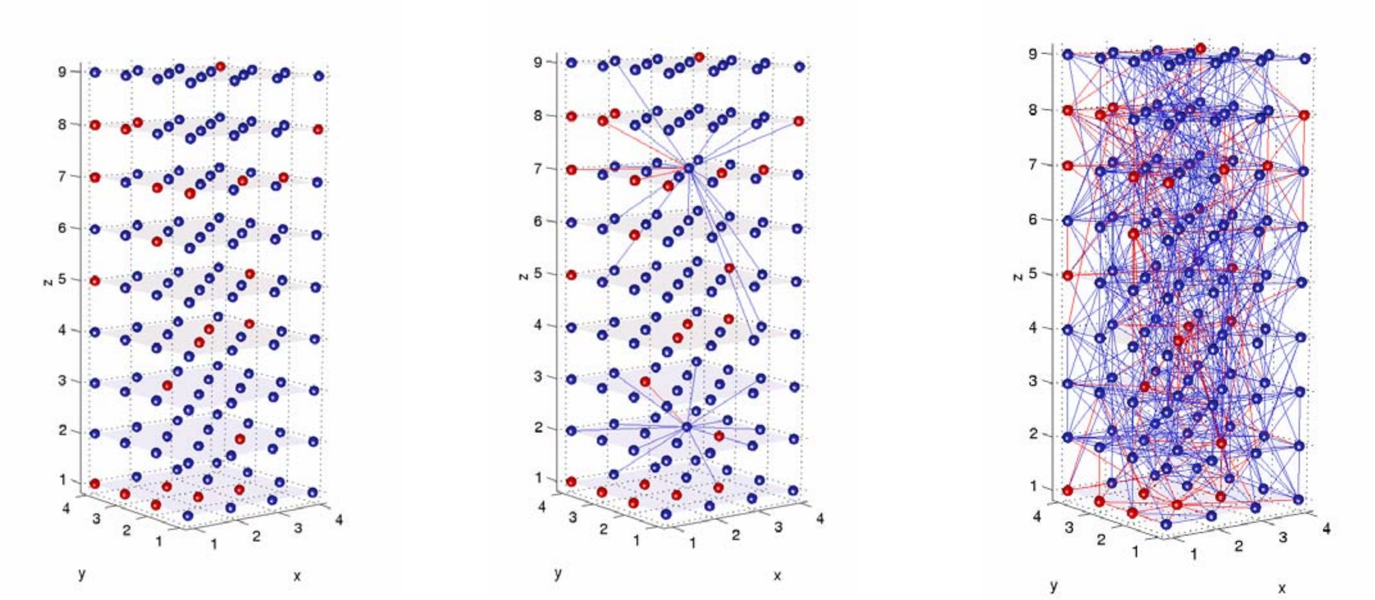
Example: If a circuit precomputes all products $x_i \cdot x_j$ *of n input variables* $x_1,\ldots,x_n$, *then every subsequent linear readout can compute any quadratic function of the original input variables* $x_1,\ldots,x_n$.

Remark 1: A clear theoretical advantage of linear readouts: their learning cannot get stuck in local minima of the error function.

This fact suggests that it is advantageous for nature to restrict learning to linear devices.

Remark 2: Because of Vapnik's "kernel trick" one can use in machine learning big kernels without additional computational cost. This is different for neural circuits that have to implement a kernel explicitly !

2. Testing the hypothesis through computer simulations of generic cortical microcircuits



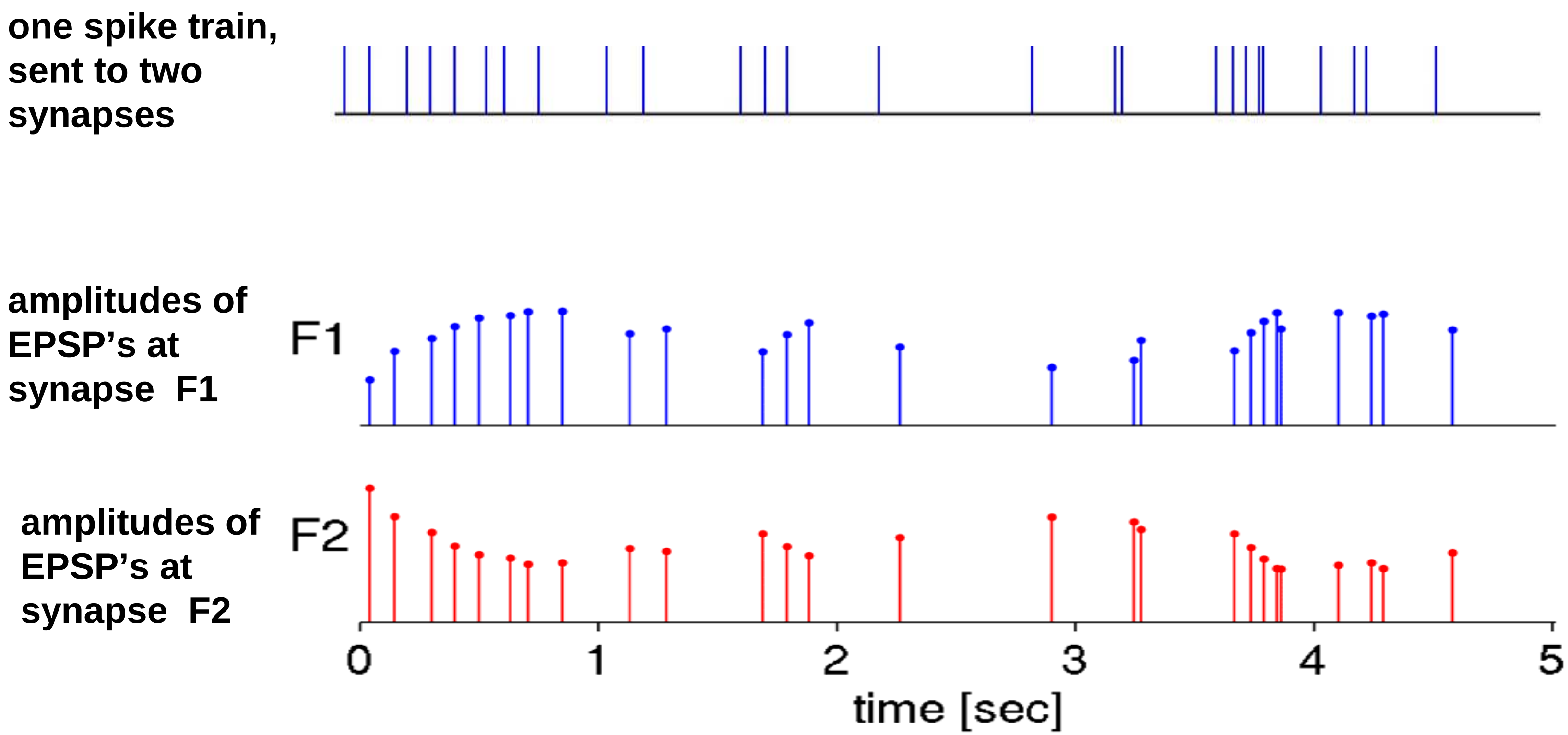
neurons: leaky integrate-and-fire neurons, 20% of them inhibitory, neuron a is synaptically connected to neuron b with probability $C \cdot \exp(-D^2(a,b)/\lambda^2)$

synapses: dynamic synapses with fixed parameters w, U, D, F chosen from distributions based on empirical data from the Lab of Markram

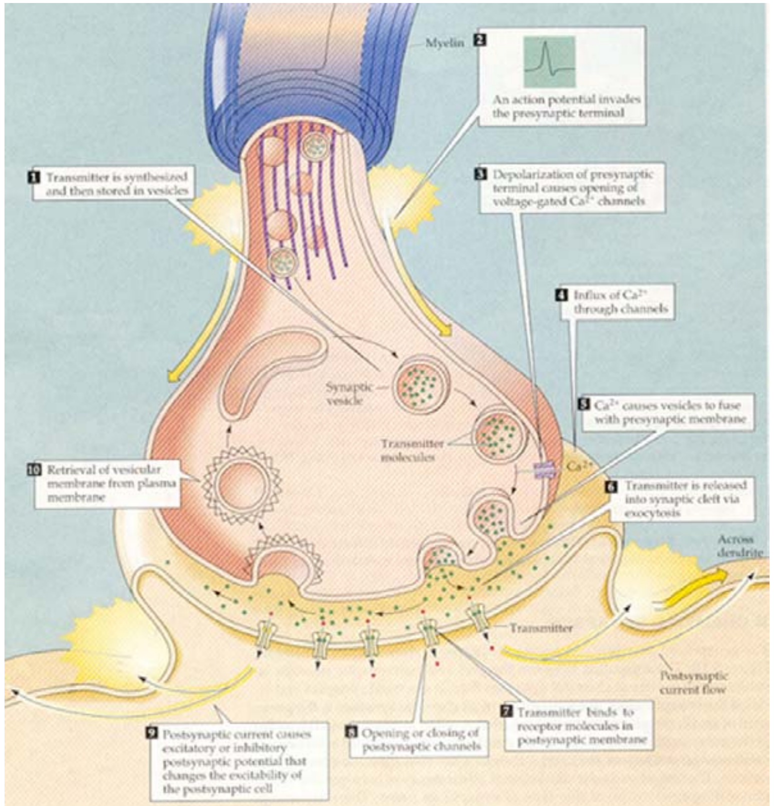
input spike trains injected into 30% randomly chosen neurons, with fixed randomly chosen amplitudes

Models for neural microcircuits differ strongly from artificial neural networks, since biological neurons produce spike trains as outputs, and synapses respond to these spike trains in diverse nonlinear ways

Shown here are the amplitudes of EPSP's for two common types of synapses, for the same spike train (F1 is facilitating and F2 is depressing):



We model synapses in our circuit simulations with parameters w, U, D, F



according to
[Markram, Wang, Tsodyks, PNAS 1998]:

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

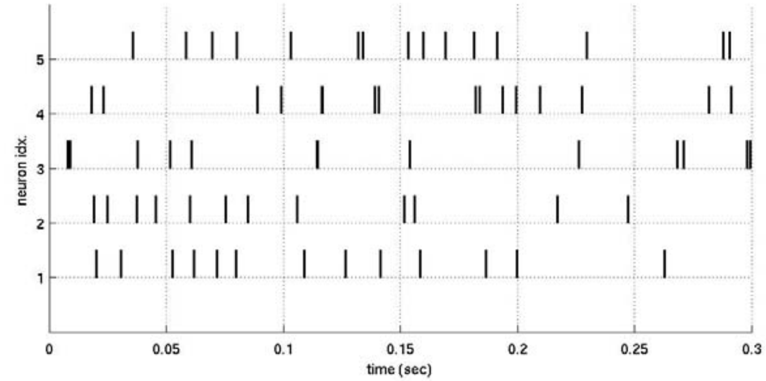
$$A_k = w \cdot u_k \cdot R_k$$
$$u_k = U + u_{k-1}\,(1-U)\, exp(-\,\Delta_{k-1}/F)$$
$$R_k = 1 + (R_{k-1} - u_{k-1}\, R_{k-1} - 1)\, exp(-\,\Delta_{k-1}/D)$$

Mean values of U, D, F according to experimental data from the Lab of Henry Markram (in dependence of the type of pre-and post-synaptic neuron):

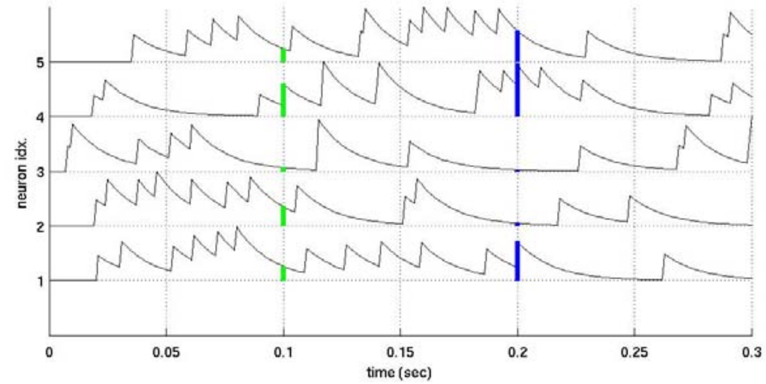
from \ to	E	I
E	**0.5, 1.1, 0.05**	**0.05, 0.125, 1.2**
I	**0.25, 0.7, 0.02**	**0.32, 0.144, 0.06**

A simple model for a neural readout:
a linear weighted sum with adaptive weights **w**

Each readout neuron receives as input a vector x(t), which has as many components as it has presynaptic neurons in the circuit.



The i-th component of x(t) results from the spike train of the i-th presynaptic neuron by applying a low-pass filter, which models the low-pass filtering properties of receptors and membrane of the readout neuron.

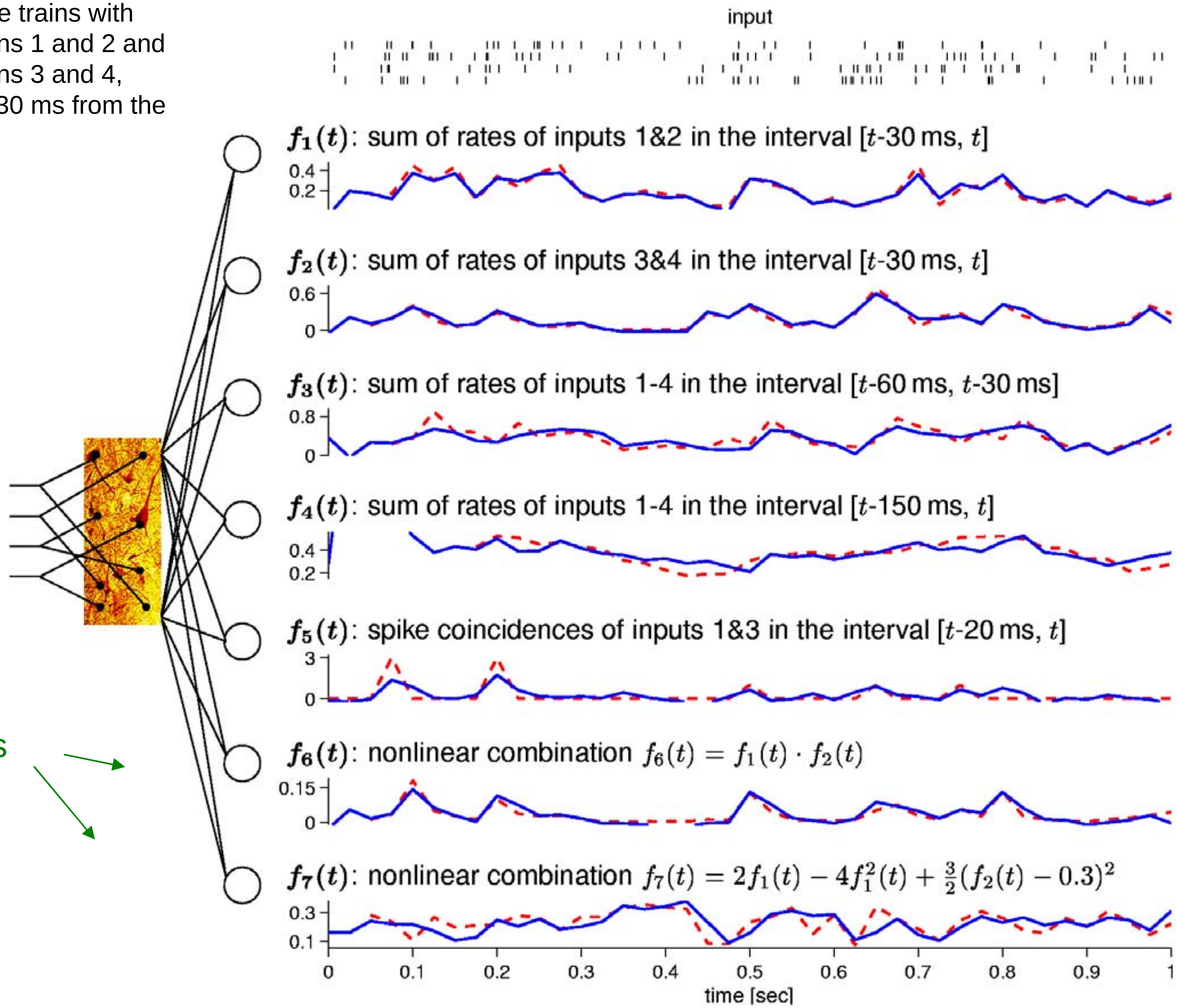


We assume that a readout neuron has at time t a firing rate proportional to $\mathbf{w} \cdot x(t)$.

What can a generic cortical microcircuit compute in this way ?

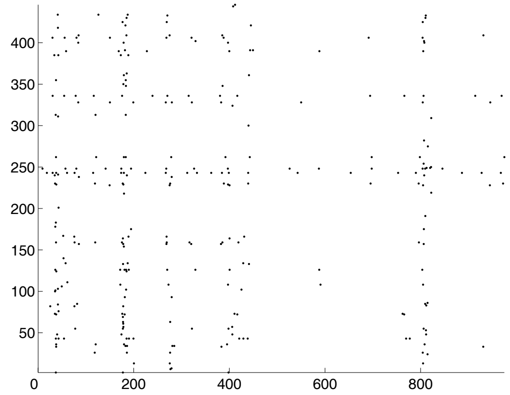
Circuit input: 4 Poisson spike trains with firing rates $f_1(t)$ for spike trains 1 and 2 and firing rates $f_2(t)$ for spike trains 3 and 4, drawn independently every 30 ms from the interval [0, 80] Hz

7 linear readouts with adjustable weights



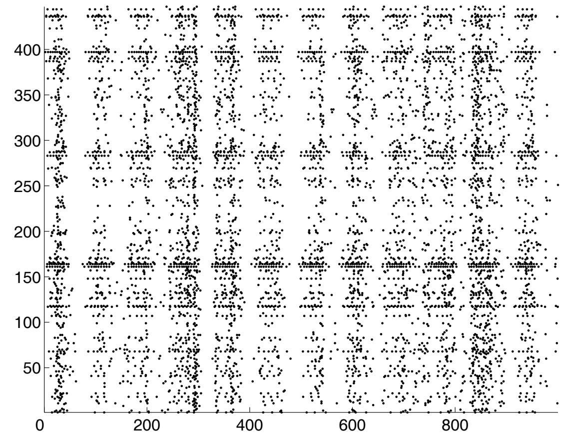
This is even possible if the circuit has a spontaneous dynamics (which is not related to its input):

a

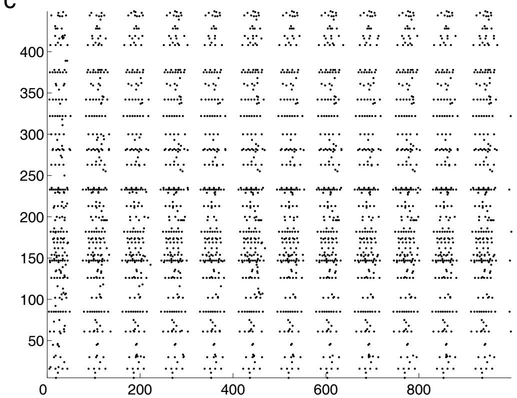


Spike raster resulting from spike input

b



c



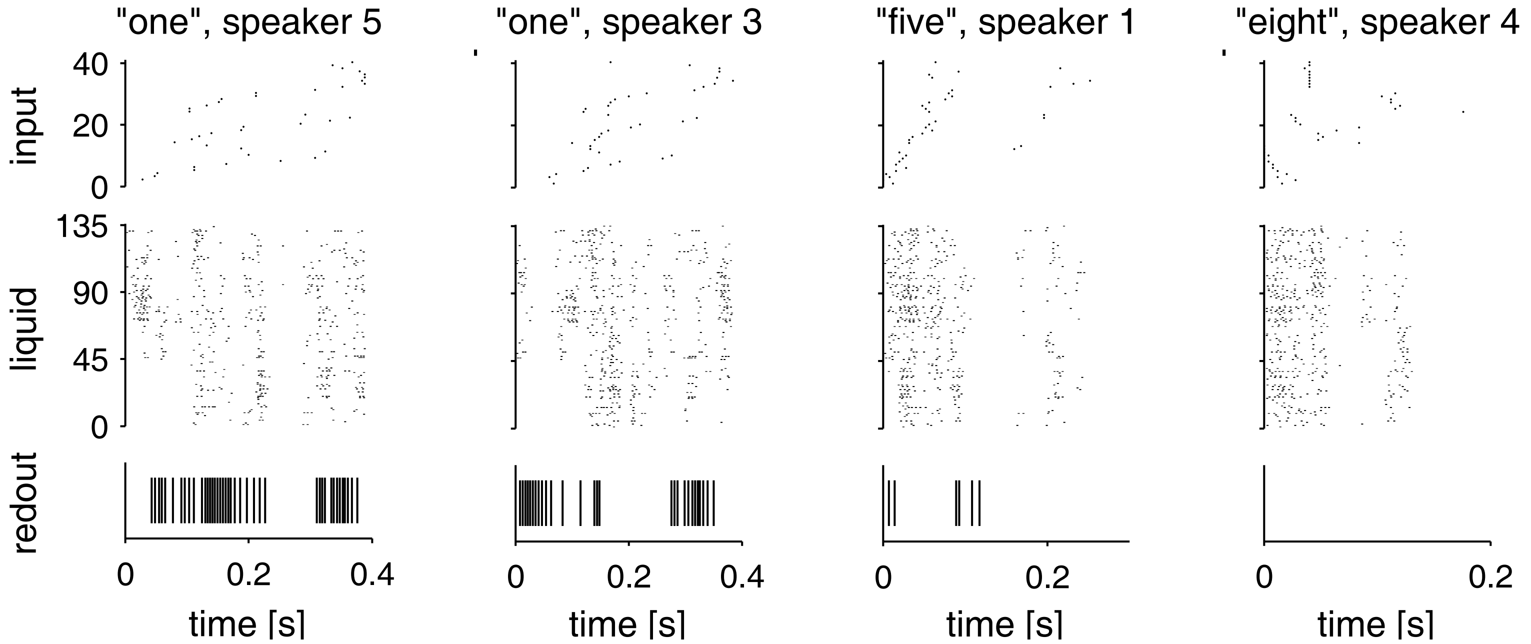
Spike raster resulting from a separate periodic input

Spike raster resulting from the superposition of both types of inputs.

It is shown in [Kaske and Maass, 2006] that this ongoing activity does not destroy the generic computational properties of the circuit.

Linear readouts from a generic microcircuit model can also be trained to classify a spoken word (encoded by spike trains), even before the spoken word ends. Hence generic circuit models can implement "anytime algorithms".

Example: anytime recognition of "one" :



3. Theoretical analysis

Unfortunately most traditional models and theoretical approaches are designed for other types of computations :

	Turing machines	cellular automata	iterative maps	differential equations	threshold circuits	cortical microcircuits
analog?	no	no	yes	yes	no	yes
continuous time?	no	no	no	yes	no	yes
high-dimensional ?	yes	yes	no	usually no	yes	yes
with noise?	no	no	no	usually no	no	yes
with online input?	no	no	no	yes	usually no (exception: Bertschinger et al, Neural Comp. 2004)	yes

A common mathematical framework for characterizing filters:

Volterra series (or Wiener series)

$$(F\,u(\cdot))\,(t) \;=\; \alpha_1 \int_0^{\infty} d\,\tau_1\; h_1\,(\tau_1)\, u\,(t-\tau_1)$$
$$+\,\alpha_2 \int_0^{\infty}\!\!\int_0^{\infty} d\,\tau_1\; d\,\tau_2\; h_1\,(\tau_1,\tau_2)\, u\,(t-\tau_1)\cdot u\,(t-\tau_2)$$
$$+\ldots$$

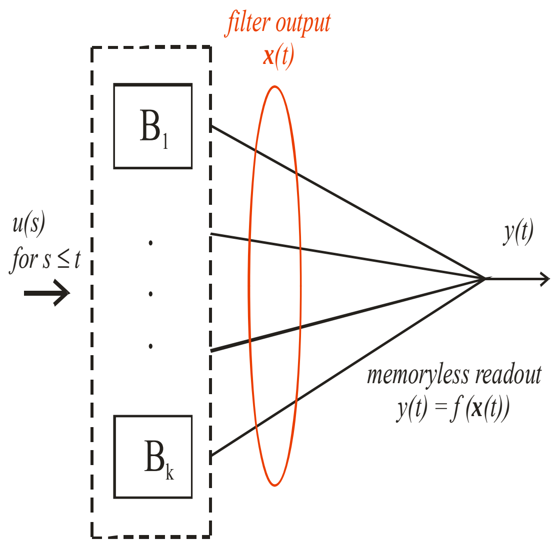
Note that any such filter F has fading memory:
In order to determine the output $(Fu(\cdot))(t)$ with a given precision ε it suffices to know the values of $u\,(t-\tau)$ up to some finite precision δ for all τ from some finite time interval $[0, T]$.

Question: Might cortical microcircuits be able to approximate any filter F that can be defined by a Volterra series?

A basic mathematical results: Any filter F which is defined by a Volterra series can be approximated with any desired degree of precision by a circuit of the type shown on the r.h.s.:

- ***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*

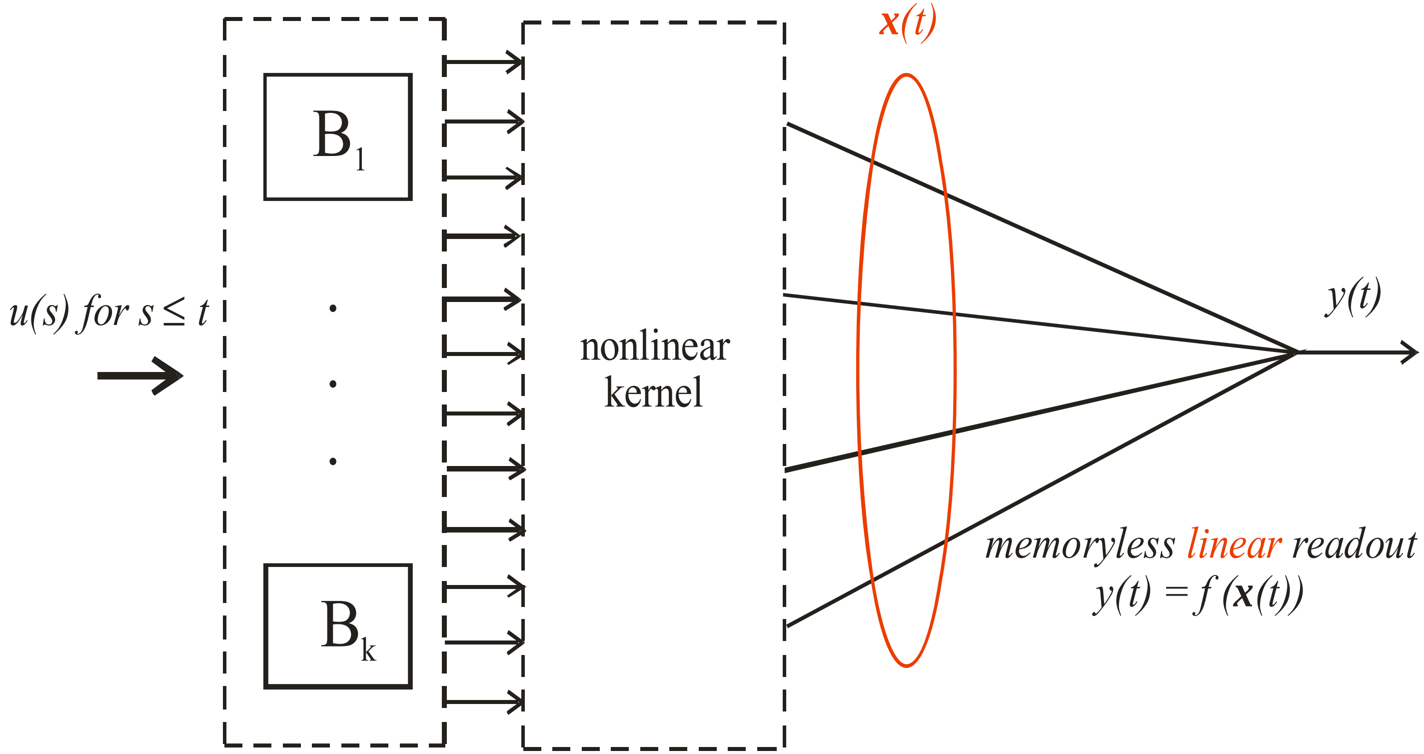
- ***if*** there is a rich enough pool **R** from which the readout functions f can be chosen
  (**R** needs to have the universal approximation property).



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 $s \leq t$ a basis filter $B \in$ **B** with $(Bu)(t) \neq (Bv)(t)$.

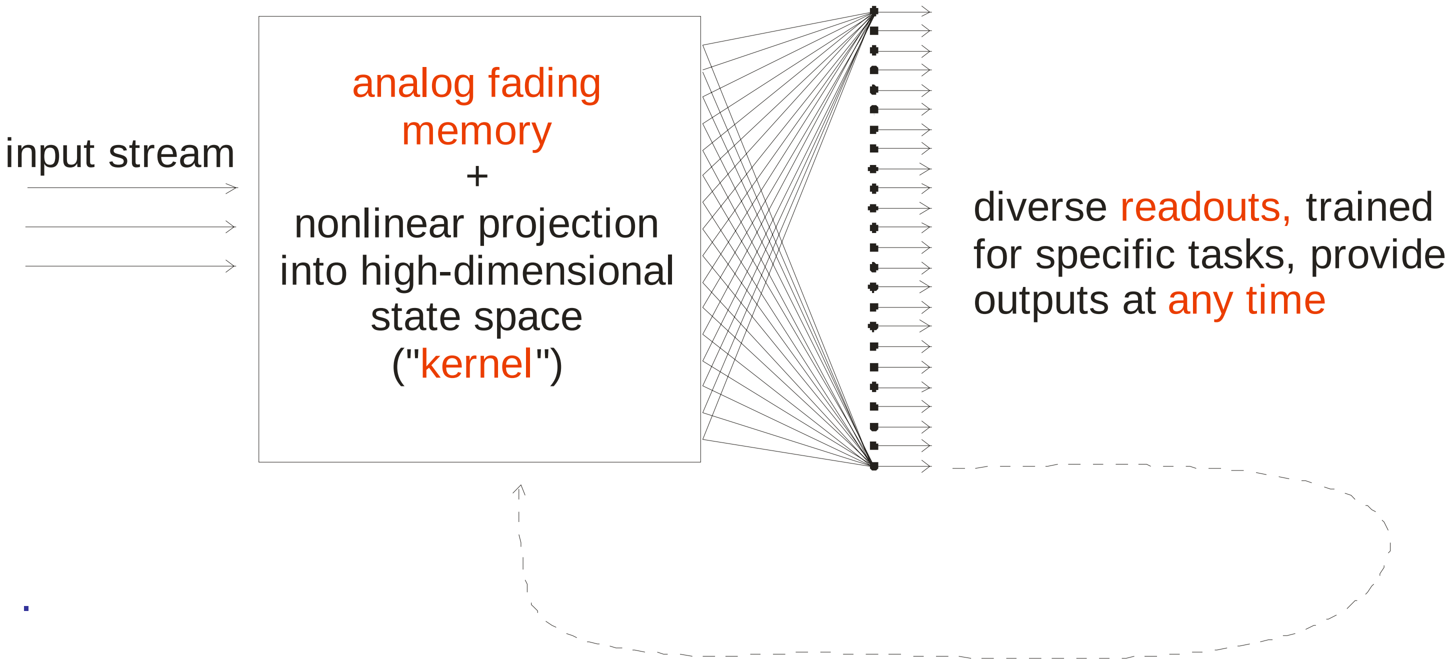
A possible functional interpretation of ongoing activity:
Ongoing activity in a recurrent circuit ("cortical songs", synfire chains, "avalanches", etc) **contributes in a cortical circuit to the realization of this separation property.**

Hence this model suffices to guarantee theoretically that any computation with fading memory (i.e., any Volterra series) can be approximated:



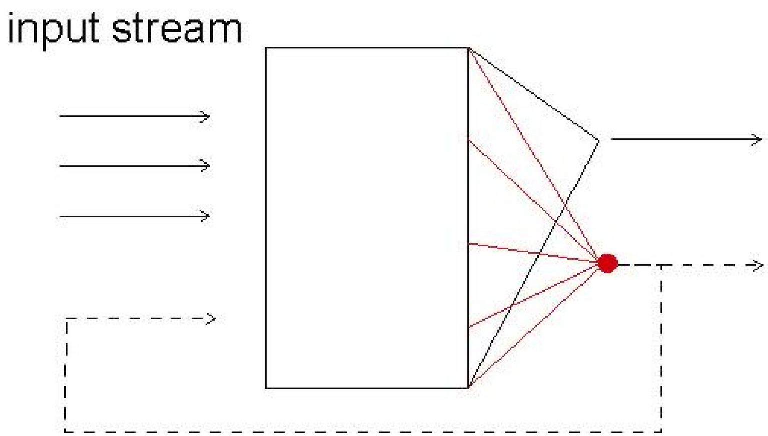
Can such model also carry out computations that require persistent memory, or other long-lasting internal states ?

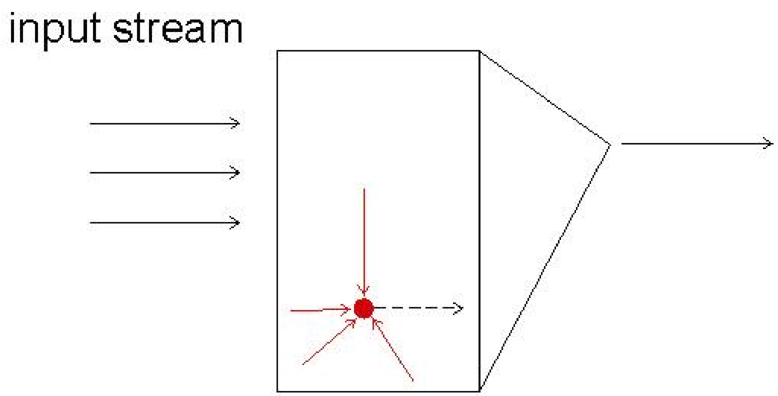
[Maass, Joshi, Sontag, 2006] :
Yes, if one allows in addition feedback from trained readouts...



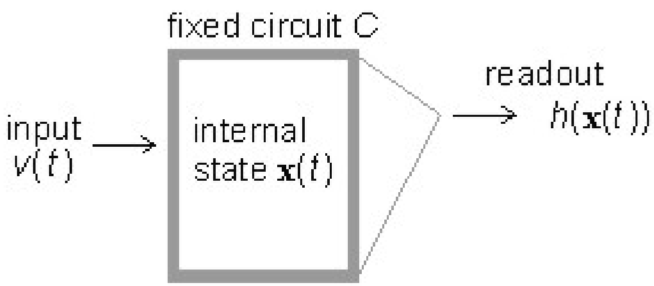
.

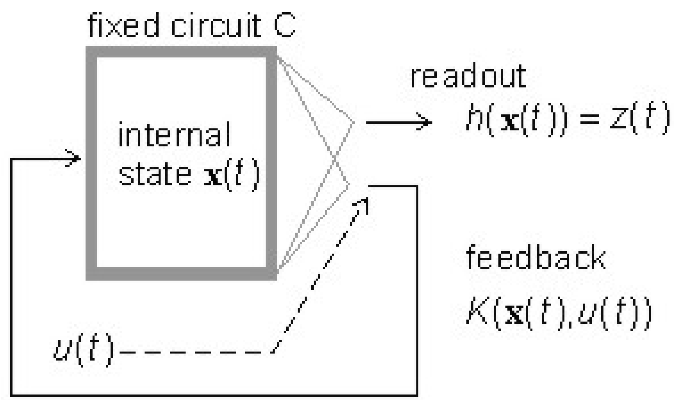
... or if one trains neurons within the circuit for specific tasks:

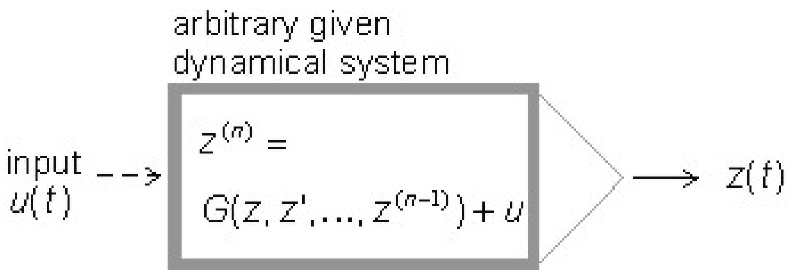




Underlying mathematical theory: There exists a large class S_n of analog circuits C with fading memory (described by systems of n first order differential equations) that gain through feedback universal computational capabilities for analog computing.







Note: Any Turing machine can be simulated by such dynamical system [Branicky, 1995],

hence all digital computations (including those that require a non-fading memory).

This holds in particular for neural circuits *C* defined by DEs of the form

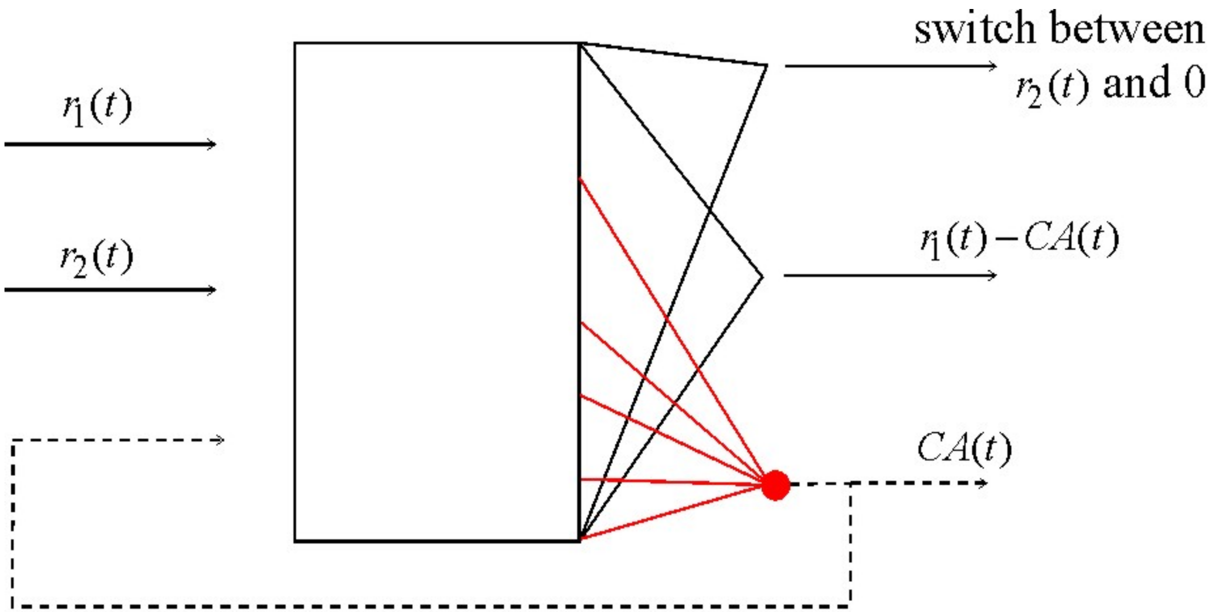
$$x_i'(t) = -\lambda_i x_i(t) + \sigma\left(\sum_{j=1}^{n} a_{ij} x_j(t)\right) + b_i \cdot \sigma(v(t))$$

(under some conditions on the λ_i, a_{ij}, b_i).

Note: The required feedback functions K and readout functions h are always continuous (and memory-less), hence they provide suitable targets for learning.

If the circuit C has sufficient kernel-capability, then K and h can be chosen to be linear.

Application of this theoretical result for a particular computational task (that cannot be carried out with a fading memory):



feedback that creates the continuous attractor

with target value $\int_0^t (r_1(s) - r_2(s))\,ds$

What is different in the training procedure for a readout that provides feedback ?

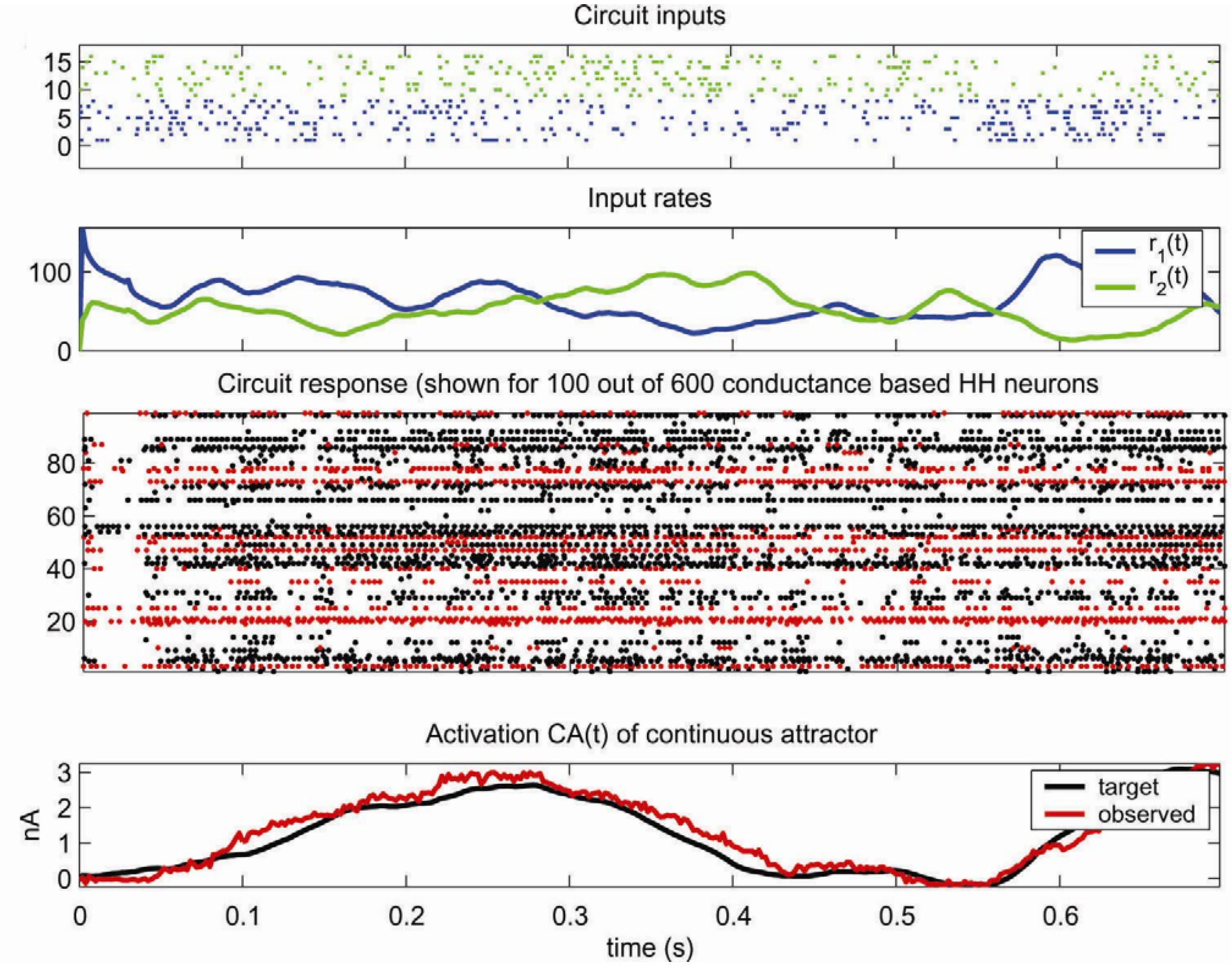
- Each such readout was trained by linear regression to map vectors $x(t)$ to specific target values $K(x(t))$

  (their feedback was injected as an extra input current into a randomly chosen subset of neurons in the generic circuit)

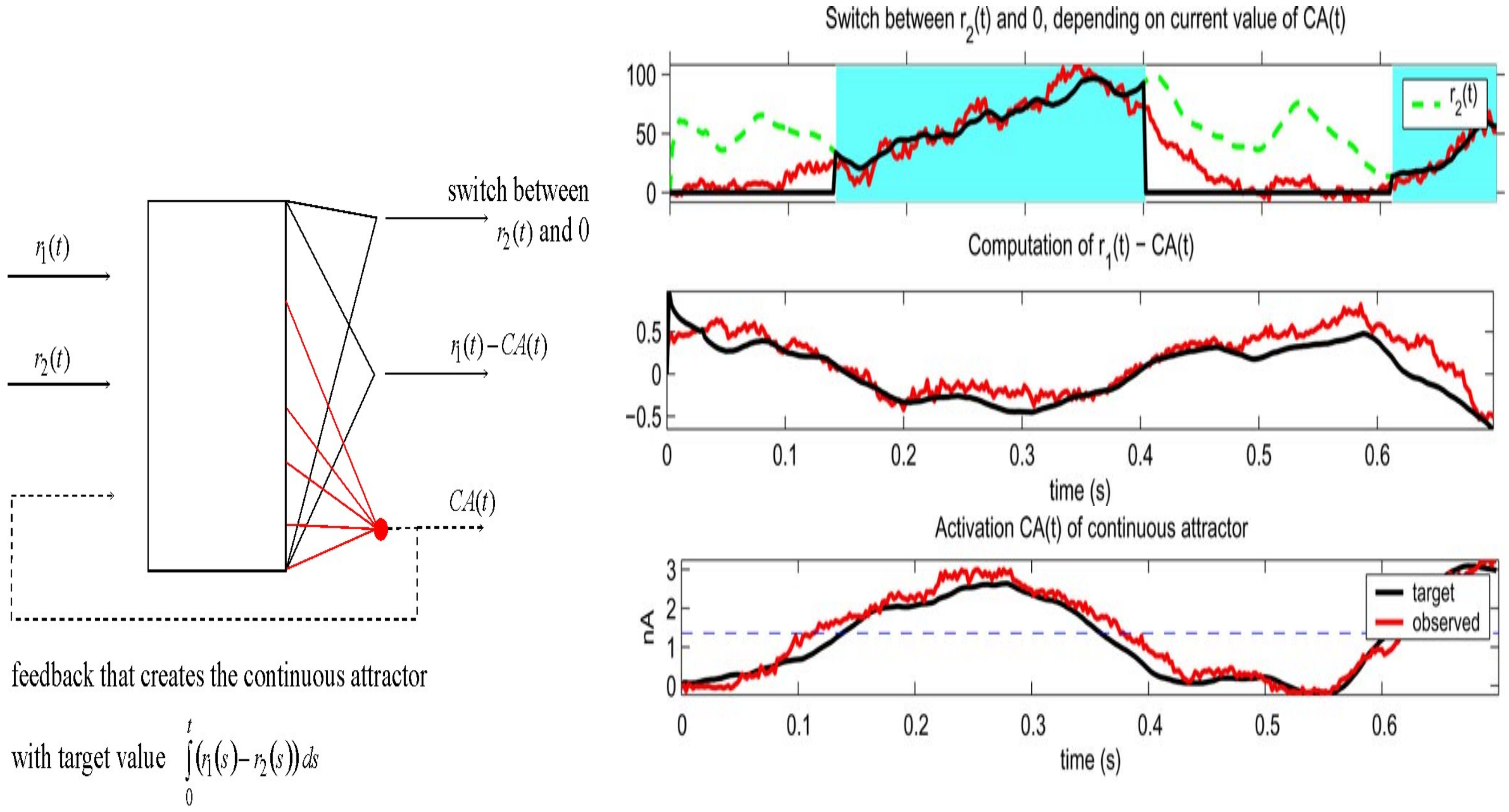
- During training their actual feedback was replaced by a noisy version of their target output $K(x(t))$ („teacher forcing")

Note: In this way they were automatically trained to correct errors resulting from noise and imprecision in their previously given feedback

Computer simulation with 600 noisy HH-neurons (and dynamic synapses) :



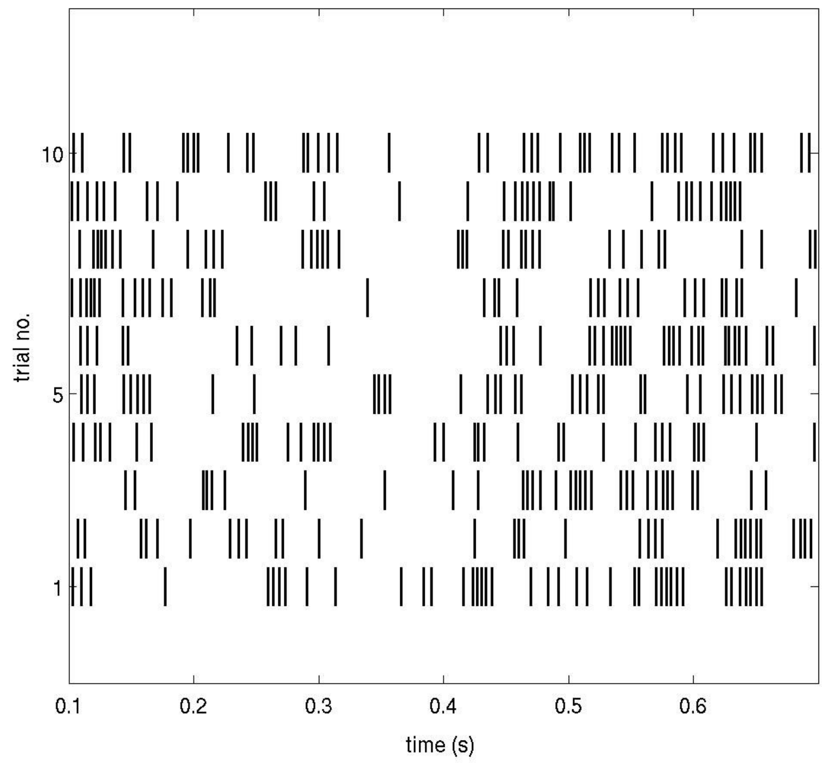
The continuous attractor CA(*t*) was trained to approximate $\int_0^t (r_1(s) - r_2(s))\, ds$



Results shown are for test inputs that had not been used for training

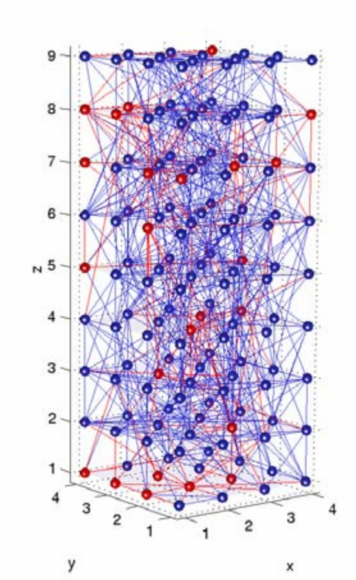
To implement a continuous attractor is quite non-trivial for such circuit, because of its ***in-vivo like*** high ***trial-to-trial variability*** (due to realistic background noise applied to each neuron):

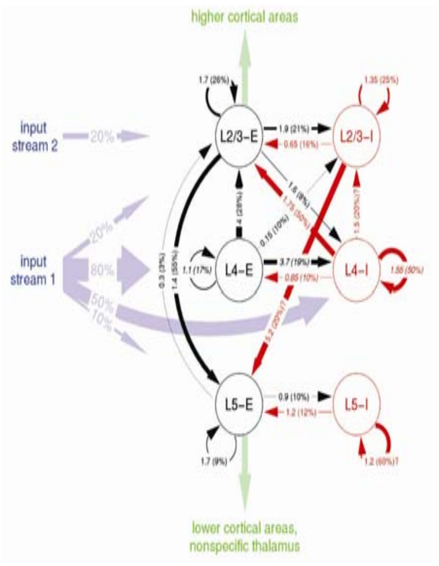
This is the firing activity of a single (randomly selected) neuron in the circuit for 10 trials with the same circuit input:

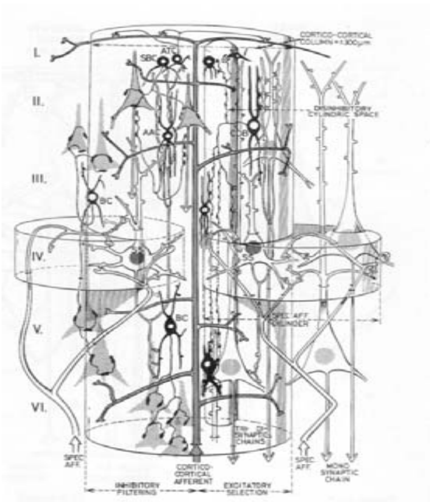


4. Does a more detailed cortical microcircuit model perform better ?

[Haeusler and Maass, Cerebral Cortex 2006]

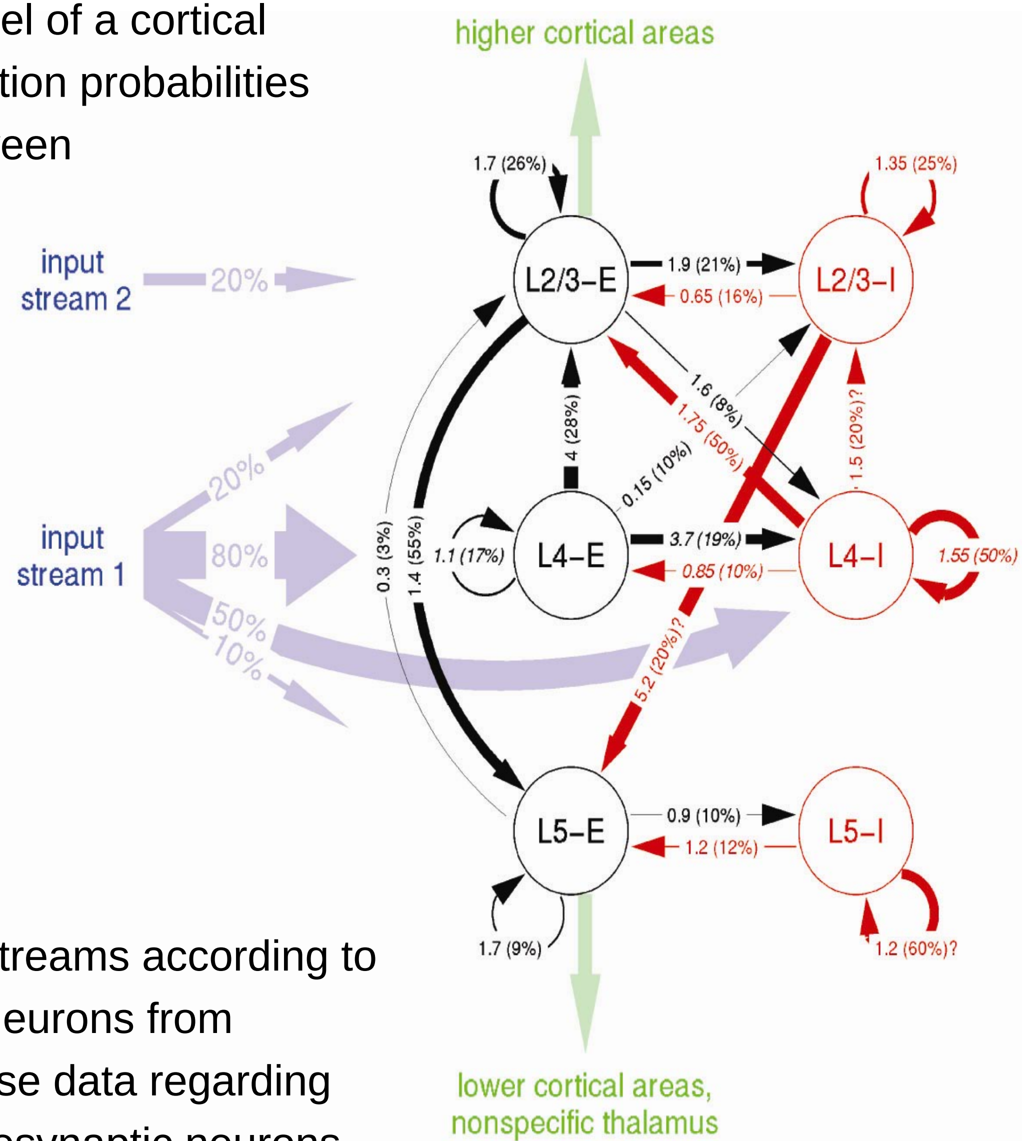






We have built a computer model of a cortical microcircuit, where the connection probabilities and connection strenghts between 6 populations of neurons are chosen according to the data by Alex Thomson et al., Cerebral Cortex, 2002

(who carried out intracellular recordings from 998 identified pairs of neurons).



We distribute two spike input streams according to these data, and train readout neurons from layers 2/3 and 5, based on these data regarding the distribution and signs of presynaptic neurons

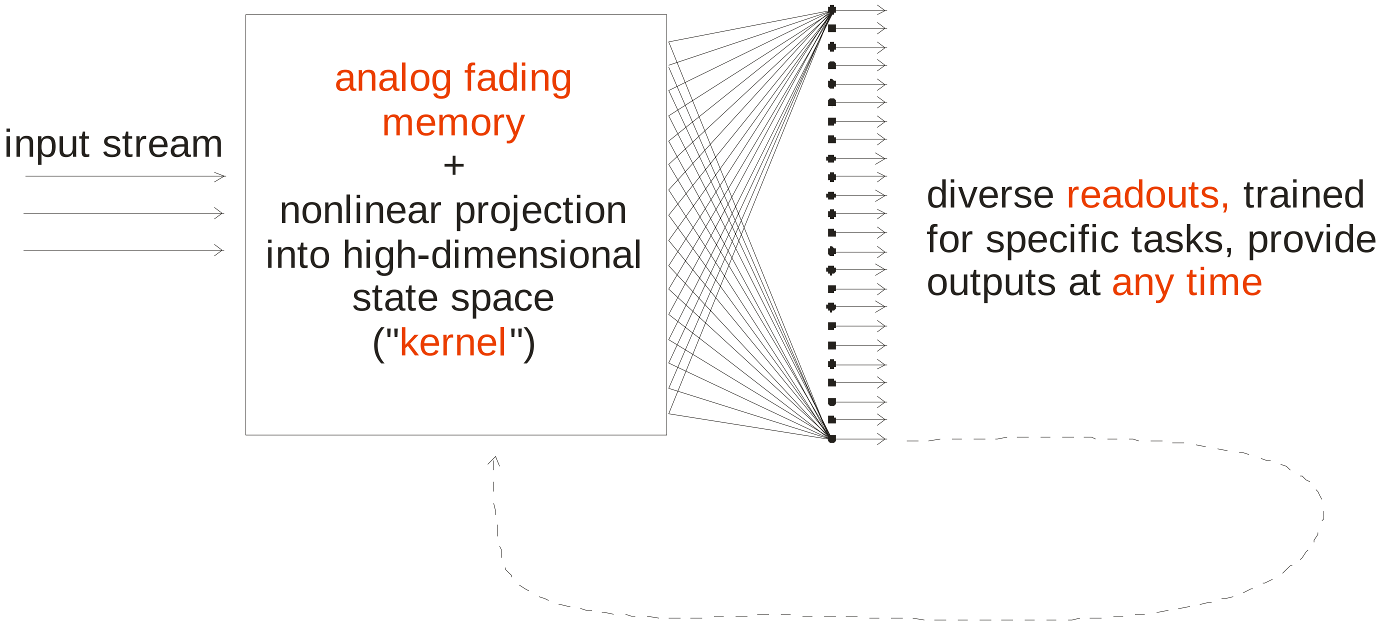
Loss in computational performance (in percent) if this detailed biological microcircuit model is replaced by various types of control circuits (with the same number of components):

tasks/circuits	amorphous	small-world	degree-controlled	degree-controlled w/o input or output specificity	random short term synaptic dynamics	static synapses
memory	32.6	41.6	12.0	35.8	48.3	65.7
non-linear	36.9	11.3	-2.3	4.6	40.1	38.7
other	12.2	5.3	-0.6	5.6	14.1	6.9
all	25.0	15.4	1.6	12.0	30.4	30.6

We tested these different types of circuits on 7 computational tasks, requiring temporal integration of information from spike patterns, nonlinear computations on spike patterns, linear and nonlinear computations on time-varying firing rates from the two input streams.

Result: The laminar structure, the dynamics of synapses, and the biologically found assignment of synapse types are essential for the computational performance of the circuit (for these 7 tasks).

5. Is the main hypothesis biologically realistic ?



Predictions of this model: Generic cortical microcircuits exhibit

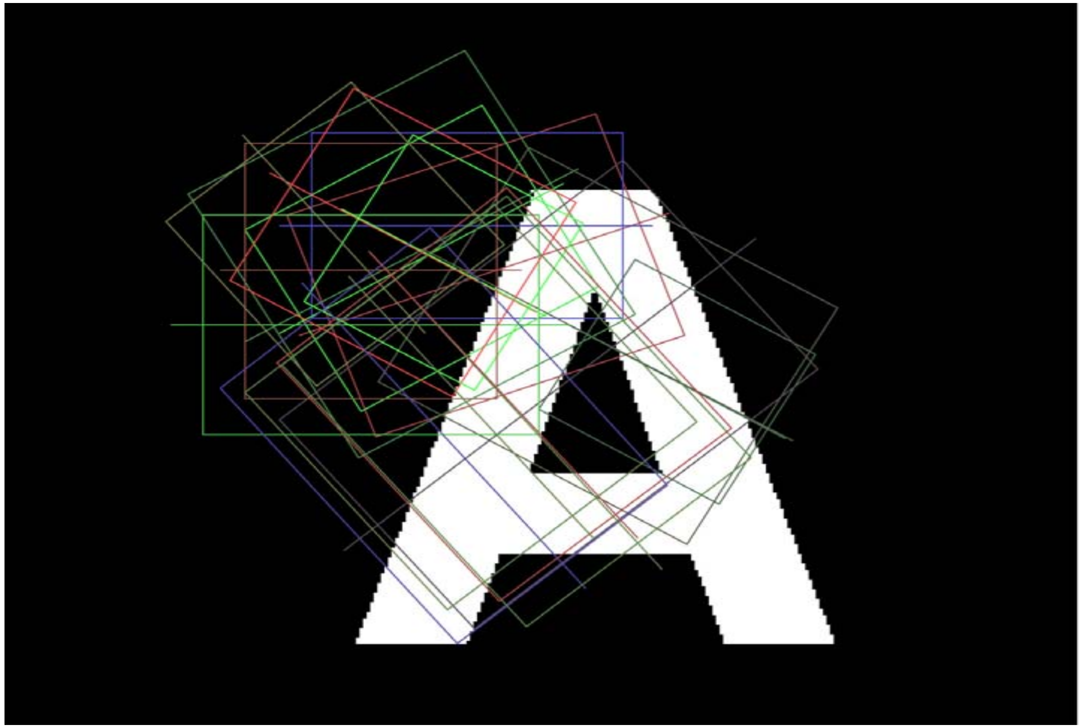
a) Temporal integration
b) Nonlinear preprocessing (kernel)
c) Diversity of readouts

Ad a): Temporal integration

Danko Nicolic at the MPI for Brain Research in Frankfurt has recently started to test systematically the temporal integration property of cat visual cortex:

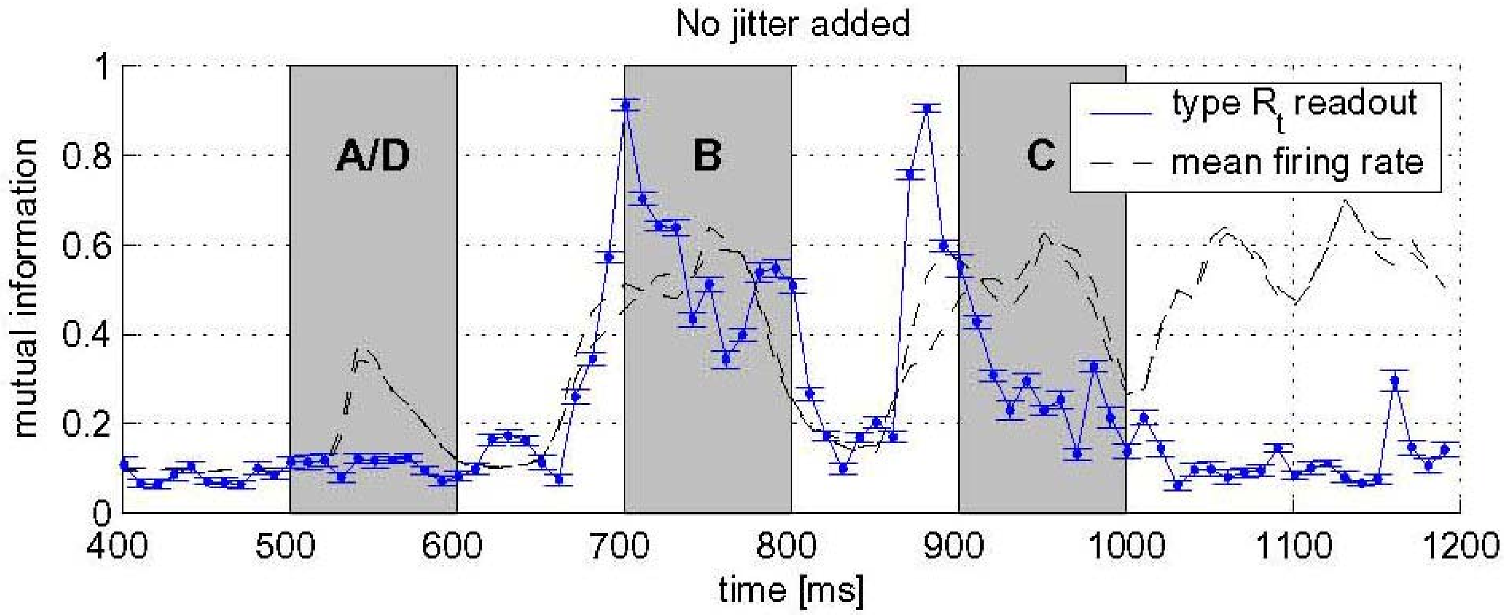
He used sequences of letters as stimuli for anaestesized cats, and recorded from 31 electrodes in area 17

Typical distribution of receptive fields of neurons relative to a stimulus



Time course of information about the first letter A/D (measured every 20ms) in the recorded spike trains:

[Data analysis by Stefan Haeusler and WM]



Ad b) Nonlinear preprocessing (kernel)

It is a somewhat surprising fact that the labs of Nicolelis, Poggio, and Schwartz all report just very small performance improvements for various motor control and object recognition tasks if (artificial) linear readouts from multi-unit recordings are replaced by nonlinear readouts.

Hence one may argue that the neural systems from which they record have in fact kernel capabilities.

Ad c) Diversity of readouts

I am not aware of direct experiments which have tested that (one should for example record simultaneously from two different pyramidal cells on layers 2/3 or 5/6 from the same column).

Some unpublished data from recordings with several tetrodes in area V1 of awake monkeys (natural stimuli) show that the correlation between neurons picked up by the same tetrode are not more correlated than neurons at a larger distance.

6. Discussion

My hypothesis from the beginning:

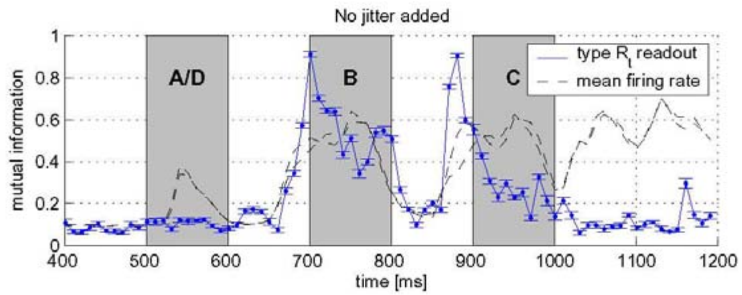
„The computational task of a typical cortical microcircuit is to support (as a general preprocessor, e.g. via temporal integration and nonlinear operations) diverse computational goals of a variety of readout neurons"

I have shown in this talk that

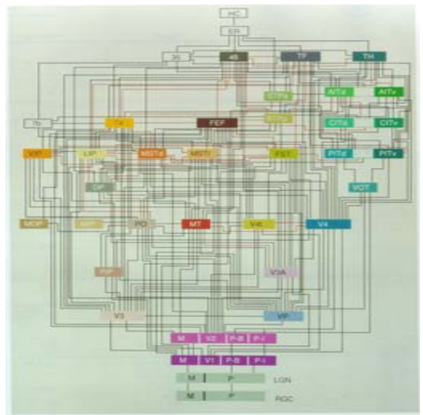
- Generic cortical microcircuit models do in fact have such general preprocessing qualities; in particular they support anytime computing, and they even can support many different computations simultaneously
- This approach yields a new method for analyzing the computational function of specific details of cortical microcircuits (such as laminar structure; specific synapse types)
- The resulting computational model is in principle powerful enough to carry out all computations that a brain might need to execute; including real-time computations that combine internal states with external information

I have also shown

- results of a first direct test of the predicted temporal integration property (data by Danko Nicolic)



Remark: One can in principle use a similar set-up to track the temporal dynamics of information simultaneously in several brain areas, and in communication streams between them. Such experiments are likely to provide insight into the large-scale computational organization of the brain.



We need:

Further biological experiments which directly test predictions of our model for the computational role of a generic cortical microcircuit (temporal integration, kernel capability, and diversity of readouts); both for isolated cortical circuits and for multiple brain areas.

We also need:

---Theoretical (and simulation) studies which examine how unsupervised learning and self-organization in generic cortical circuits can optimize the function of a generic cortical microcircuit as generic preprocessor for multiple neural readouts.

New results of Robert Legenstein and WM show that more realistic models for neural readouts (with sign constraints) require additional preprocessing qualities of the circuit (e.g. sparse activity in a small dynamic range).

--- Studies which examine whether further details of cortical microcircuits (such as those collected in the European FACETS-Project) enhance their generic preprocessing capabilities

--- A theoretical result which clarifies under what conditions intrinsic activity patterns of cortical circuits ("cortical songs", synfire chains, avalanches) can implement the "separation property" which is needed to approximate arbitrary Volterra series by such circuits.