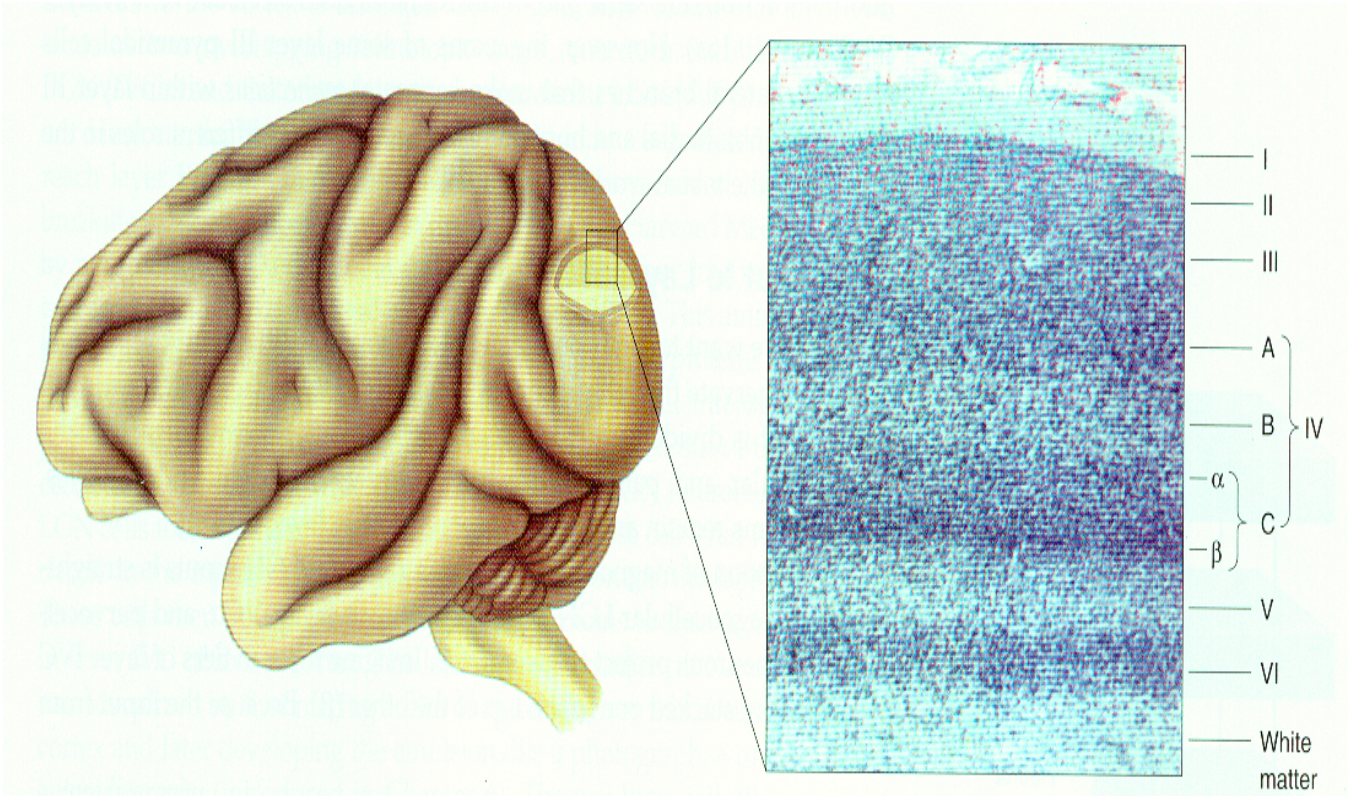
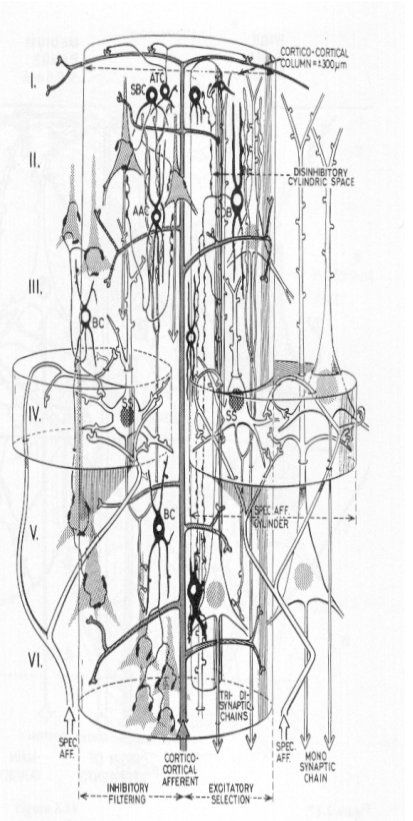
How could Networks of Neurons Learn to Carry Out Probabilistic Inference ?

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

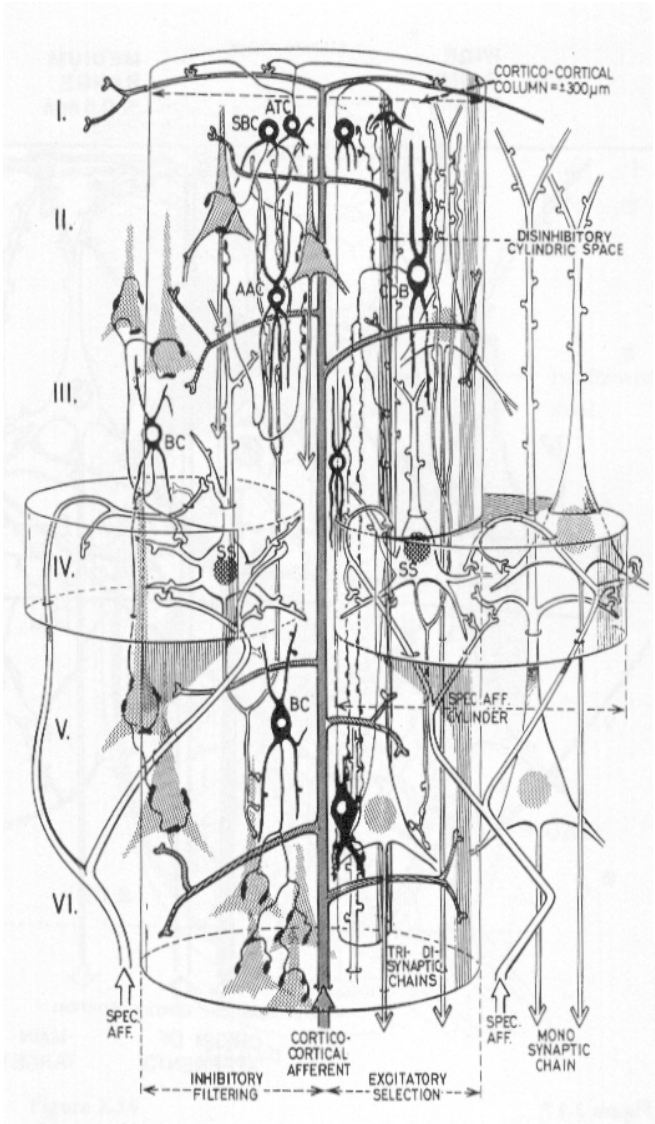
The computational machinery of the cortex consists of a pizza-size 2-mm thick sheet of more or less generic „cortical microcircuits“

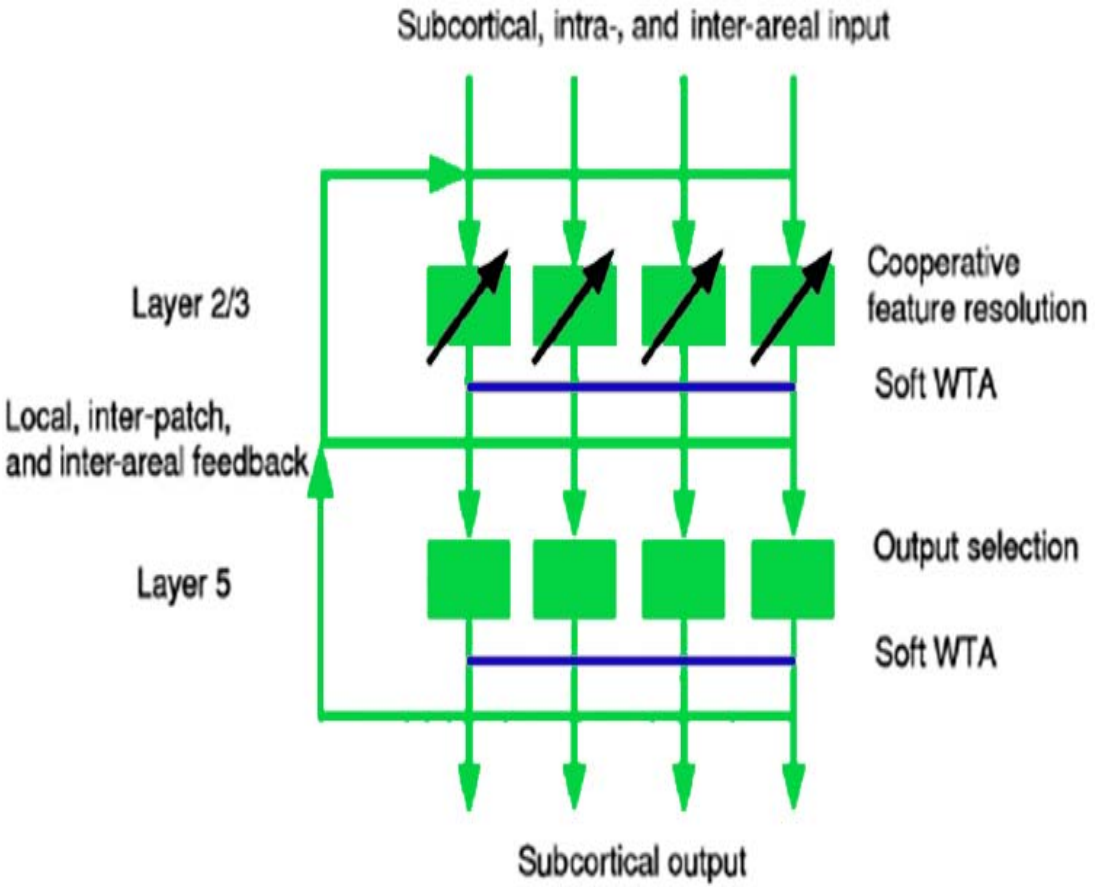
If the brain applies probabilistic inference, these cortical microcircuits are likely to provide a generically useful module for that. Of what nature could that module be ?





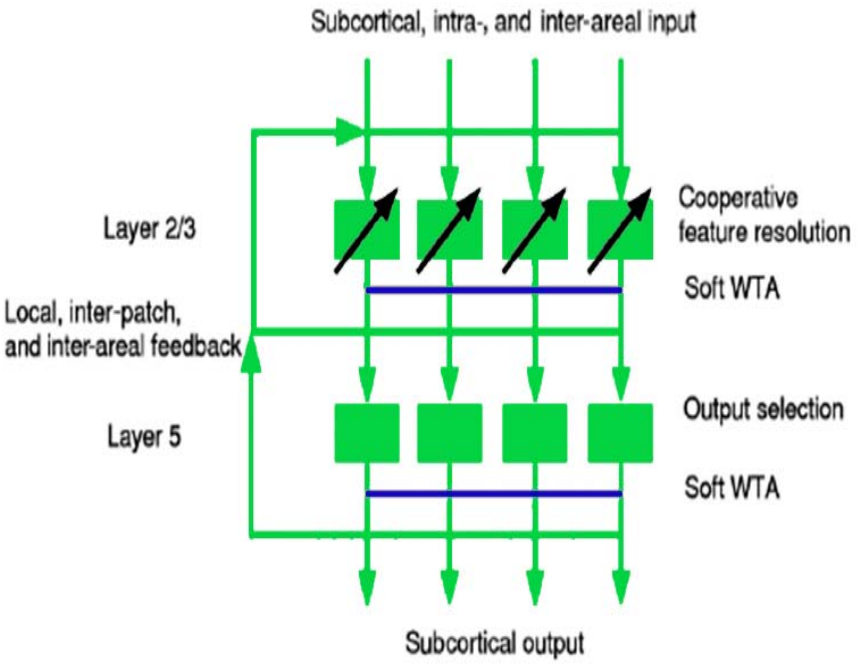
Local structure in generic cortical microcircuits





[Douglas and Martin, 2004]:
„canonical microcircuit“ of the cortex

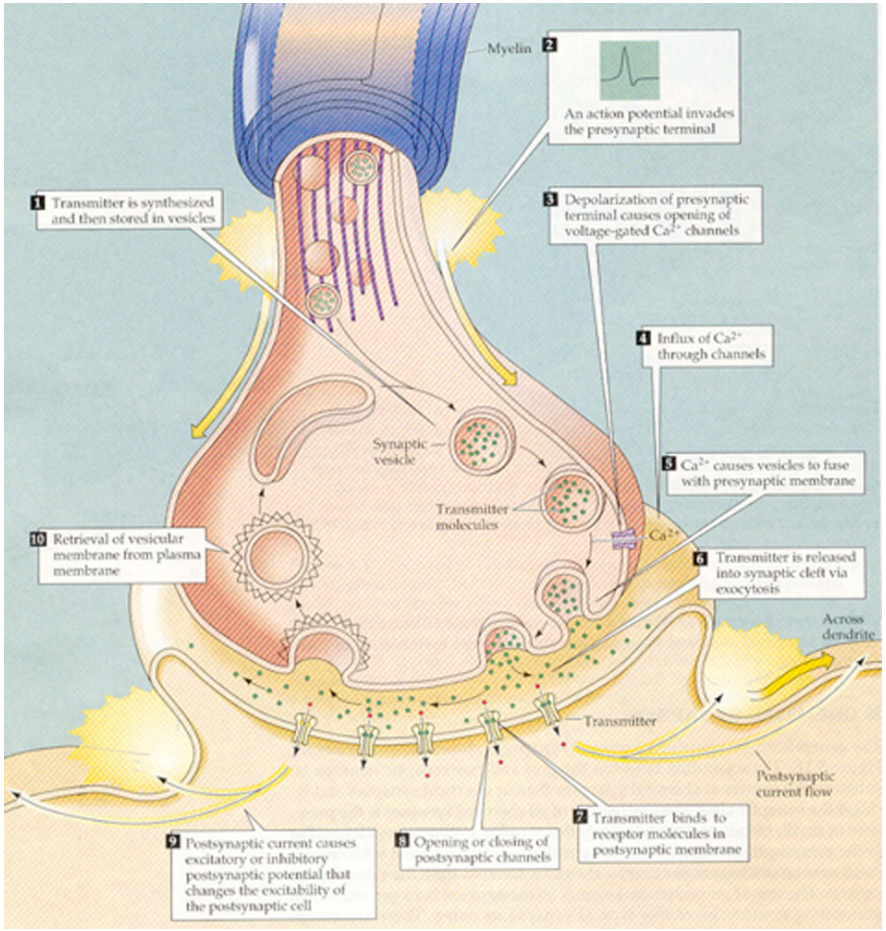
These generic computational modules would either have to be genetically encoded to carry out probabilistic inference, or automatically acquire this capability through plasticity



Experimental data suggest that these „soft WTA-circuits“ in the cortex are stochastic:

- Spontaneous activity
- Large trial-to-trial variability

Problem for understanding synaptic plasticity: Synapses are very complicated devices, and their plasticity is only partially understood



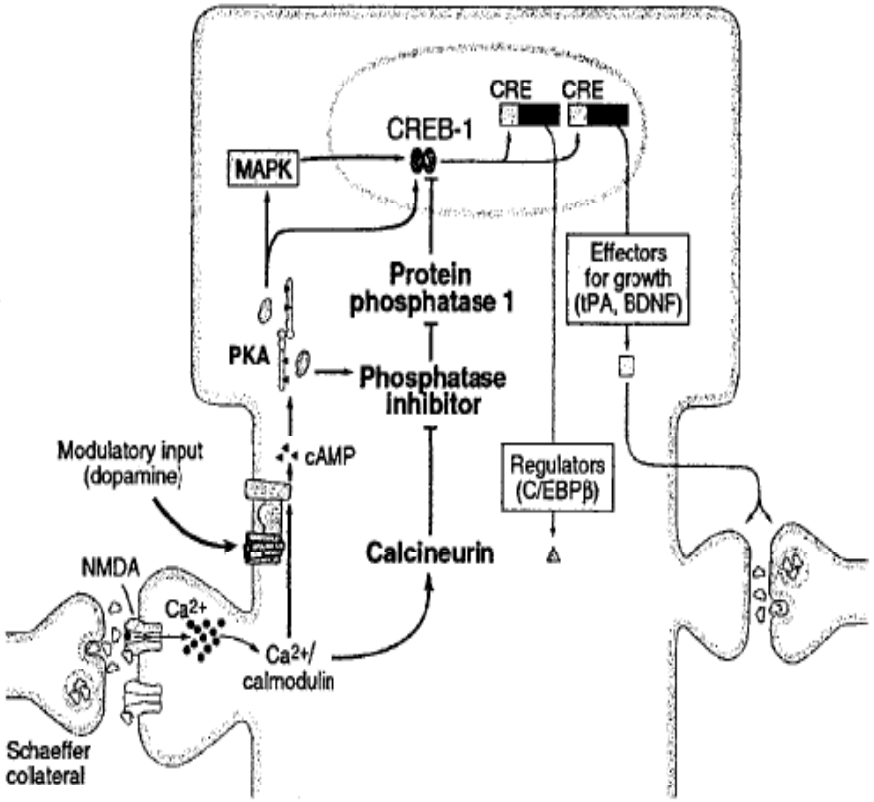
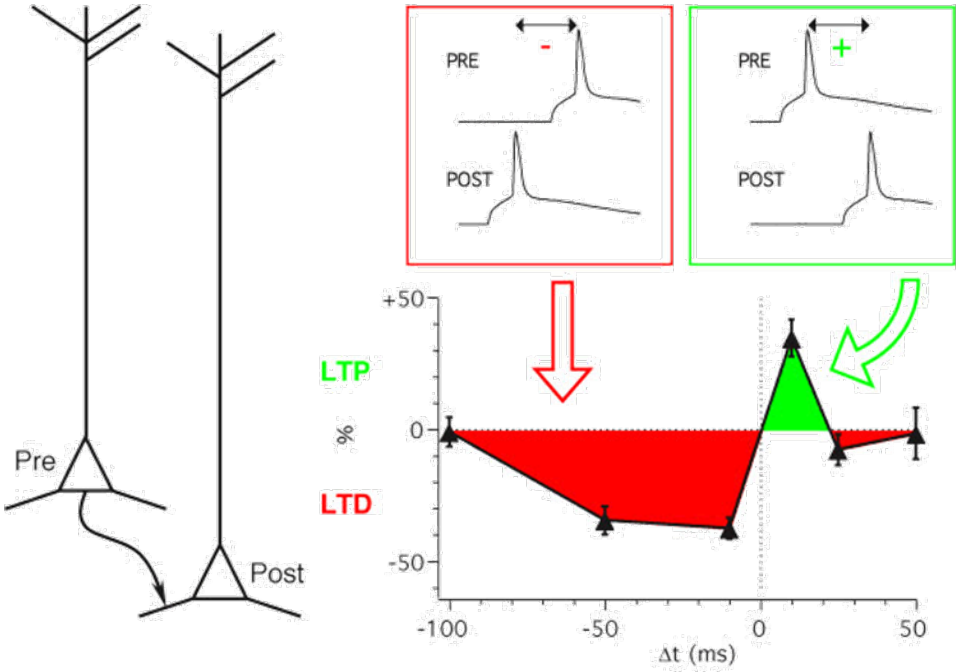


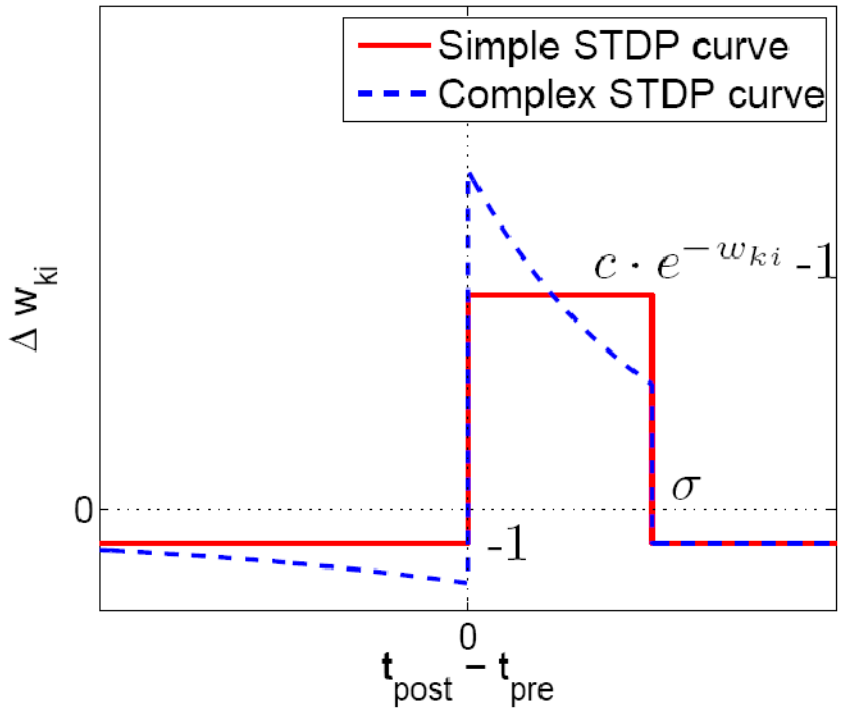
Fig. 30. Long-term potentiation requires regulation not only of kinases but also of phosphatases. The phosphatase cascade initiated by calcineurin shuts off a phosphatase inhibitor and thereby disinhibits the protein phosphatase, which can now inhibit the kinase cascade. [Based on 92.]

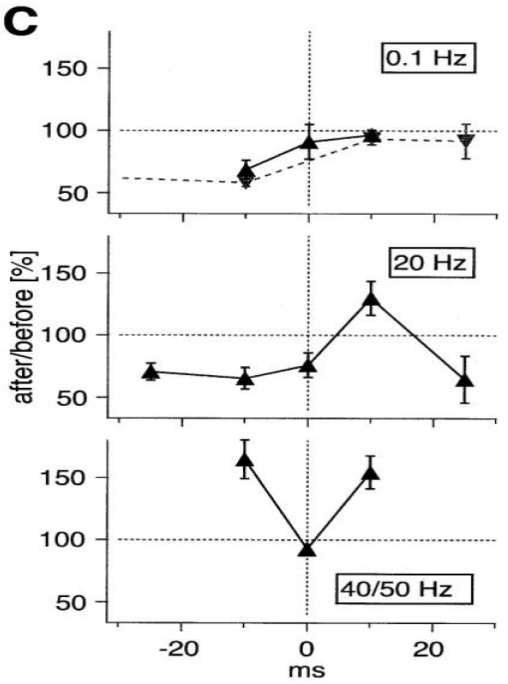
STDP (= Spike-Timing-Dependent plasticity) is currently the best understood experimental method for inducing synaptic plasticity



The key mechanism of STDP is the interaction of the incoming EPSP with the backpropagating action potential (BAP). The amplitude of the BAP is modulated by neuromodulators, as well as other inputs to the same neuron. One assumes that the LTP- and LTD-parts of STDP are implemented by separate molecular mechanisms.

STDP curves that were used for our computer experiments





These STDP curves are qualitatively similar to the data of [Sjöström et al., 2001] for a rate of 20 Hz.

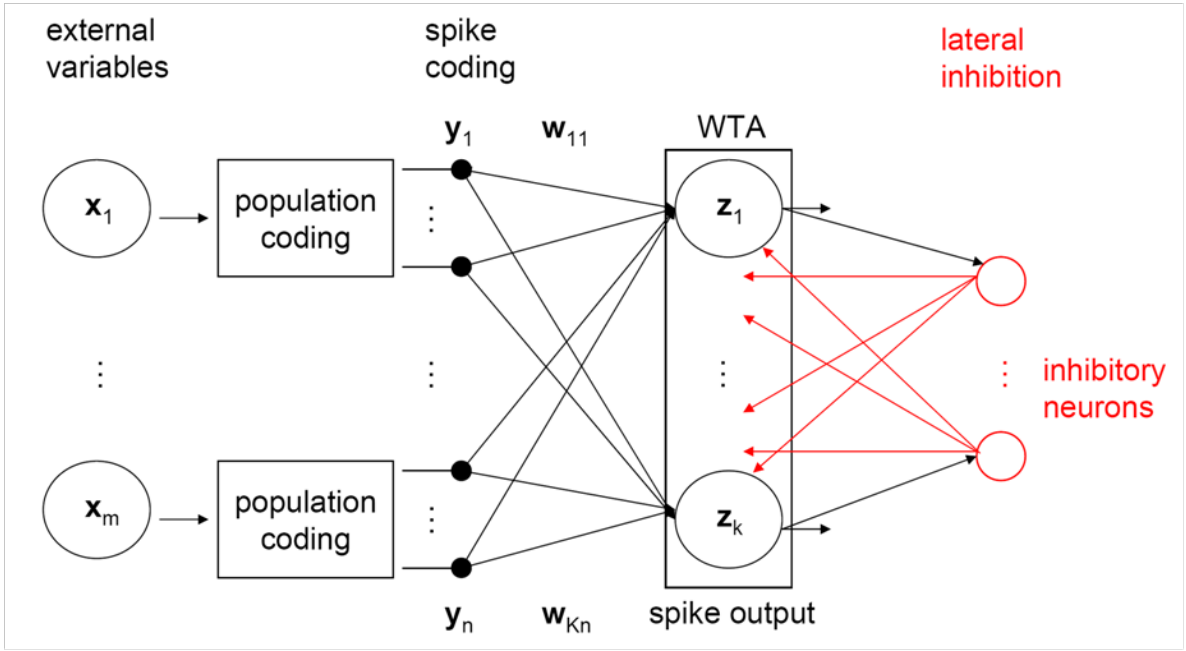
There are other kinds of use-dependent changes in neurons

Example:

Use-dependent adaptation of the intrinsic excitability of a neuron: When a neuron is made to fire for a number of times, its excitability may increase (i.e., it fires with less excitatory input).

We included a rule for the adaptation of intrinsic excitability in our model.

Result: STDP induces implicit generative models in the z-neurons of a stochastic-WTA circuit



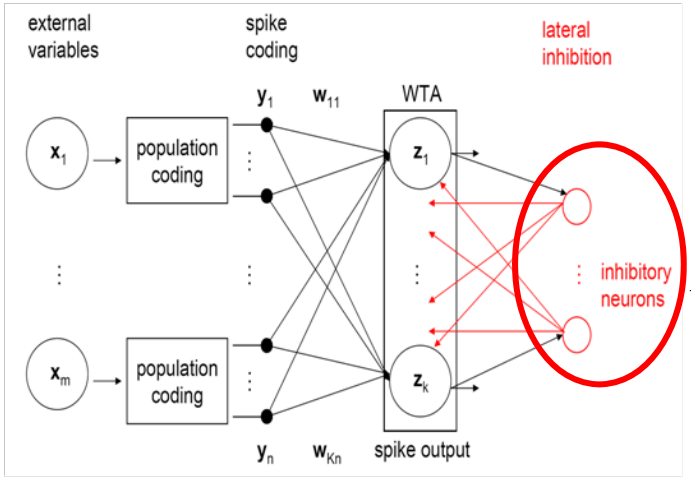
$$p(z_k \text{ fires at time } t|\mathbf{y}) = \frac{e^{u_k(t)}}{\sum_{l=1}^{K} e^{u_l(t)}}$$

Membrane potential of this neuron:

$$u_k(t) = \sum_{i=1}^{n} w_{ki}\tilde{y}_i(t) + w_{k0}$$

This exponential firing rule fits experimental data quite well [Jolivet et al., 2006]

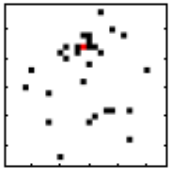
Role of lateral inhibition in this context: it implements the normalization

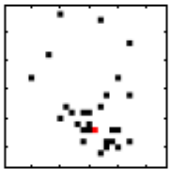


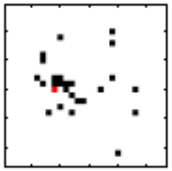
$$p(z_k \text{ fires at time } t|\mathbf{y}) = \frac{e^{u_k(t)}}{\sum_{l=1}^{K} e^{u_l(t)}}$$

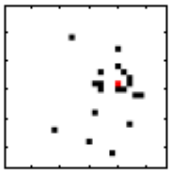
Demonstrating the possibility of this idea in a concrete example:

We encode in the spike input to the circuit
spatial patterns that are
generated by a hidden generation process



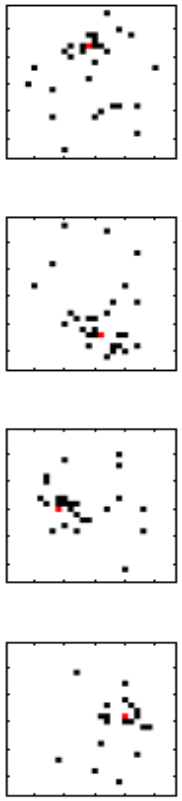




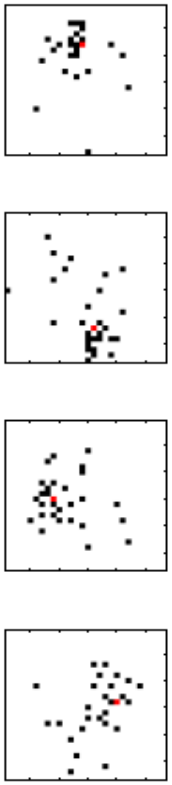


Uncovering for you *(but not for the circuit)*
the hidden process which generates these input patterns:
Gaussians with different centers in 2D, with priors 0.1, …, 0.4

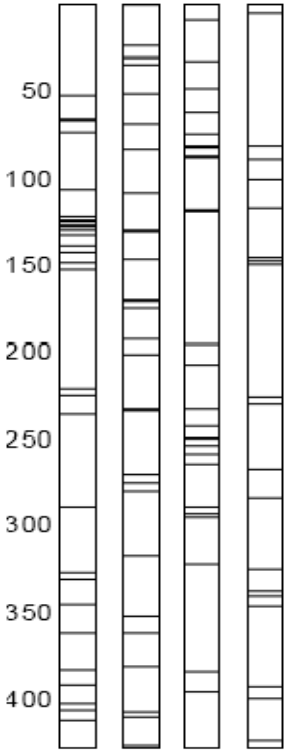
4 samples from the 4 Gaussians



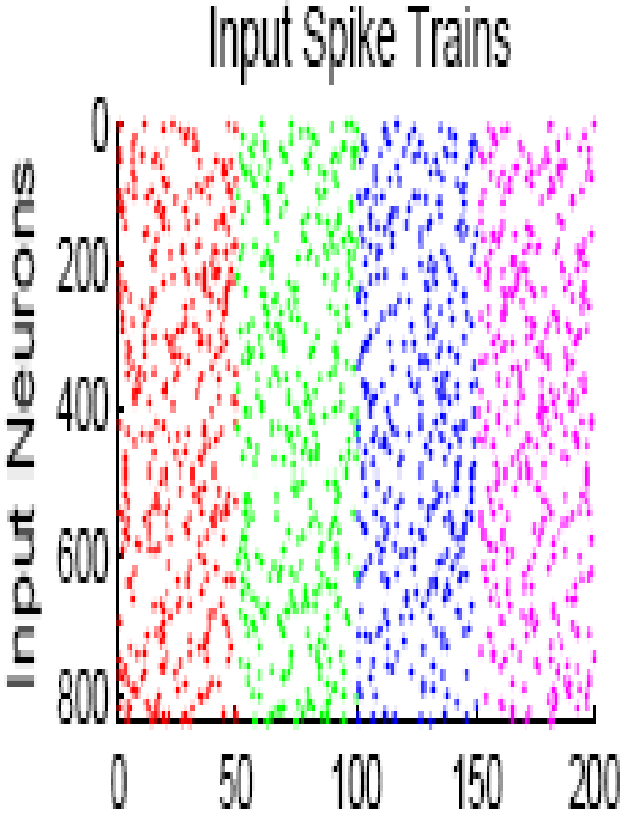
4 further samples, generated in the same way



each sample is transformed into a linear array

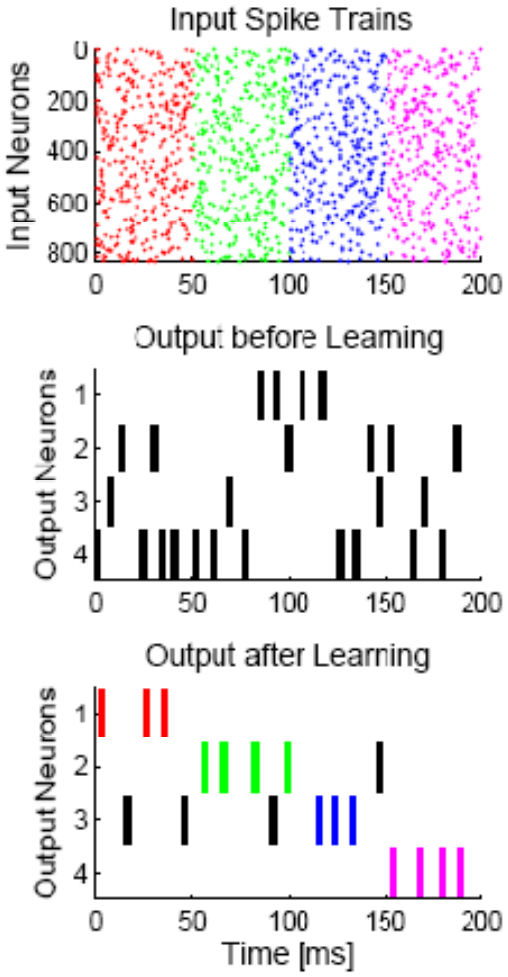


resulting spike input for these 4 samples (each pixel encoded by 2 spike trains)

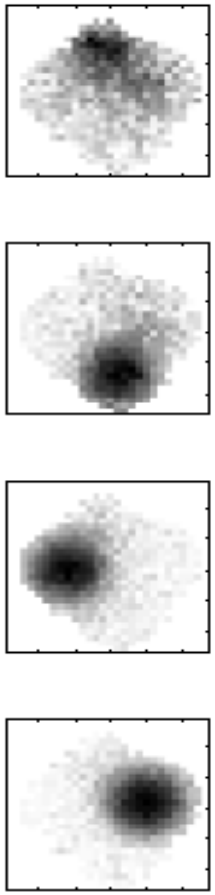


Output of the 4 z-neurons at the beginning, and after having seen a 20 s stream of such input spike trains

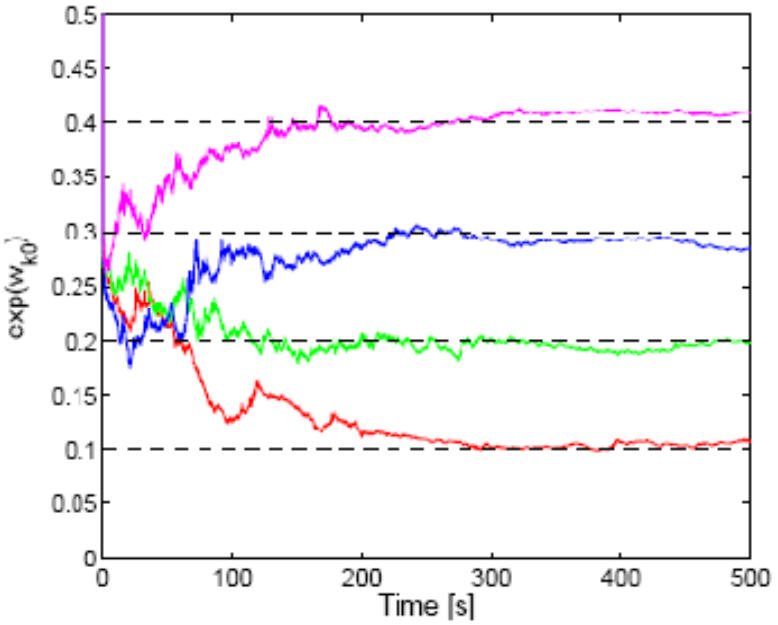
(while STDP and excitability adaptation are active)



Weight vectors of the 4 z-neurons (projected back into the 2D input space)



Autonomous adaptation of neuronal excitability

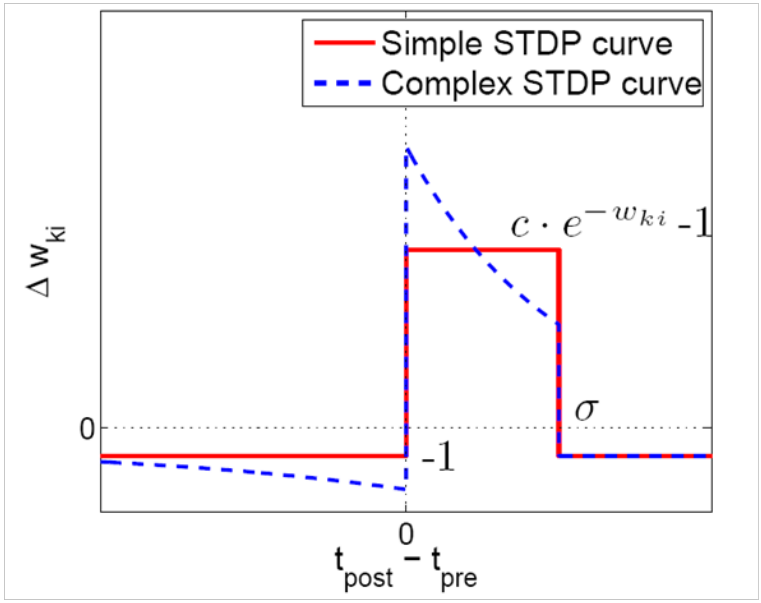


$$p(z_k \text{ fires at time } t|\mathbf{y}) = \frac{e^{u_k(t)}}{\sum_{l=1}^{K} e^{u_l(t)}}$$

$$u_k(t) = w_{k0} + \sum_{i=1}^{n} w_{ki} \cdot \tilde{y}_i(t)$$

Link between the STDP rule that was applied, and probability theory

The STDP rule



causes each synaptic weight to converge (in fact, optimally fast) to the log conditional probability

log p(presyn. neuron has fired just before time t / postsyn. neuron fires at time t)

(These log probabilities are shifted into the positive range by adding a constant term.)

Result: **This simple stochastic WTA-circuit learns through STDP (and adaptation of neuronal excitability) to carry out Bayesian inference**

More precisely: it learns to implement Bayes Theorem

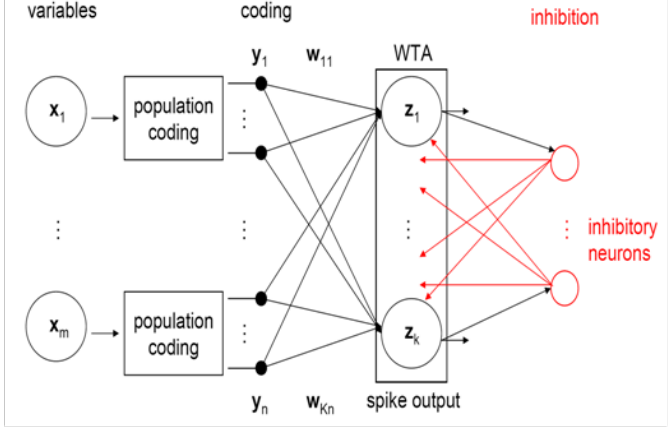
We have $$p(z_k \text{ fires at time } t|\mathbf{y}) = \frac{e^{u_k(t)}}{\sum_{l=1}^{K} e^{u_l(t)}}$$

where the membrane potential is defined by $$u_k(t) = w_{k0} + \sum_{i=1}^{n} w_{ki} \cdot \tilde{y}_i(t)$$

If the weights and the bias encode suitable log-probabilities, then the summation in this membrane potential implements multiplication of probabilities (in the log-domain).
In particular: This can implement the multiplication in Bayes Theorem:

Resulting posterior distribution $$p(k|\mathbf{y}, \mathbf{w}) = \frac{p(k|\mathbf{w}) \cdot p(\mathbf{y}|k, \mathbf{w})}{\sum_{k'=1}^{K} p(k'|\mathbf{w}) \cdot p(\mathbf{y}|k', \mathbf{w})}$$

Analysis of the implicit generative model that can be implemented through the weights of this WTA circuit



Joint probability that the k-th output neuron fires for spike-input **y :**

$$p(\mathbf{y}, k|\mathbf{w}) = \frac{1}{C} e^{u_k}$$

Marginalization yields:

$$p(\mathbf{y}|\mathbf{w}) = \frac{1}{C} \sum_{k=1}^{K} e^{u_k}$$

where

$$u_k(t) = w_{k0} + \sum_{i=1}^{n} w_{ki} \cdot \tilde{y}_i(t)$$

Rewriting this term as

$$p(\mathbf{y}|\mathbf{w}) = \sum_{k=1}^{K} \left[\pi_k \cdot \prod_{j=1}^{m} \prod_{i \in G_j} \mu_{ki}^{[x_j = v(i)]} \right]$$

reveals a mixture distribution, where $\mu_{ki} = e^{w_{ki}}$ is the probability that x_j assumes the value $v(i)$ in the k^{th} multinomial of this mixture of K multinomials, with mixture coefficients $\pi_k = e^{w_{k0}}$.

When the inputs are generated by a mixture of monomials, and are encoded by spike trains as in the preceding example, these are those values to which the weights converge under STDP:

log p(presyn. neuron has fired just before time t / postsyn. neuron fires at time t)

More abstract analysis of the behaviour of STDP, even when the input distribution is not a mixture of multinomials:

STDP approximates stochastic online EM for fitting the implicit internal model (a mixture of multinomials) to the actual distribution of spike inputs to the circuit

STDP is applied only to the synapses of that z-neuron which fires. This application of STDP increases the chance, that this neuron would fire again for the same input, hence it corresponds to the *M-step of EM* (one can prove rigorously that it makes one step in the right direction).

Each change of a synaptic weight modifies the resulting guesses for the latent variables. The *E-step* simply consists of applying for the next spike input **y** the WTA-network with these slightly changed weights. In other words, in lack of a teacher for supervised learning, the network uses the guess provided by the current state of the WTA-circuit as a substitute teacher.

The general theory of EM (Expectation Maximization) guarantees that iterations of these E- and M-steps yields convergence to a (local) optimum of the objective function.

We refer to this new unsupervised learning principle for networks of spiking neurons as SEM (spike-based EM)

The objective function that is minimized by this application of SEM through STDP:

The Kullback-Leibler divergence between the external distribution of spike inputs y and the implicit generative model:

$$\mathrm{KL}(p^*(\mathbf{y})||p(\mathbf{y}|\mathbf{w})) = \sum_{\mathbf{y}} p^*(\mathbf{y}) \log \frac{p^*(\mathbf{y})}{p(\mathbf{y}|\mathbf{w})}$$

Each application of STDP makes a move in the direction of the M-step of an application of stochastic online EM for minimizing this KL-divergence.

[Nessler, Pfeiffer, Maass, NIPS 2009] ; journal version in preparation

Bernhard Nessler

Michael Pfeiffer

This unraveling of STDP as a spike-based EM approximation is not limited to mixtures of multinomials as internal models, and it does not require „perfect“ WTA circuits

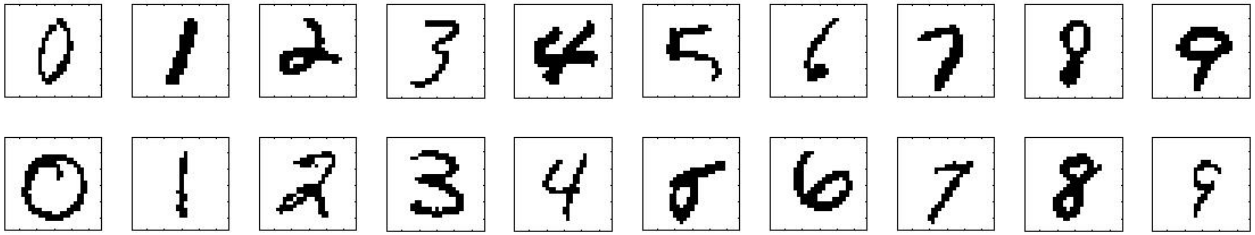
- [Habenschuss et al, in preparation]

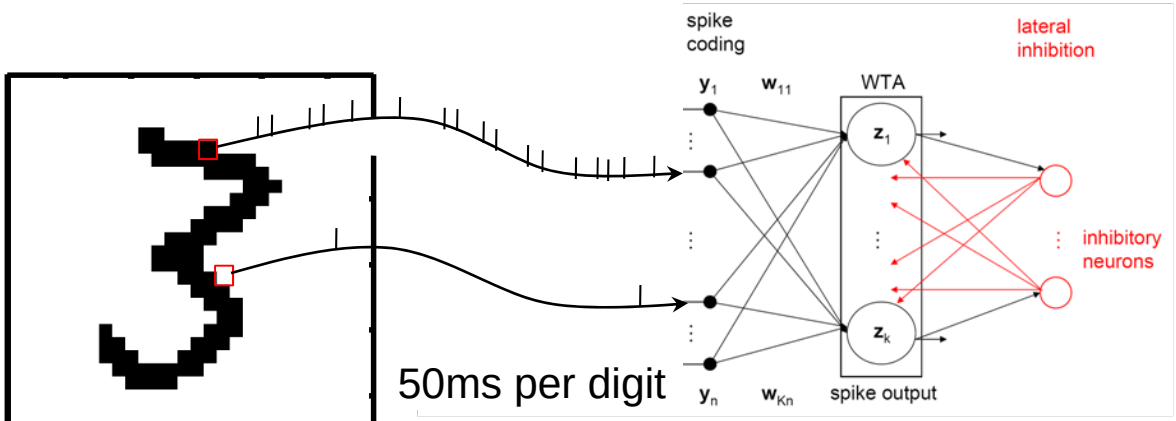
- [Büsing et al, in preparation]

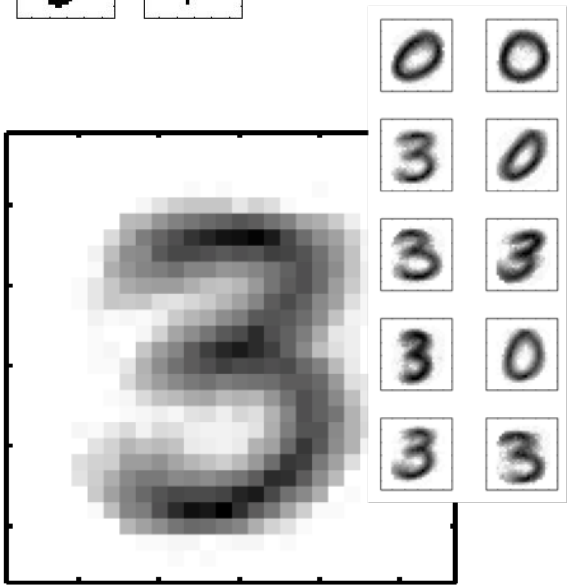
This theoretical understanding of unsupervised learning with STDP makes it possible to generate networks of spiking neurons with quite impressive computational power and learning capability:

Application to a generic machine learning task (but WITHOUT supervision): MNIST dataset

These are 50 random samples from the 70 000 samples in the MNIST dataset.



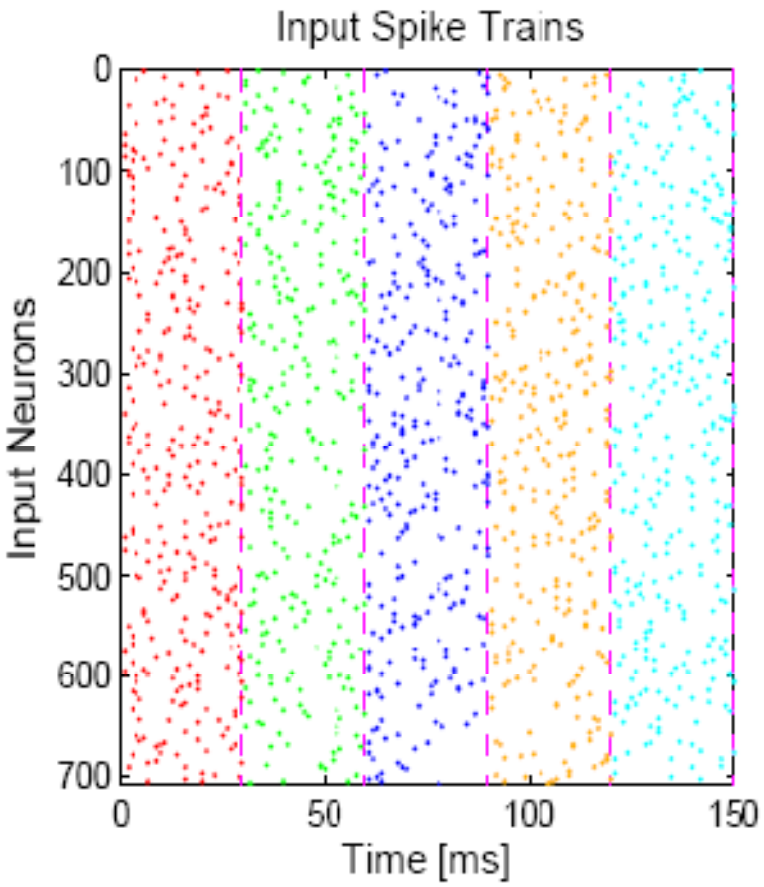




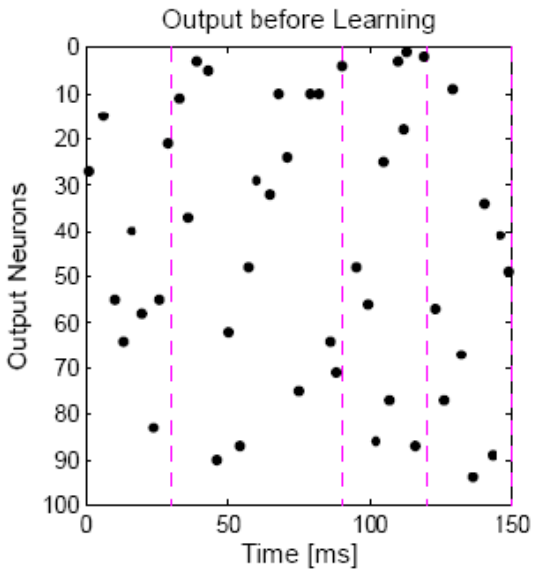
weights of one neuron

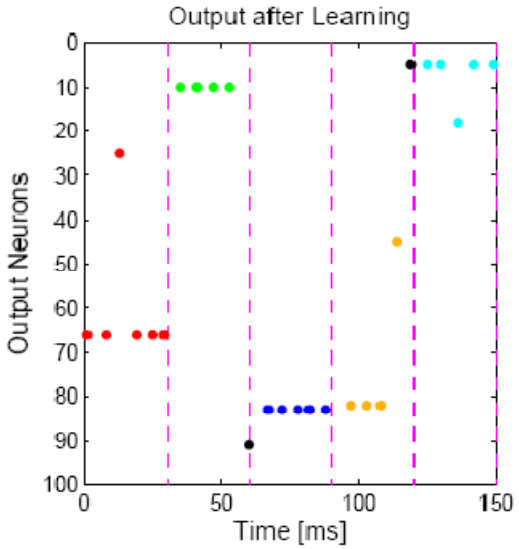
Resulting implicit generative models of 100 z-neurons

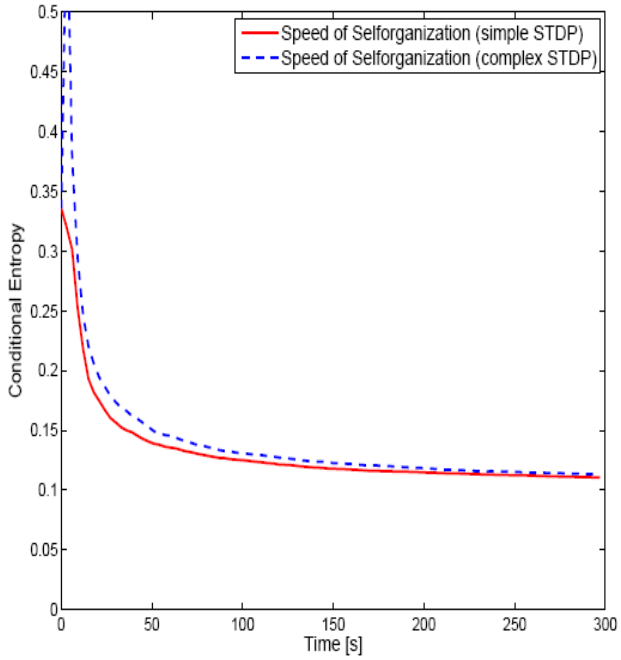
after exposing the circuit to 300 s of spike inputs, where a different sample of a handwritten digit is encoded in each window of 30 ms



Resulting spike output of the WTA-circuit before and after learning





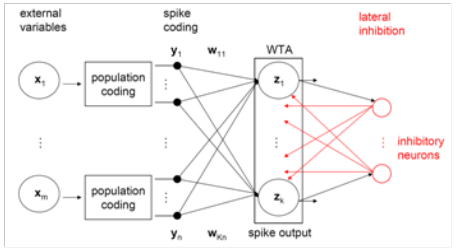


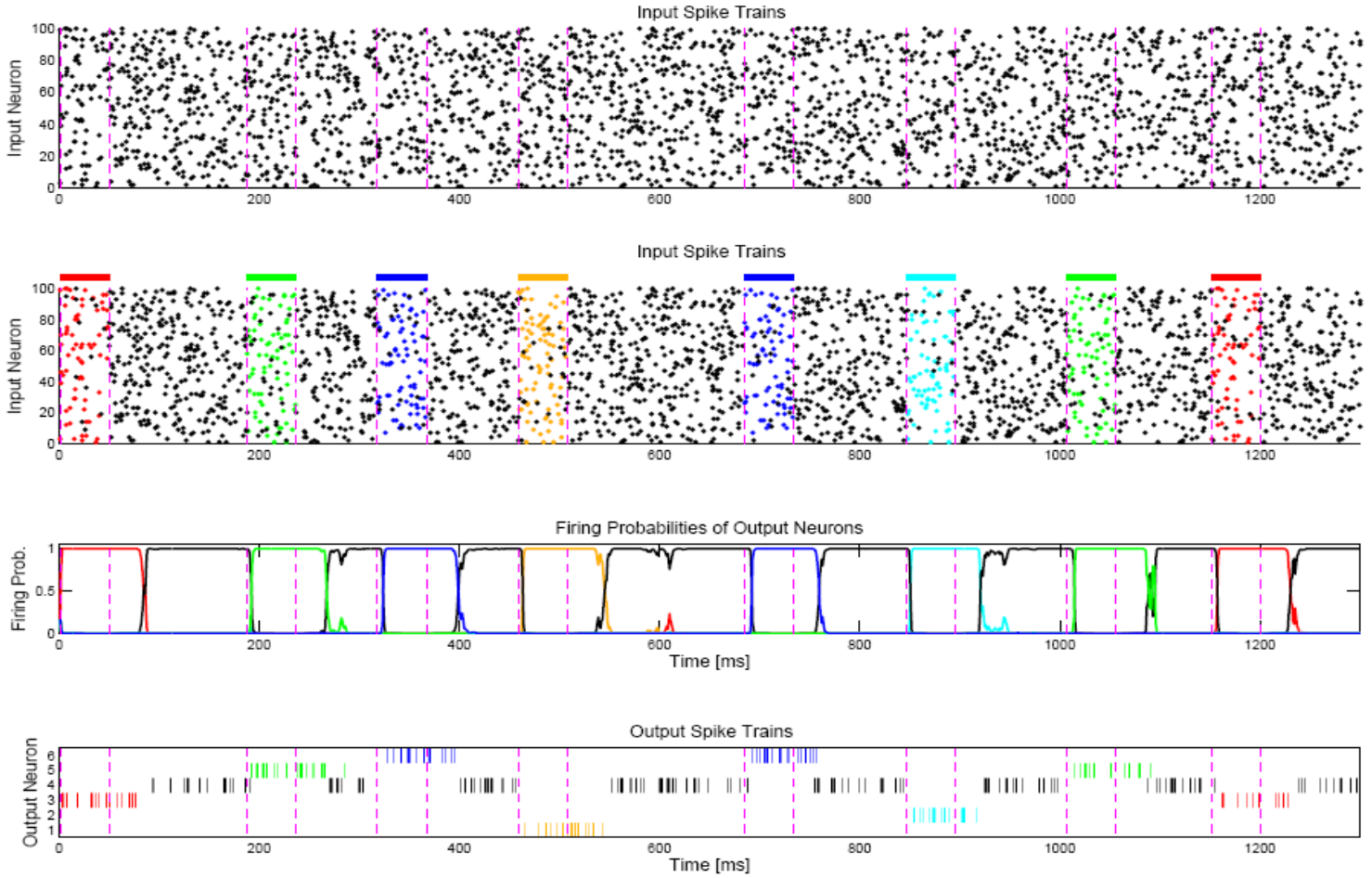
The resulting sparser and more reproducable spike output corresponds to results on perceptual learning in neuroscience (which works without supervision).

Application to a more brain-like discrimination task:

Emergence of detectors for repeating spatio-temporal spike patterns through STDP

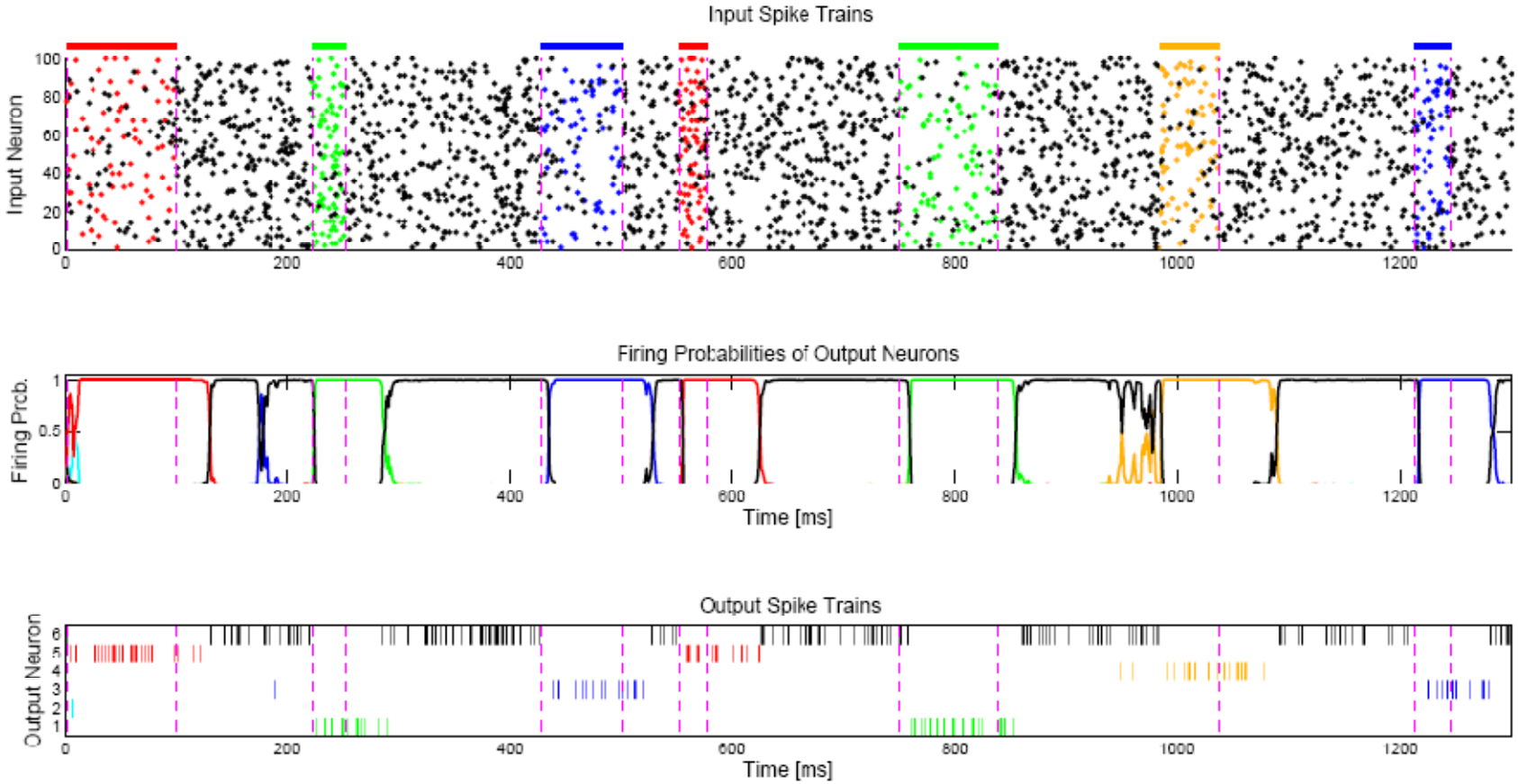
(after 20 s of unsupervised training)



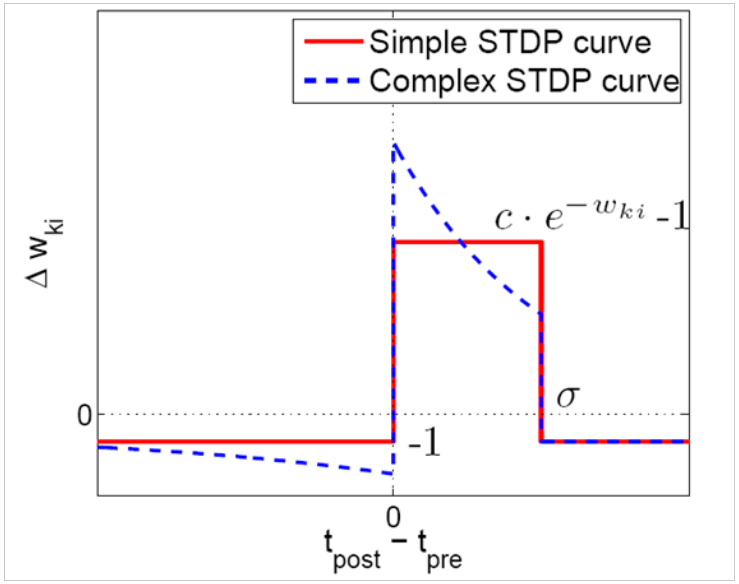


Spike patterns: fixed Poisson spike trains at 15 Hz for 50 ms (colored); they are always superimposed by 5 Hz noise Poisson spike trains (black)

These emerging detectors for spike-patterns automatically generalize to time-warped variations of these patterns



Experimentally testable predictions of those aspects of the STDP rule that are critical for these results, and the underlying theory that STDP approximates EM



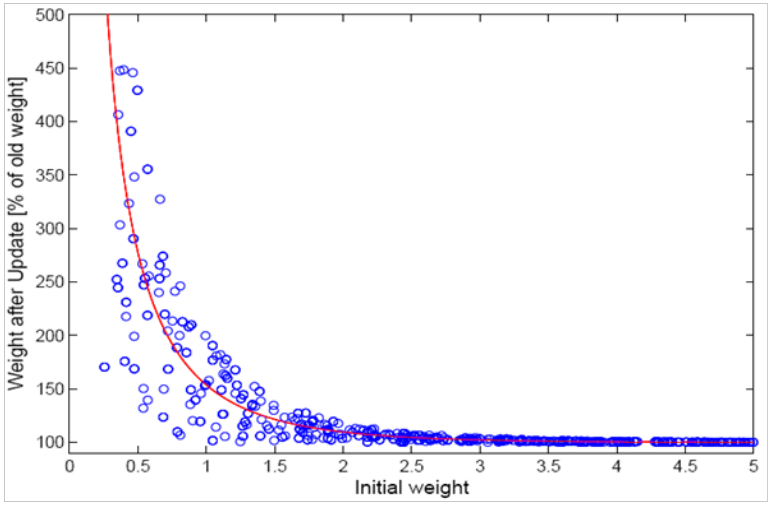
1. Weight increases become exponentially smaller in dependence of the current weight size

2. Weight decreases are independent of the current weight size.

Experimental data confirm both of these two predictions

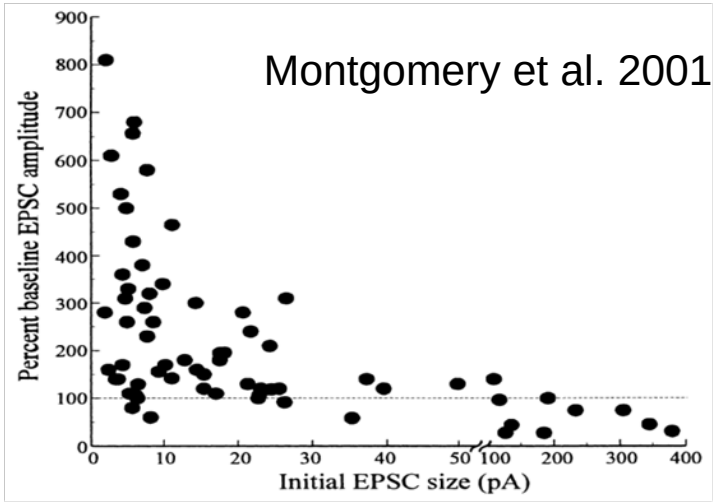
1. Weight increases become exponentially smaller in dependence of the current weight size

Theoretical prediction



Noise does not harm the effectiveness of the STDP rule

Experimental data



See similar data by [Liao et al., 1992], [Bi and Poo, 1998], [Sjöström et al., 2001]

2. Weight decreases are independent of the current weight size.

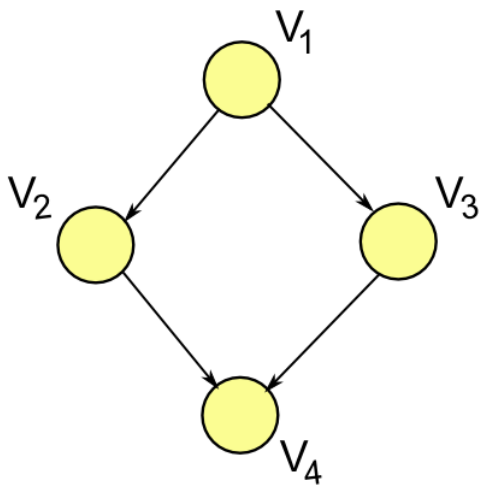
[Jacob et al., J. of Neurophys. ,2007] report that weight decreases of STDP are not correlated with the current weight size

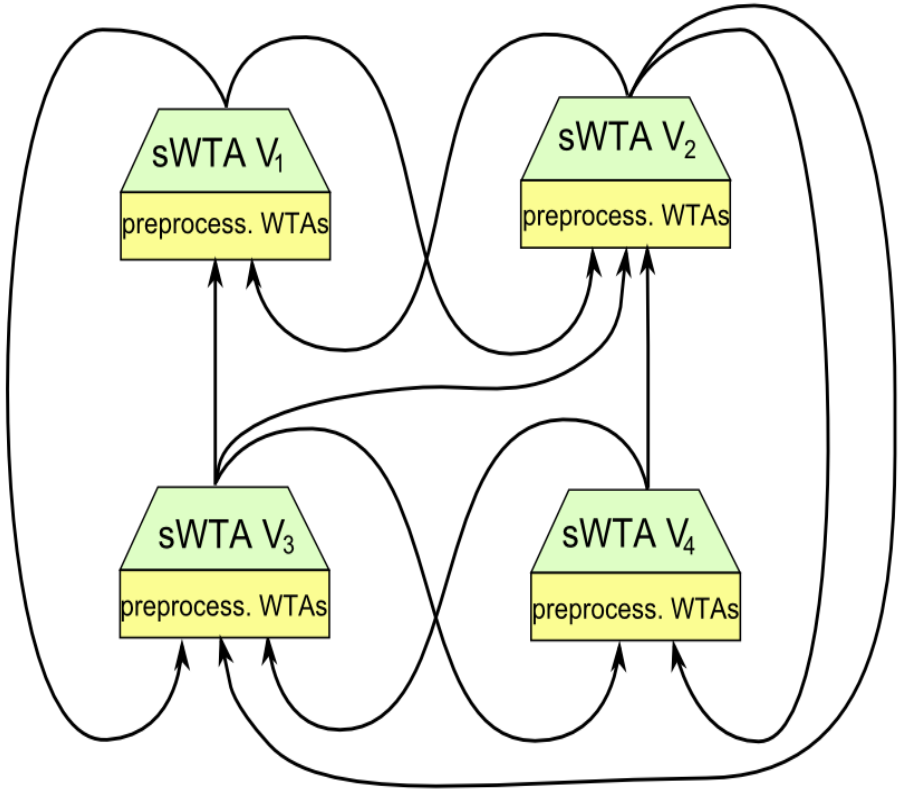
Outlook on ongoing work

[Pecevski et al, in preparation]:

Networks of very similar neural circuit moduls can implement Gibbs sampling in arbitrary Bayesian network

Example:
A simple (but non-trvial)
Bayesian network





Design of local computing moduls for implementing Gibbs sampling

- We consider a Bayesian network B (directed acyclic graph) with discrete variables $\{x_1, x_2, ..., x_m\}$

where each x_i node has a set of parents $PA(x_i)$

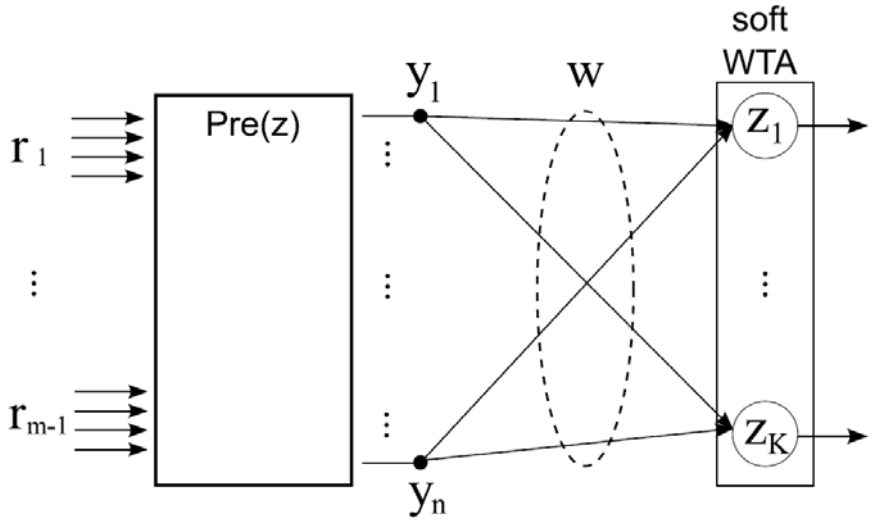
$$p(X) = \prod_i P(x_i|PA(x_i))$$

- The conditional probability for sampling z conditioned on the other variables

$$p(z|r_1, ..., r_{m-1}) = R(r_1, ..., r_{m-1}) \cdot p(z|PA(z)) \cdot \prod_{r:z\in PA(r)} p(r|z, PA^-(r))$$

where $PA^-(r) = PA(r) \setminus \{z\}$

A generic neural network module can learn to represent this conditional distribution of z



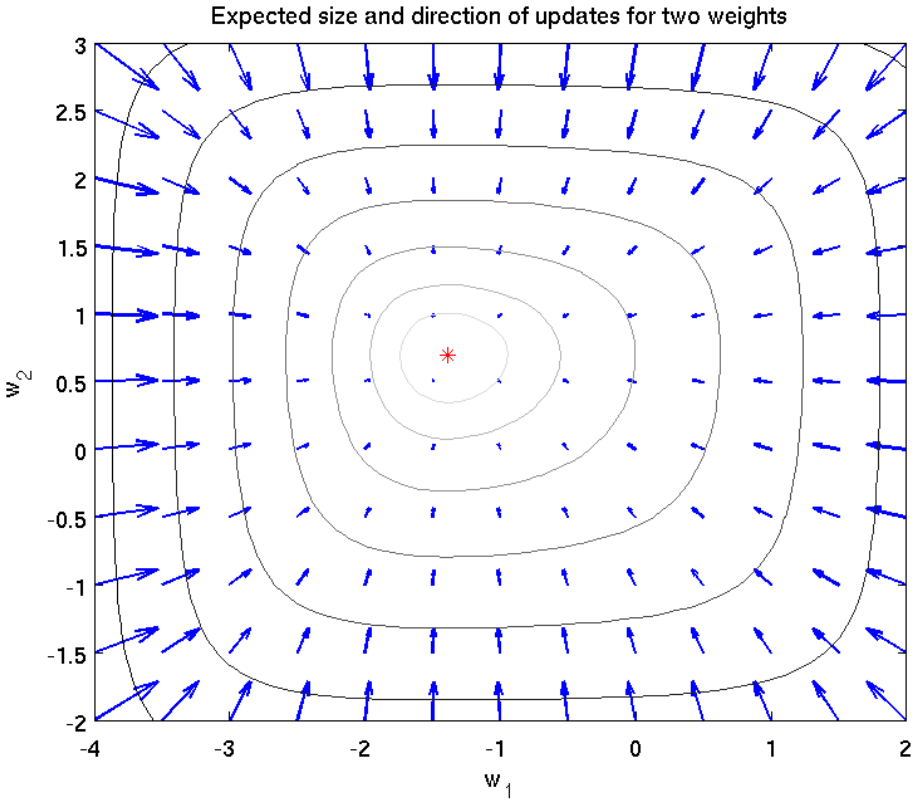
$$p(z|r_1, ..., r_{m-1})$$

$$p(z = k|\mathbf{y}) = \frac{e^{u_k}}{\sum_{l=1}^{k} e^{u_l}} \qquad u_k = \sum_{i=1}^{n} w_{ki} y_i$$

$$y_i = [PA(z) = \mathbf{c}] \qquad w_{ki} = \log p(z = k|PA(z) = \mathbf{c})$$

$$y_i = [r = a \wedge PA^-(r) = \mathbf{b}] \qquad w_{ki} = \log p(r = a|z = k \wedge PA^-(r) = \mathbf{b})$$

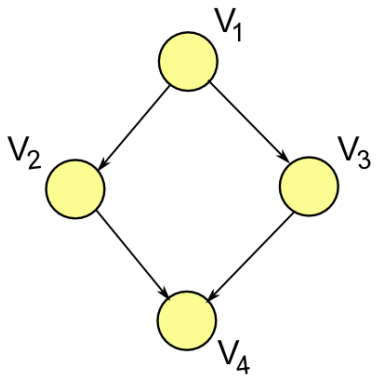
Suitable STDP learning rules (applied to complete data samples) yield exponential fast convergence to the desired weight values (with regard to expected weight changes)

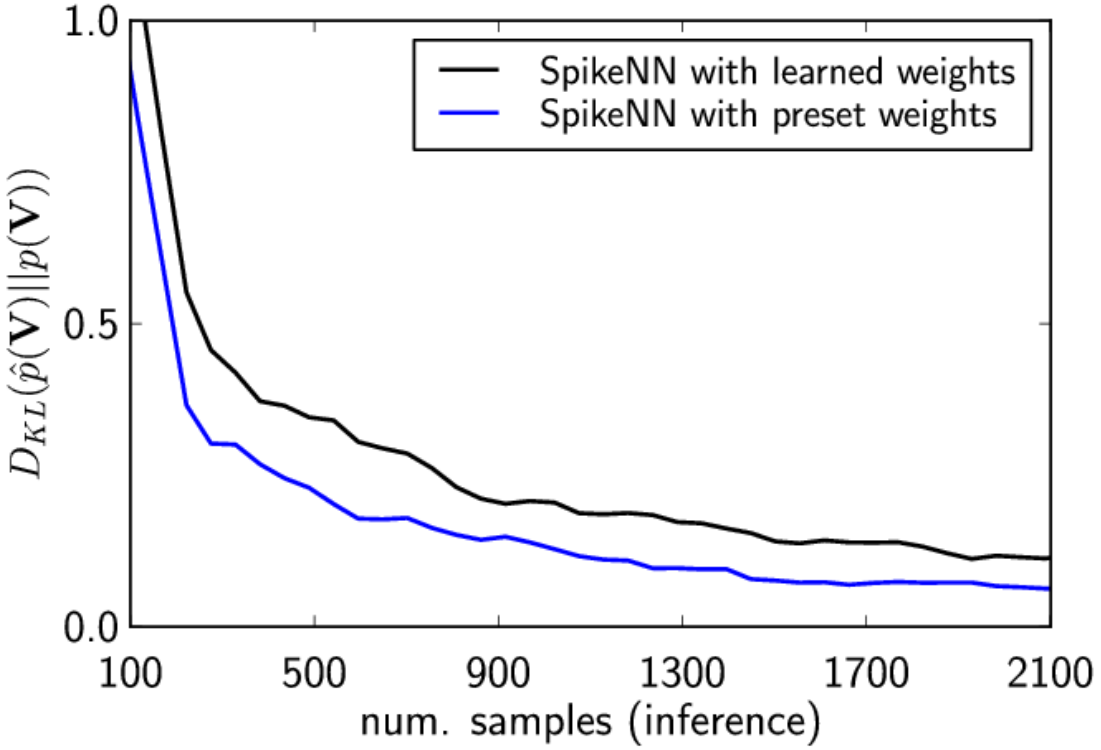


Computer test for a concrete Bayesian network:

It first learn the weights from complete samples by STDP, then generates samples from the learnt distribution

This reproduces quite well the original distribution:





Outlook

- This example suggests that we could consider brain activity as a possible implementation of Gibbs sampling (as suggested in Cognitive Science by Tenenbaum, Vul, and others)

- It opens the door to an analysis of neuronal dynamics and brain connectivity from a new perspective

- The brain may also have discovered more efficient sampling methods (for constrained distributions), that yield faster convergence to a stationary distribution

Conclusions of my talk

Generative versus discriminative learning:

- I have shown that generative models can emerge even without any explicit backwards propagation of internally generated patterns. This implicit form of generative models is less in conflict with data from neuroscience.

- These implicit generative models, that are encoded in synaptic weights (of forward connections) can provide the same theoretically predicted benefits as explicit generative models (including better generalization capability, see [Jordan and Ghahramani, 2006])

Probabilistic inference as a possible framework for understanding the organization of cortical computations:

- I have proposed a new understanding of the functional role of STDP as spike-based EM (SEM)
- This principle suggests that Bayesian computation modules are autonomously created in each WTA-circuit (that represents a posterior distribution)
- Networks of such Bayesian computation modules provide a new model for cortical computation on a probabilistic level
- This approach provides a functional explanation for the ubiquitous trial-to-trial variability of neuronal responses to stimuli (and explain it as sampling from an internally generated posterior distributions)
- The mysterious recurrent connectivity structure of cortical networks of neurons would make sense in the context of (Gibbs-) sampling
- Plasticity of the intrinsic excitability of neurons could implement the learning of priors

Probabilistic inference and learning as an inspiration for the design of a new generation of massively parallel computing devices consisting of stochastic computational units

- The „noise“ of computing elements on the molecular level could potentially become a useful resource for novel artificial computing devices (as in the stochastic WTA-circuits that I have considered)

- Gibbs sampling in networks of spiking neurons is consistent with new energy-efficient computer architectures based on event-based asynchronous parallel processing

- New spike-based hardware (for example the hardware created in the SYNAPSE project in the USA; or in the FACETS project of the EU) could be used to create devices that „think“ and autonomously generate theories for explaining their input streams

- .