Principles of Network Optimization through STDP and Rewiring

Wolfgang Maass
Institute for Theoretical Computer Science
Graz University of Technology, Austria
Motivation

- Backprop is successful because it links a functionally desirable **global** principle to **local** rules for synaptic plasticity.

- But this link gets largely lost in networks of **spiking** neurons, which are attractive because of their energy efficiency.

- **Hence new principles** are needed that provide links between desirable **global** network properties and local plasticity rules in networks of spiking neurons.

- I will sketch two principles of that type:

  1. **Expectation Maximization (EM)**

  2. The Fokker-Planck equations relates stationary distributions of network configurations to stochastic local plasticity rules.
1. Expectation maximization (EM) relates local learning through STDP to global network function for specific network architectures.
Simplest case: EM tells us how a WTA circuit can fit a mixture distribution to network inputs through STDP

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

with a linear membrane potential:

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

STDP is applied to all synapses from the input neurons ("y-neurons") to the competing neurons ("z-neurons").

EM together with a suitable version of STDP guarantees that this WTA circuit fits a mixture distribution to the distribution of inputs \(y\).

Nessler et al., PLOS Comp. Biol. 2013, Habenschuss et al., Neural Computation 2013
Implementing EM through STDP

- One starts with a „random guess“, that lets a randomly chosen $z$-neuron fire for the first spike inputs $y$

- After applying STDP to the synapses of this neuron *(M-step)*, the initial random guess is replaced for the next circuit input by the stochastic circuit response *(E-step)* resulting from the updated weights. Again, only the winner of the competition can adjust its weights via STDP to the current $y$ *(next M-step)*

- *Iterate* (just let STDP run; no separate E- or M- steps are necessary)

- The *theory of EM* [Dempster et al., 1977] *guarantees*, that these iterations do not lead to a random walk in weight-space, but rather yield *convergence* to a (local) minimum of the KL-distance between the resulting implicit generative model $p(y|w) = \frac{1}{Z} \sum_{k=1}^{K} e^{u_k(y)}$ and the external distribution $p^*(y)$ of inputs $y$

- More precisely, the circuit executes under STDP an approximation to online stochastic EM *(noise is essential* for that)

- The source of this implicit generative model is the analytic description of the equilibrium point for a synaptic weight under (idealized) STDP:
  $$\log p \ (\text{PRE has fired just before time } t \ | \ \text{POST fires at time } t) + \log c$$
  for some positive constant $c$. 
New application of this approach for learning arbitrary distributions of discrete random variables (RVs) (graphical models) by networks of spiking neurons

Assumption: Some external distribution $p^*$ generates examples $y$.

Goal: Learn an internal model of $p^*$.

The network consists of 3-layer modules, that each learn the probability table for one random variable (RV), conditioned on the RVs in its Markov blanket:

Learning takes place through STDP on synapses to hidden layer neurons, that are split into several WTA circuits.

Pecevski and Maass, Learning probabilistic inference through STDP, eNeuro, in press
Application to learning a simple Bayesian network from examples that were generated by that Bayesian network

We consider the Bayesian network from (Knill, Kersten, 1991) for explaining away in visual perception.

The 4 binary RVs of the Bayesian network require 4 WTA-like learning modules:

Seeing EM at work:
Time courses of the Kullback-Leibler divergences for the 4 learning modules:

Actual (black) and learnt (green) probability distribution.
This approach enables networks of spiking neurons to implement through STDP parameter learning and elements of \textit{structure learning} for graphical models

- If the Markov blanket of a random variable is unknown, one can still approximate its probability table through a mixture model over value assignments to all other RVs

- The number of components of the mixture model depends on the number of hidden neurons in the corresponding learning module:
2. The Fokker-Planck equations relates stationary distributions of network configurations to stochastic local plasticity rules

- Biological neural networks are subject to continuous rewiring (spine dynamics and axonal sprouting, even in the adult cortex):

![Image of network configuration over days 1 to 8 with legend for transient, semi-stable, and stable states.](image)

- Experimental data are from adult mice in the Svoboda and Rumpel Labs (Holtmaat et al., 2005), (Loewenstein et al., 2015).

- This continuous network configuration is likely to enhance network learning, but it is rarely considered in theory or neuromorphic engineering.
Contributors from our Lab

David Kappel

Robert Legenstein
A theoretical framework for network learning through rewiring and STDP

- Experimental data suggest that spine dynamics is an inherently stochastic process, that takes place even in the absence of neural activity.
- Rewiring needs to find good compromises between priors (such as sparse connectivity) and good network performance.
- We propose to use stochastic differential equations (SDEs) to formulate local plasticity rules for parameters $\theta_i$ that control synaptic connections (if $\theta_i > 0$) and synaptic weights $w_i = \exp(\theta_i - \theta_0)$:

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

where $d\mathcal{W}_i$ denotes an infinitesimal step of a random walk (Wiener process), b = learning rate, T = temperature

- The Fokker-Planck (FP) equation tracks the resulting evolution of network configurations $\theta$ over time, yielding the stationary distribution $\frac{1}{Z} p^*(\theta)^{\frac{1}{T}}$.
- Resulting new goal of network learning: sampling from a suitable distribution of network configurations ("synaptic sampling")
Practically relevant forms of the target distribution \( p^*(\theta) \) of network configurations depend on the type of learning:

**unsupervised learning** (generative model integrated with a prior)

\[
p^*(\theta|x) \propto p_S(\theta) p_N(x|\theta)
\]

where

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

**reinforcement learning** with a prior

\[
p^*(\theta) \propto p_S(\theta) \cdot p_N(R = \text{max} |\theta|)
\]

where \( R \) signals reward

This integrates policy gradient reinforcement learning with probabilistic inference.

---

Kappel, Habenschuss, Legenstein, Maass; Network plasticity as Bayesian inference, PLoS Comp Biol. 2015 and NIPS 2015

Kappel, Habenschuss, Legenstein, Maass; Reward-based network plasticity as Bayesian inference, RLDM 2015
A combined learning rule for rewriring and synaptic plasticity

Ansatz: A single parameter $\theta_i$ controls the spine volume and --once a synaptic connection has been formed-- the weight of this synaptic connection $w_i = \exp(\theta_i - \theta_0)$:

$$d\theta_i = b \left( \frac{\partial}{\partial \theta_i} \log p_S(\theta) + N \exp(\theta_i - \theta_0) \frac{\partial}{\partial w_i} \log p_N(x^n|w) \right) dt + \sqrt{2Tb} dW_i$$

We use a Gaussian as prior $p_S(\theta)$

I will focus on the generative model of (Nessler et al., 2013), where

$$\frac{\partial}{\partial w_i} \log p_N(x^n|w) \approx S(t)(x_i(t) - \alpha e^{w_i})$$

approximates STDP.

S(t) is the postsynaptic spike train, $x_i(t)$ denotes the trace of EPSPs from presynaptic neuron $i$.

Features of this learning rule:

• it reproduces experimental data on stochastic changes of spine volumes even in the absence of neural activity (Ornstein-Uhlenback process)
• it reproduces experimental data on multiplicative updates of spine volumes and synaptic weights
• It reproduces power-law survival curves for synaptic connections
• it approximates STDP for $\theta_i > 0$
This learning approach 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 potential connections between inputs and hidden neurons, and among hidden neurons are subject to the combined learning rule (rewiring and STDP)

The performance of a network configuration can be measured by testing how well its predicts the visual input if only auditory input is presented (or vice versa).
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)

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

First 3 principal components of a subset of the parameters $\theta$:
Maximum Likelihood learning vs. learning a posterior

- Maximum Likelihood (ML) learning for generative models: Find parameters that maximize the likelihood of the actually occurring stimuli $x$:
  $$\theta^* = \arg \max_{\theta} p_N(x|\theta)$$

- A Bayesian approach suggests to learn instead a posterior
  $$p^*(\theta|x) \propto p_S(\theta) p_N(x|\theta)$$
  that integrates a prior $p_S(\theta)$

But how can such a posterior be represented by the network, and learnt?
We propose: The posterior is represented implicitly as a distribution of network configurations, from which the network samples.

The Fokker-Planck equation provides a tool for deriving a from target posterior:

\[ p^*(\theta|x) = p_S(\theta) p_N(x|\theta) / Z \]

on the global level. Local plasticity rules that produce (or rather approximate) this given posterior:

\[ d\theta_i = \left( b \frac{\partial}{\partial \theta_i} \log p_S(\theta) + b \frac{\partial}{\partial \theta_i} \log p_N(x|\theta) \right) dt + \sqrt{2Tb} d\mathcal{W}_i \]
Application to reinforcement learning can not only realize Bayesian reinforcement learning, but also provides a principled away of escaping local optima and saddle points assuming a prior $p_S(\theta)$

unsupervised learning (generative model integrated with a prior)

$$p^*(\theta|x) \propto p_S(\theta) p_N(x|\theta)$$

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

reinforcement learning with a prior

$$p^*(\theta) \propto p_S(\theta) \cdot p_N(R = 1|\theta)$$

where $R$ signals reward

This integrates policy gradient reinforcement learning with probabilistic inference.

Kappel, Habenschuss, Legenstein, Maass; Network plasticity as Bayesian inference, PLoS Comp Biol. 2015 and NIPS 2015

Kappel, Habenschuss, Legenstein, Maass; Reward-based network plasticity as Bayesian inference, RLDM 2015
From network learning to network optimization:
Temperature regulation engages simulated annealing

The local plasticity rule \( d\theta_i = \left( b \frac{\partial}{\partial \theta_i} \log p^*(\theta) \right) dt + \sqrt{2Tb} \cdot d\mathcal{W}_i \) yields for general values of \( T \) (and a flat prior) the distribution \( p_N(R = 1 | \theta) \frac{1}{T} \).

For binary rewards \( R = 0, 1 \) the resulting expected reward reaches the global optimum when the temperature \( T \) decreases:

\[
E[R] = \frac{1}{Z} \int p_N(R = 1 | \theta) \ p_N(R = 1 | \theta)^{\frac{1}{T}} \ d\theta
\]

Hence synaptic sampling in conjunction with cooling can in principle find **globally optimal solutions** for maximizing rewards (exactly like in simulated annealing).

Fast burn-in at any temperature \( T \) is needed for that. Work on methods for that are in progress.
**Summary:** Networks can learn a lot without a supervisor

- EM helps us to understand how a suitably modular network of spiking neurons can learn *any* distribution over discrete RVs through STDP

- The same network of spiking neurons can immediately extract knowledge from this learnt distribution through *probabilistic inference*

- It is well known in machine learning that *learning of a posterior* is more powerful than Maximum Likelihood learning

- If one represents the posterior as distribution of network configurations, one gets a *principled way for deriving local plasticity rules*

- Also a *link* between network learning (through local learning rules) *general nonlinear optimization* (simulated annealing) can be created in this way

- The 2 principles that I have discussed *provide gold standards* for the design of neural circuits and local plasticity rules in neuromorphic devices