

Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity

Bernhard Nessler^{1*}, Michael Pfeiffer^{1,2}, Lars Buesing¹, Wolfgang Maass¹

¹ Institute for Theoretical Computer Science, Graz University of Technology, Graz, Austria, ² Institute of Neuroinformatics, University of Zürich and ETH Zürich, Zürich, Switzerland

Abstract

The principles by which networks of neurons compute, and how spike-timing dependent plasticity (STDP) of synaptic weights generates and maintains their computational function, are unknown. Preceding work has shown that soft winner-take-all (WTA) circuits, where pyramidal neurons inhibit each other via interneurons, are a common motif of cortical microcircuits. We show through theoretical analysis and computer simulations that Bayesian computation is induced in these network motifs through STDP in combination with activity-dependent changes in the excitability of neurons. The fundamental components of this emergent Bayesian computation are priors that result from adaptation of neuronal excitability and implicit generative models for hidden causes that are created in the synaptic weights through STDP. In fact, a surprising result is that STDP is able to approximate a powerful principle for fitting such implicit generative models to high-dimensional spike inputs: Expectation Maximization. Our results suggest that the experimentally observed spontaneous activity and trial-to-trial variability of cortical neurons are essential features of their information processing capability, since their functional role is to represent probability distributions rather than static neural codes. Furthermore it suggests networks of Bayesian computation modules as a new model for distributed information processing in the cortex.

Citation: Nessler B, Pfeiffer M, Buesing L, Maass W (2013) Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity. PLoS Comput Biol 9(4): e1003037. doi:10.1371/journal.pcbi.1003037

Editor: Olaf Sporns, Indiana University, United States of America

Received: May 19, 2012; **Accepted:** March 4, 2013; **Published:** April 25, 2013

Copyright: © 2013 Nessler et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: This work was written under partial support by project #FP7-216593 (SECO), project #FP7-506778 (PASCAL2), project #FP7-243914 (BRAIN-I-NETS) and project #FP7-269921 (BrainScaleS) of the European Union. MP has been supported by the Samsung Advanced Institute of Technology and a Forschungskredit grant of the University of Zurich. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: nessler@igi.tugraz.at

Introduction

Numerous experimental data show that the brain applies principles of Bayesian inference for analyzing sensory stimuli, for reasoning and for producing adequate motor outputs [1–5]. Bayesian inference has been suggested as a mechanism for the important task of probabilistic perception [6], in which hidden causes (e.g. the categories of objects) that explain noisy and potentially ambiguous sensory inputs have to be inferred. This process requires the combination of prior beliefs about the availability of causes in the environment, and probabilistic generative models of likely sensory observations that result from any given cause. By Bayes Theorem, the result of the inference process yields a *posterior* probability distribution over hidden causes that is computed by multiplying the *prior* probability with the *likelihood* of the sensory evidence for all possible causes. In this article we refer to the computation of posterior probabilities through a combination of probabilistic prior and likelihood models as Bayesian computation. It has previously been shown that priors and models that encode likelihoods of external stimuli for a given cause can be represented in the parameters of neural network models [6,7]. However, in spite of the existing evidence that Bayesian computation is a primary information processing step in the brain, it has remained open how networks of neurons can acquire these priors and likelihood models, and how they combine them to arrive at posterior distributions of hidden causes.

The fundamental computational units of the brain, neurons and synapses, are well characterized. The synaptic connections are subject to various forms of plasticity, and recent experimental results have emphasized the role of STDP, which constantly modifies synaptic strengths (weights) in dependence of the difference between the firing times of the pre- and postsynaptic neurons (see [8,9] for reviews). Functional consequences of STDP can resemble those of rate-based Hebbian models [10], but may also lead to the emergence of temporal coding [11] and rate-normalization [12,13]. In addition, the excitability of neurons is modified through their firing activity [14]. Some hints about the organization of local computations in stereotypical columns or so-called cortical microcircuits [15] arises from data about the anatomical structure of these hypothesized basis computational modules of the brain. In particular, it has been observed that local ensembles of pyramidal neurons on layers 2/3 and layers 5/6 typically inhibit each other, via indirect synaptic connections involving inhibitory neurons [16]. These ubiquitous network motifs were called soft winner-take-all (WTA) circuits, and have been suggested as neural network models for implementing functions like non-linear selection [16,17], normalization [18], selective attention [19], decision making [20,21], or as primitives for general purpose computation [22,23].

A comprehensive theory that explains the emergence of computational function in WTA networks of spiking neurons through STDP has so far been lacking. We show in this article that

Author Summary

How do neurons learn to extract information from their inputs, and perform meaningful computations? Neurons receive inputs as continuous streams of action potentials or “spikes” that arrive at thousands of synapses. The strength of these synapses - the synaptic weight - undergoes constant modification. It has been demonstrated in numerous experiments that this modification depends on the temporal order of spikes in the pre- and postsynaptic neuron, a rule known as STDP, but it has remained unclear, how this contributes to higher level functions in neural network architectures. In this paper we show that STDP induces in a commonly found connectivity motif in the cortex - a winner-take-all (WTA) network - autonomous, self-organized learning of probabilistic models of the input. The resulting function of the neural circuit is Bayesian computation on the input spike trains. Such unsupervised learning has previously been studied extensively on an abstract, algorithmical level. We show that STDP approximates one of the most powerful learning methods in machine learning, Expectation-Maximization (EM). In a series of computer simulations we demonstrate that this enables STDP in WTA circuits to solve complex learning tasks, reaching a performance level that surpasses previous uses of spiking neural networks.

STDP and adaptations of neural excitability are likely to provide the fundamental components of Bayesian computation in soft WTA circuits, yielding representations of posterior distributions for hidden causes of high-dimensional spike inputs through the firing probabilities of pyramidal neurons. This is shown in detail for a simple, but very relevant feed-forward model of Bayesian inference, in which the distribution for a single hidden cause is inferred from the afferent spike trains. Our new theory thus describes how modules of soft WTA circuits can acquire and perform Bayesian computations to solve one of the fundamental tasks in perception, namely approximately inferring the category of an object from feed-forward input. Neural network models that can handle Bayesian inference in general graphical models, including bi-directional inference over arbitrary sets of random variables, explaining away effects, different statistical dependency models, or inference over time require more complex network architectures [24,25], and are the topic of ongoing research. Such networks can be composed out of interconnected soft WTA circuits, which has been shown to be a powerful principle for designing neural networks that can solve arbitrary deterministic or stochastic computations [22,23,25]. Our theory can thus be seen as a first step towards learning the desired functionality of individual modules.

At the heart of this link between Bayesian computation and network motifs of cortical microcircuits lies a new theoretical insight on the micro-scale: If the STDP-induced changes in synaptic strength depend in a particular way on the current synaptic strength, STDP approximates for each synapse exponentially fast the conditional probability that the presynaptic neuron has fired just before the postsynaptic neuron (given that the postsynaptic neuron fires). This principle suggests that synaptic weights can be understood as conditional probabilities, and the ensemble of all weights of a neuron as a generative model for high-dimensional inputs that - after learning - causes it to fire with a probability that depends on how well its current input agrees with this generative model. The concept of a generative model is well known in theoretical neuroscience [26,27], but it has so far primarily been applied in the context of an abstract non-spiking

neural circuit architecture. In the Bayesian computations that we consider in this article, internal generative models are represented implicitly through the learned values of bottom-up weights in spiking soft-WTA circuits, and inference is carried out by neurons that integrate such synaptic inputs and compete for firing in a WTA circuit. In contrast to previous rate-based models for probabilistic inference [28–30] every spike in our model has a clear semantic interpretation: one spike indicates the instantaneous assignment of a certain value to an abstract variable represented by the firing neuron. In a Bayesian inference context, every input spike provides evidence for an observed variable, whereas every output spike represents one stochastic sample from the posterior distribution over hidden causes encoded in the circuit.

We show that STDP is able to approximate the arguably most powerful known learning principle for creating these implicit generative models in the synaptic weights: Expectation Maximization (EM). The fact that STDP approximates EM is remarkable, since it is known from machine learning that EM can solve a fundamental chicken-and-egg problem of unsupervised learning systems [31]: To detect - without a teacher - hidden causes for complex input data, and to induce separate learning agents to specialize each on one of the hidden causes. The problem is that as long as the hidden causes are unknown to the learning system, it cannot tell the hidden units what to specialize on. EM is an iterative process, where initial guesses of hidden causes are applied to the current input (E-step) and successively improved (M-step), until a local maximum in the log-likelihood of the input data is reached. In fact, the basic idea of EM is so widely applicable and powerful that most state-of-the-art machine learning approaches for discovering salient patterns or structures in real-world data without a human supervisor rely on some form of EM [32]. We show that in our spiking soft-WTA circuit each output spike can be viewed as an application of the E-step of EM. The subsequent modification of the synaptic weights between the presynaptic input neurons and the very neuron that has fired the postsynaptic spike according to STDP can be viewed as a move in the direction of the M-step of a stochastic online EM procedure. This procedure strives to create optimal internal models for high-dimensional spike inputs by maximizing their *log*-likelihood. We refer to this interpretation of the functional role of STDP in the context of spiking WTA circuits as **spike-based Expectation Maximization (SEM)**.

This analysis gives rise to a new perspective of the computational role of local WTA circuits as parts of cortical microcircuits, and the role of STDP in such circuits: The fundamental computational operations of Bayesian computation (Bayes Theorem) for the inference of hidden causes from bottom-up input emerge in these local circuits through plasticity. The pyramidal neurons in the WTA circuit encode in their spikes samples from a posterior distribution over hidden causes for high-dimensional spike inputs. Inhibition in the WTA accounts for normalization [18], and in addition controls the rate at which samples are generated. The necessary multiplication of likelihoods (given by implicit generative models that are learned and encoded in their synaptic weights) with simultaneously learned priors for hidden causes (in our model encoded in the neuronal excitability), does not require any extra computational machinery. Instead, it is automatically carried out (on the *log* scale) through linear features of standard neuron models. We demonstrate the emergent computational capability of these self-organizing modules for Bayesian computation through computer simulations. In fact, it turns out that a resulting configuration of networks of spiking neurons can solve demanding computational tasks, such as the discovery of prototypes for handwritten digits without any

supervision. We also show that these emergent Bayesian computation modules are able to discover, and communicate through a sparse output spike code, repeating spatio-temporal patterns of input spikes. Since such self-adaptive computing and discrimination capability on high-dimensional spatio-temporal spike patterns is not only essential for early sensory processing, but could represent a generic information processing step also in higher cortical areas, our analysis suggests to consider networks of self-organizing modules for spike-based Bayesian computation as a new model for distributed real-time information processing in the brain.

Preliminary ideas for a spike-based implementation of EM were already presented in the extended abstract [20], where we analyzed the relationship of a simple STDP rule to a Hebbian learning rule, and sketched a proof for stochastic online EM. In the present work we provide a rigorous mathematical analysis of the learning procedure, a proof of convergence, expand the framework towards learning spatio-temporal spike patterns, and discuss in detail the relationship of our STDP rule to experimental results, as well as the interpretation of spikes as samples from instantaneous posterior probability distributions in the context of EM.

Results

In this section we define a simple model circuit and show that every spiking event of the circuit can be described as one independent sample of a discrete probability distribution, which itself evolves over time in response to the spiking input. Within this network we analyze a variant of a STDP rule, in which the strength of potentiation depends on the current weight value. This local learning rule, which is supported by experimental data, and at intermediate spike frequencies closely resembles typical STDP rules from the literature, drives every synaptic weight to converge stochastically to the log of the probability that the presynaptic input neuron fired a spike within a short time window $[t' - \sigma, t']$, before the postsynaptic neuron spikes at time t' :

$$w \rightarrow \log p(\text{presynaptic neuron fired within } [t' - \sigma, t'] \mid \text{postsynaptic neuron fires at } t'). \quad (1)$$

We then show that the network model can be viewed as performing Bayesian computation, meaning that every spike can be understood as a sample from a posterior distribution over hidden causes in a generative probabilistic model, which combines prior probabilities and evidence from current input spike trains.

This understanding of spikes as samples of hidden causes leads to the central result of this paper. We show that STDP implements a stochastic version of Expectation Maximization for the unsupervised learning of the generative model and present convergence results for SEM. Importantly, this implementation of EM is based on spike events, rather than spike rates.

Finally we discuss how our model can be implemented with biologically realistic mechanisms. In particular this provides a link between mechanisms for lateral inhibition in WTA circuits and learning of probabilistic models. We finally demonstrate in several computer experiments that SEM can solve very demanding tasks, such as detecting and learning repeatedly occurring spike patterns, and learning models for images of handwritten digits without any supervision.

Definition of the network model

Our model consists of a network of spiking neurons, arranged in a WTA circuit, which is one of the most frequently studied

connectivity patterns (or network motifs) of cortical microcircuits [16]. The input of the circuit is represented by the excitatory neurons y_1, \dots, y_n . This input projects to a population of excitatory neurons z_1, \dots, z_K that are arranged in a WTA circuit (see Fig. 1). We model the effect of lateral inhibition, which is the competition mechanism of a WTA circuit [33], by a common inhibitory signal $I(t)$ that is fed to all z neurons and in turn depends on the activity of the z neurons. Evidence for such common local inhibitory signals for nearby neurons arises from numerous experimental results, see e.g. [16,34–36]. We do not a priori impose a specific functional relationship between the common inhibition signal and the excitatory activity. Instead we will later derive necessary conditions for this relationship, and propose a mechanism that we use for the experiments.

The individual units z_k are modeled by a simplified Spike Response Model [37] in which the membrane potential is computed as the difference between the excitatory input $u_k(t)$ and the common inhibition term $I(t)$. $u_k(t)$ sums up the excitatory inputs from neurons y_1, \dots, y_n as

$$u_k(t) = w_{k0} + \sum_{i=1}^n w_{ki} \cdot y_i(t). \quad (2)$$

$w_{ki} \cdot y_i(t)$ models the EPSPs evoked by spikes of the presynaptic neuron y_i , and w_{k0} models the intrinsic excitability of the neuron z_k . In order to simplify our analysis we assume that the EPSP can be modeled as a step function with amplitude w_{ki} , i.e., $y_i(t)$ it takes on the value 1 in a finite time window of length σ after a spike and is zero before and afterwards. Further spikes within this time window do not contribute additively to the EPSP, but only extend the time window during which the EPSP is in the high state. We will later show how to extend our results to the case of realistically shaped and additive EPSPs.

We use a stochastic firing model for z_k , in which the firing probability depends exponentially on the membrane potential, i.e.,

$$p(z_k \text{ fires at time } t) \propto \exp(u_k(t) - I(t)), \quad (3)$$

which is in good agreement with most experimental data [38]. We can thus model the firing behavior of every neuron z_k in the WTA as an independent inhomogeneous Poisson process whose instantaneous firing rate is given by $r_k(t) = \exp(u_k(t) - I(t))$.

In order to understand how this network model generates samples from a probability distribution, we first observe that the combined firing activity of the neurons z_1, \dots, z_K in the WTA circuit is simply the sum of the K independent Poisson processes, and can thus again be modeled as an inhomogeneous Poisson process with rate $R(t) = \sum_{k=1}^K r_k(t)$. Furthermore, in any infinitesimally small time interval $[t, t + \delta t]$, the neuron z_k spikes with probability $r_k(t) \delta t$. Thus, if we know that at some point in time t , i.e. within $[t, t + \delta t]$, one of the neurons z_1, \dots, z_K produces an output spike, the conditional probability $q_k(t)$ that this spike originated from neuron z_k can be expressed as

$$q_k(t) = \frac{r_k(t) \delta t}{R(t) \delta t} = \frac{e^{u_k(t)}}{\sum_{k'=1}^K e^{u_{k'}(t)}}. \quad (4)$$

Every single spike from the WTA circuit can thus be seen as an independent sample from the instantaneous distribution in Eq. (4) at the time of the spike. Although the instantaneous firing rate of every neuron directly depends on the value of the inhibition $I(t)$, the relative proportion of the rate $r_k(t)$ to the total WTA firing rate

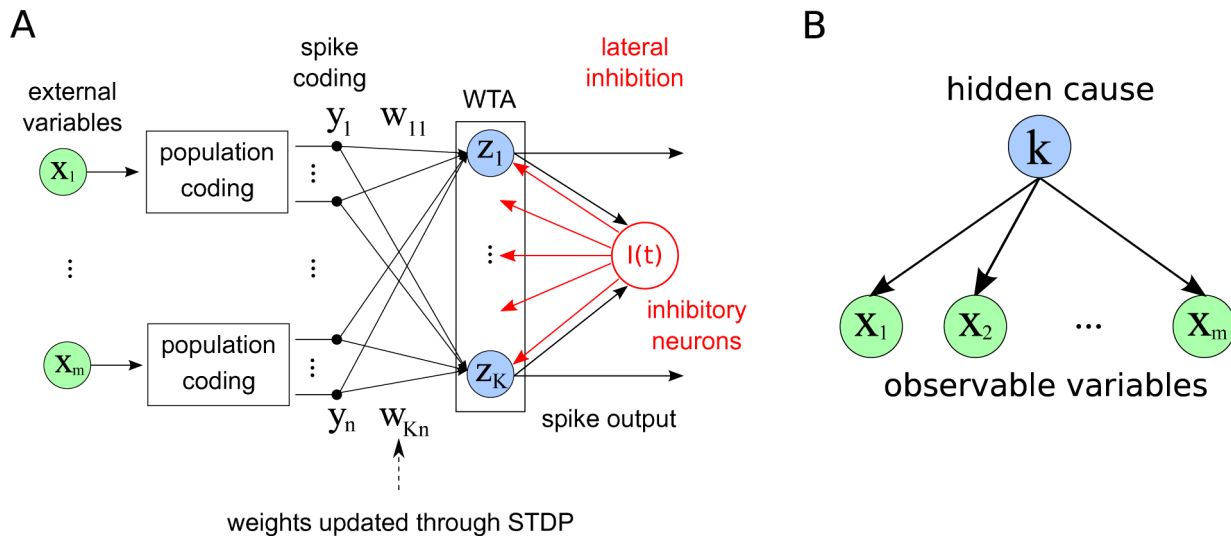


Figure 1. The network model and its probabilistic interpretation. **A** Circuit architecture. External input variables are encoded by populations of spiking neurons, which feed into a Winner-take-all (WTA) circuit. Neurons within the WTA circuit compete via lateral inhibition and have their input weights updated through STDP. Spikes from the WTA circuit constitute the output of the system. **B** Generative probabilistic model for a multinomial mixture: A vector of external input variables x_1, \dots, x_m is dependent on a hidden cause, which is represented by the discrete random variable k . In this model it is assumed that the x_i 's are conditionally independent of each other, given k . The inference task is to infer the value of k , given the observations for x_i . Our neuronal network model encodes the conditional probabilities of the graphical model into the weight vector w , such that the activity of the network can be understood as execution of this inference task. doi:10.1371/journal.pcbi.1003037.g001

$R(t)$ is independent of the inhibition, because all neurons receive the same inhibition signal $I(t)$. Note that $q_k(t)$ determines only the value of the sample at time t , but not the time point at which a sample is created. The temporal structure of the sampling process depends only on the overall firing rate $R(t)$.

This implementation of a stochastic WTA circuit does not constrain in any way the kind of spike patterns that can be produced. Every neuron fires independently according to a Poisson process, so it is perfectly possible (and sometimes desirable) that there are two or more neurons that fire (quasi) simultaneously. This is no contradiction to the above theoretical argument of single spikes as samples. There we assumed that there was only one spike at a time inside a time window, but since we assumed these windows to be infinitesimally small, the probability of two spikes occurring exactly at the same point in continuous time is zero.

Synaptic and intrinsic plasticity. We can now establish a link between biologically plausible forms of spike-based learning in the above network model and learning via EM in probabilistic graphical models. The synaptic weights w_{ki} of excitatory connections between input neurons y_i and neurons z_k in the WTA circuit change due to STDP. Many different versions of STDP rules have emerged from experimental data [8,39,40]. For synaptic connections between excitatory neurons, most of them yield a long term potentiation (LTP) when the presynaptic neuron y_i fires before the postsynaptic neuron z_k , otherwise a long term depression (LTD). In our model we use a STDP rule in which the shape of the positive update follows the shape of EPSPs at the synapses, and in which the amplitude of the update Δw_{ki} depends on the value of the synaptic weight w_{ki} before the update as in Fig. 2. Specifically, we propose a rule in which the ratio of LTP and LTD amplitudes is inversely exponentially dependent on the current synaptic weight. LTP curves that mirror the EPSP shape are in accordance with previous studies, which analyzed optimal shapes of STDP curves under different mathematical criteria [41,42]. The depression part of the rule in Fig. 2 is a flat offset that contrasts the potentiation. We will show later that this form of

LTD occurs in our simulations only at very low repetition frequencies, and instead at natural frequencies our model gives rise to a form of STDP with spike-timing dependent LTD that is very similar to plasticity curves observed in biology [40,43]. We will also analyze the relationship between this rule and a biologically more realistic STDP rule with an explicit time-decaying LTD part.

We can formulate this STDP-rule as a Hebbian learning rule $w_{ki} \leftarrow w_{ki} + \eta \Delta w_{ki}$ - with learning rate η - which is triggered by a spike of the postsynaptic neuron z_k at time t^f . The dependence of Δw_{ki} on the synaptic activity $y_i(t)$ and the current value of the synaptic weight is given by

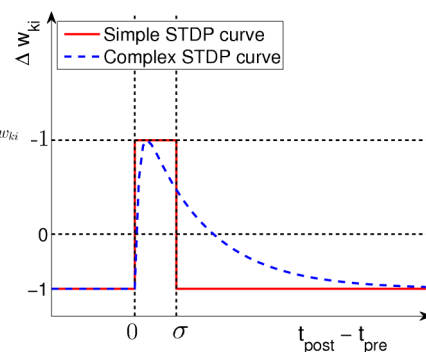


Figure 2. Learning curves for STDP. Under the simple STDP model (red curve), potentiation occurs only if the postsynaptic spike falls within a time window of length σ (typically 10ms) after the presynaptic spike. The convergence properties of this simpler version in conjunction with rectangular non-additive EPSPs are easier to analyze. In our simulations we use the more complex version (blue dashed curve) in combination with EPSPs that are modeled as biologically realistic α -kernels (with plausible time-constants for rise and decay of 1 respectively 15 ms). doi:10.1371/journal.pcbi.1003037.g002

$$\Delta w_{ki} = \begin{cases} ce^{-w_{ki}} - 1, & \text{if } y_i(t^f) = 1, \text{ i.e. } y_i \text{ fired in } [t^f - \sigma, t^f] \\ -1, & \text{if } y_i(t^f) = 0, \text{ i.e. } y_i \text{ did not fire in } [t^f - \sigma, t^f] \end{cases} \quad (5)$$

Since $y_i(t)$ reflects the previously defined step function shape of the EPSP, this update rule is exactly equivalent to the simple STDP rule (solid red curve) in Fig. 2 for the case of the pairing of one pre- and one postsynaptic spike. The dependence on the presynaptic activity y_i is reflected directly by the time difference $t_{\text{post}} - t_{\text{pre}}$ between the pre- and the postsynaptic spikes. According to this rule positive updates are only performed if the presynaptic neuron fired in a time window of σ ms before the postsynaptic spike. This learning rule therefore respects the causality principle of LTP that is implied in Hebb's original formulation [44], rather than looking only at correlations of firing rates.

We can interpret the learning behavior of this simple STDP rule from a probabilistic perspective. Defining a stationary joint distribution $p^*(\mathbf{y}, \mathbf{z})$ over the binary input activations \mathbf{y} at the times of the postsynaptic spikes, and the binary vector \mathbf{z} , which indicates the source of the postsynaptic spike by setting one $z_k = 1$, we show in Methods that the equilibrium condition of the expected update $E[\Delta w_{ki}]$ leads to the single solution

$$E[\Delta w_{ki}] = 0 \Leftrightarrow w_{ki} = \log p^*(y_i = 1 | z_k = 1) + \log c. \quad (6)$$

This stochastic convergence to the log-probability of the presynaptic neuron being active right before the postsynaptic neuron fires is due to the exponential dependence of the potentiation term on the current weight value. Log-probabilities are necessarily negative values, whereas for biological neural networks we typically expect excitatory, i.e. positive weights from the excitatory input neurons. The parameter c shifts the range of the values for the weights w_{ki} into the positive regime for $c > 1$. For the sake of simplicity we assume that $c = 1$ for the following theoretical analysis and we show in Methods that all results remain true for any positive value of c .

In analogy to the plasticity of the synaptic weights we also explore a form of intrinsic plasticity of the neurons. We interpret w_{k0} as an indicator for the excitability of the neuron z_k and apply a circuit-spike triggered update rule $w_{k0} \leftarrow w_{k0} + \eta \Delta w_{k0}$ with

$$\Delta w_{k0} = e^{-w_{k0}} z_k - 1. \quad (7)$$

Whenever a neuron z_k fires, the excitability is increased and the amount of increase is inversely exponentially dependent on the current excitability. Otherwise the excitability is decreased by a constant. Such positive feedback through use-dependent changes in the excitability of neurons were found in numerous experimental studies (see e.g. [14,45]). This concrete model of intrinsic plasticity drives the excitability w_{k0} towards the only equilibrium point of the update rule, which is $\log p^*(z_k = 1)$. In Methods (see 'Weight offsets and positive weights') we show that the depression of the excitability can be modeled either as an effect of lateral inhibition from firing of neighboring neurons, or as a constant decay, independent of the instantaneous circuit activity. Both methods lead to different values w_{k0} , it is true, but encode identical instantaneous distributions $q_k(t)$.

Note, however, that also negative feedback effects on the excitability through homeostatic mechanisms were observed in experiments [13,46]. In a forthcoming article [47] we show that the use of such homeostatic mechanisms instead of Eq. (7) in an, otherwise unchanged, network model may be interpreted as a posterior constraint in the context of EM.

Generative probabilistic model. The instantaneous spike distribution $q_k(t)$ from Eq. (4) can be understood as the result of Bayesian inference in an underlying generative probabilistic model for the abstract multinomial observed variables x_1, \dots, x_m and a hidden cause k . We define the probability distribution of the variables k and \mathbf{x} , as shown by the graphical model in Fig. 1B, as $p(k, \mathbf{x} | \theta) = p(k | \theta) \cdot \prod_{j=1}^m p(x_j | k, \theta)$. The parametrization θ of the graphical model consists of a prior $p(k | \theta)$ on k , and conditional probabilities $p(x_j | k, \theta)$ for every x_j .

The probabilistic model $p(k, \mathbf{x} | \theta)$ is a generative model and therefore serves two purposes: On the one hand, it can be used to generate samples of the hidden variable k and the observable variables x_1, \dots, x_m . This is done by sampling k from the prior distribution, and then sampling the x_j 's, which depend on k and can be generated according to the conditional probability tables. The resulting marginal distribution $p(\mathbf{x} | \theta)$ is a special case of a multinomial mixture distribution.

On the other hand, for any given observation of the vector \mathbf{x} , one can infer the value of the hidden cause k that led to the generation of this value for \mathbf{x} . By application of Bayes' rule one can infer the posterior distribution $p(k | \mathbf{x}, \theta)$ over all possible values of k , which is proportional to the product of the prior $p(k | \theta)$ and the likelihood $p(\mathbf{x} | k, \theta)$.

We define population codes to represent the external observable variables x_1, \dots, x_m by the input neurons y_1, \dots, y_n , and the hidden variable k by the circuit neurons z_1, \dots, z_K : For every variable x_j and every possible (discrete) value that x_j can adopt, there is exactly one neuron y_i which represents this combination. We call G_j the set of the indices of all y_i 's that represent x_j , and we call $v(i)$ the possible value of x_j that is represented by neuron y_i . Thus we can define an interpretation for the spikes from the input neurons by

$$\text{neuron } y_i \text{ fires at } t^f \Rightarrow x_j(t^f) = v(i), \quad \text{for } i \in G_j. \quad (8)$$

A spike from the group G_j represents an instantaneous evidence about the observable variable x_j at the time of the spike. In the same way every neuron z_1, \dots, z_K represents one of the K possible values for the hidden variable k , and every single spike conveys an instantaneous value for k . We can safely assume that all neurons - including the input neurons - fire according to their individual local stochastic processes or at least exhibit some local stochastic jitter. For the theoretical analysis one can regard a spike as an instantaneous event at a single point in time. Thus in a continuous time no two events from such local stochastic processes can happen at exactly the same point in time. Thus, there is never more than one spike at any single point in time within a group G_j , and every spike can be treated as a proper sample from x_j . However, the neurons z_k coding for hidden causes need to integrate evidence from multiple inputs, and thus need a mechanism to retain the instantaneous evidence from a single spike over time, in order to learn from spatial and temporal correlations in the input.

In our framework this is modeled by postsynaptic potentials on the side of the receiving neurons that are generated in response to input spikes, and, by their shape, represent evidence over time. In the simple case of the non-additive step-function model of the EPSP in Eq. (2), every spike indicates new evidence for the encoded variable that remains valid during a time window of σ , after which the evidence is cleared. In the case that there is no spike from one group G_j within a time window of length σ , this is interpreted as missing evidence (or missing value) for x_j in a subsequent inference. In practice it may also occur that EPSPs

within a group G_j of input neurons overlap, which would indicate contradicting evidence for x_j . For the theoretical analysis we will first assume that spikes from different input neurons within the same group G_j are not closer in time than σ , in order to avoid such conflicts. We will later drop this restriction in the extension to more realistically shaped additive EPSPs by slightly enhancing the probabilistic model.

In our experiments with static input patterns we typically use the following basis scheme to encode the external input variables $x_j(t)$ by populations of stochastic spiking neurons y_i : at every point in time t there is exactly one neuron y_i in every group G_j that represents the instantaneous value of $x_j(t)$. We call this neuron the active neuron of the group, whereas all other neurons of the group are inactive. During the time where a neuron y_i is active it fires stochastically according to a Poisson processes with a certain constant or oscillating rate. The inactive neurons, however, remain silent, i.e. they fire with a rate near 0. Although not explicitly modeled here, such an effect can result from strong lateral inhibition in the input populations. This scheme certainly fulfills the definition in Eq. (8).

Here and in the following we will write $\mathbf{y}(t)$ to denote the input activation through the EPSPs of the network model, and \mathbf{y} to denote a variable in the probabilistic model, which models the distribution of $\mathbf{y}(t)$ over all time points t . We will also use notations like $p(\mathbf{z}|\mathbf{y}(t), \mathbf{w})$, which refers to the variable \mathbf{y} in the probabilistic model taking on the value $\mathbf{y}(t)$. We can then reformulate the abstract probabilistic model $p(\mathbf{x}, k|\theta)$ using the above population codes that define the binary variable vectors \mathbf{y} and \mathbf{z} , with k s.t. $z_k = 1$ as:

$$p(\mathbf{z}, \mathbf{y}|\mathbf{w}) = \frac{1}{Z} \sum_{k=1}^K z_k \cdot e^{w_{k0} + \sum_{i=1}^n w_{ki} y_i}. \quad (9)$$

Under the normalization conditions

$$\sum_{k=1}^K e^{w_{k0}} = 1 \quad \text{and} \quad \forall k, j: \sum_{i \in G_j} e^{w_{ki}} = 1, \quad (10)$$

the normalization constant Z vanishes and the parametrization of the distribution simplifies to $w_{ki} = \log p(y_i = 1|z_k = 1, \mathbf{w})$ and $w_{k0} = \log p(z_k = 1|\mathbf{w})$. Even for non-normalized weights, the definition in Eq. (9) still represents the same type of distribution, although there is no more one-to-one mapping between the weights \mathbf{w} and the parameters of the graphical model (see Methods for details). Note also that such log-probabilities are exactly (up to additive constants) the local equilibrium points in Eq. (6) of the STDP rule in Fig. 2. In the section ‘‘STDP approximates Expectation Maximization’’ we will discuss in detail how this leads to unsupervised learning of a generative model of the input data in a WTA circuit.

Spike-based Bayesian computation. We can now formulate an exact link between the above generative probabilistic model and our neural network model of a simplified spike-based WTA circuit. We show that at any point in time t' at which the network generates an output spike, the relative firing probabilities $q_k(t')$ of the output neurons z_k as in Eq. (4), are equal to the posterior distribution of the hidden cause k , given the current evidences encoded in the input activations $\mathbf{y}(t')$. For a given input $\mathbf{y}(t')$ we use Bayes' rule to calculate the posterior probability of cause k as $p(k|\mathbf{y}(t'), \mathbf{w})$. We can identify the prior $p(k|\mathbf{w})$ with the excitabilities w_{k0} of the neurons. The log-likelihood

$\log p(\mathbf{y}(t')|k, \mathbf{w})$ of the current evidences given the cause k corresponds to the sum of excitatory EPSPs, which depend on the synaptic weights w_{ki} . This leads to the calculation

$$\begin{aligned} p(k|\mathbf{y}(t'), \mathbf{w}) &= \frac{\text{prior } p(k|\mathbf{w}) \cdot \text{likelihood } p(\mathbf{y}(t')|k, \mathbf{w})}{\sum_{k'=1}^K \underbrace{e^{w_{k'0} + \sum_{i=1}^n w_{k'i} y_i(t')}}_{p(\mathbf{y}(t')|\mathbf{w})}} \\ &= \frac{e^{w_{k0} + \sum_{i=1}^n w_{ki} y_i(t')}}{\sum_{k'=1}^K e^{w_{k'0} + \sum_{i=1}^n w_{k'i} y_i(t')}} = q_k(t'). \end{aligned} \quad (11)$$

This shows that at all times t' every spike from the WTA circuit represents one sample of the instantaneous posterior distribution $p(k|\mathbf{y}(t'), \mathbf{w})$.

The crucial observation, however, is that this relation is valid at any point in time, independently of the inhibitory signal $I(t)$. It is only the ratio between the quantities $e^{w_{ki}(t)}$ that determines the relative firing probabilities $q_k(t)$ of the neurons z_k .

Background oscillations and learning with missing values. We will now show that for the case of a low average input firing rate, a modulation of the firing rate can be beneficial, as it can synchronize firing of pre- and post-synaptic neurons. Each active neuron then fires according to an inhomogeneous Poisson process, and we assume for simplicity that the time course of the spike rate for all neurons follows the same oscillatory (sinusoidal) pattern around a common average firing rate. Nevertheless the spikes for each y_i are drawn as samples from independent processes. In addition, let the common inhibition signal $I(t)$ be modulated by an additional oscillatory current $I_{osc}(t) = A \cdot \sin(\omega t + \phi)$ with amplitude A , oscillation frequency ω (same as for the input oscillation), and phase shift ϕ . Due to the increased number of input neurons firing simultaneously, and the additional background current, pre- and post-synaptic firing of active neurons will synchronize. The frequency of the background oscillation can be chosen in principle arbitrarily, as long as the number of periods per input example is constant. Otherwise the network will weight different input examples by the number of peaks during presentation, which might lead to learning of a different generative model.

The effect of a synchronization of pre- and post-synaptic firing can be very beneficial, since at low input firing rates it might happen that none of the input neurons in a population of neurons encoding an external variable x_j fires within the integration time window of length σ of output neurons z_k . This corresponds to learning with missing attribute values for x_j , which is known to impair learning performance in graphical models [48]. Our novel interpretation is therefore that background oscillations can reduce the percentage of missing values by synchronizing presynaptic firing rates. This agrees with previous studies, which have shown that it is easier for single detector neurons learning with phenomenological STDP rules to detect spike patterns embedded in a high-dimensional input stream, if the patterns are encoded relative to a background oscillation [49], or the patterns consist of dense and narrow bursts of synchronous activity [50]. These results still hold if only a small part of the afferents participates in the pattern, or spikes from the pattern are missing, since the increased synchrony facilitates the identification of the pattern. Although we show in experiments that this increased synchronization can improve the learning performance of spike-based

probabilistic learners in practice, it is important to note that background oscillations are not necessary for the theory of spike-based Expectation Maximization to hold. Also, brain oscillations have previously been associated with various fundamental cognitive functions like e.g. attention, memory, consciousness, or neural binding. In contrast, our suggested role for oscillations as a mechanism for improving learning and inference with missing values is very specific within our framework, and although some aspects are compatible with higher-level theories, we do not attempt here to provide alternative explanations for these phenomena.

Our particular model of oscillatory input firing rates leaves the average firing rates unchanged, hence the effect of oscillations does not simply arise due to a larger number of input or output spikes. It is the increased synchrony of input and output spikes by which background oscillations can facilitate learning for tasks in which inputs have little redundancy, and missing values during learning thus would have a strong impact. We demonstrate this in the following experiment, where a common background oscillation for the input neurons y_i and the output neurons z_k significantly speeds up and improves the learning performance. In other naturally occurring input distributions with more structured inputs, oscillations might not improve the performance.

Example 1: Learning of probabilistic models with STDP

Fig. 3 demonstrates the emergence of Bayesian computation in the generic network motif of Fig. 1A in a simple example. Spike inputs \mathbf{y} (top row of Fig. 3D) are generated through four different hidden processes (associated with four different colors). Each of them is defined by a Gauss distribution over a 2D pixel array with a different center, which defines the probability of every pixel to be on. Spike trains encode the current value of a pixel by a firing rate of 25 Hz or 0 Hz for 40 ms. Each pixel was encoded by two input neurons y_i via population coding, exactly one of them had a firing rate of 25 Hz for each input image. A 10 ms period without firing separates two images in order to avoid overlap of EPSPs for input spikes belonging to different input images.

After unsupervised learning with STDP for 500 s (applied to continuous streams of spikes as in panel D of Fig. 3) the weight vectors shown in Fig. 3F (projected back into the virtual 2D input space) emerged for the four output neurons z_1, z_2, z_3, z_4 , demonstrating that these neurons had acquired internal models for the four different processes that were used to generate inputs. The four different processes for generating the underlying 2D input patterns had been used with different prior probabilities (0.1, 0.2, 0.3, 0.4). Fig. 3G shows that this imbalance resulted in four different priors $p(k)$ encoded in the biases $e^{w_{k0}}$ of the neurons z_k . When one compares the unequal sizes of the colored areas in Fig. 3H with the completely symmetric internal models (or likelihoods) of the four neurons shown in panel F, one sees that their firing probability approximates a posterior over hidden causes that results from multiplying their learned likelihoods with their learned priors. As a result, the spike output becomes sparser, and almost all neurons only fire when the current input spikes are generated by that one of the four hidden processes on which they have specialized (Fig. 3D, bottom row). In Fig. 3I the performance of the network is quantified over time by the normalized conditional entropy $H(k|\zeta_{out})/H(k, \zeta_{out})$, where k is the correct hidden cause of each input image \mathbf{y} in the training set, and ζ_{out} denotes the discrete random variable defined by the firing probabilities of output neurons z_k for each image under the currently learned model. Low conditional entropy indicates that each neuron learns to fire predominantly for inputs from one class. Fig. 3E as well as the dashed blue line in Fig. 3I show that the learning process is

improved when a common background oscillation at 20 Hz is superimposed on the firing rate of input neurons and the membrane potential of the output neurons, while keeping the average input and output firing rates constant. The reason is that in general it may occur that an output neuron z_k receives during its integration time window (40 ms in this example) no information about the value of a pixel (because neither the neuron y_i that has a high firing rate for 40 ms if this pixel is black, nor the associated neuron y_j that has a high firing rate if this pixel is white fire during this time window). A background oscillation reduces the percentage of such missing values by driving presynaptic firing times together (see top row of Fig. 3E). Note that through these oscillations the overall output firing rate $R(t)$ fluctuates strongly, but since the same oscillation is used consistently for all four types of patterns, the circuit still learns the correct distribution of inputs.

This task had been chosen to become very fast unsolvable if many pixel values are missing. Many naturally occurring input distributions, like the ones addressed in the subsequent computer experiments, tend to have more redundancy, and background oscillations did not improve the learning performance for those.

STDP approximates Expectation Maximization

In this section we will develop the link between the unsupervised learning of the generative probabilistic model in Fig. 1B and the learning effect of STDP as defined in our spiking network model in Fig. 1A. Starting from a learning framework derived from the concept of Expectation Maximization [31], we show that the biologically plausible STDP rule from Fig. 2 can naturally approximate a stochastic, online version of this optimization algorithm. We call this principle SEM (spike-based EM).

SEM can be viewed as a bootstrapping procedure. The relation between the firing probabilities of the neurons within the WTA circuit and the continuous updates of the synaptic weights with our STDP rule in Eq. (5) drive the initially random firing of the circuit in response to an input \mathbf{y} towards learning the correct generative model of the input distribution. Whenever a neuron z_k fires in response to \mathbf{y} , the STDP rule increases the weights w_{ki} of synapses from those presynaptic neurons y_i that had fired shortly before z_k . In absence of a recent presynaptic spike from y_i the weight w_{ki} is decreased. As a consequence, when next a pattern similar to \mathbf{y} is presented, the probability for the same z_k to fire and further adapt its weights, is increased. Since z_k becomes more of an “expert” for one subclass of input patterns, it actually becomes less likely to fire for non-matching patterns. The competition in the WTA circuit ensures that other z -neurons learn to specialize for these different input categories.

In the framework of Expectation Maximization, the generation of a spike in a z -neuron creates a sample from the currently encoded posterior distribution of hidden variables, and can therefore be viewed as the stochastic Expectation, or E-step. The subsequent application of STDP to the synapses of this neuron can be understood as an approximation of the Maximization, or M-step. The online learning behavior of the network can be understood as a stochastic online EM algorithm.

Learning the parameters of the probability model by EM. The goal of learning the parameterized generative probabilistic model $p(\mathbf{y}, k|\mathbf{w})$ is to find parameter values \mathbf{w} , such that the marginal distribution $p(\mathbf{y}|\mathbf{w})$ of the model distribution approximates the actual stationary distribution of spike inputs $p^*(\mathbf{y})$ as closely as possible. We define $p^*(\mathbf{y})$ as the probability to observe the activation vector $\mathbf{y}(t)$ at some point t in time (see Eq. (72) in Methods for a precise mathematical definition). The learning task can thus be formalized as the minimization of the Kullback-Leibler divergence between the two distributions, $p(\mathbf{y}|\mathbf{w})$ and

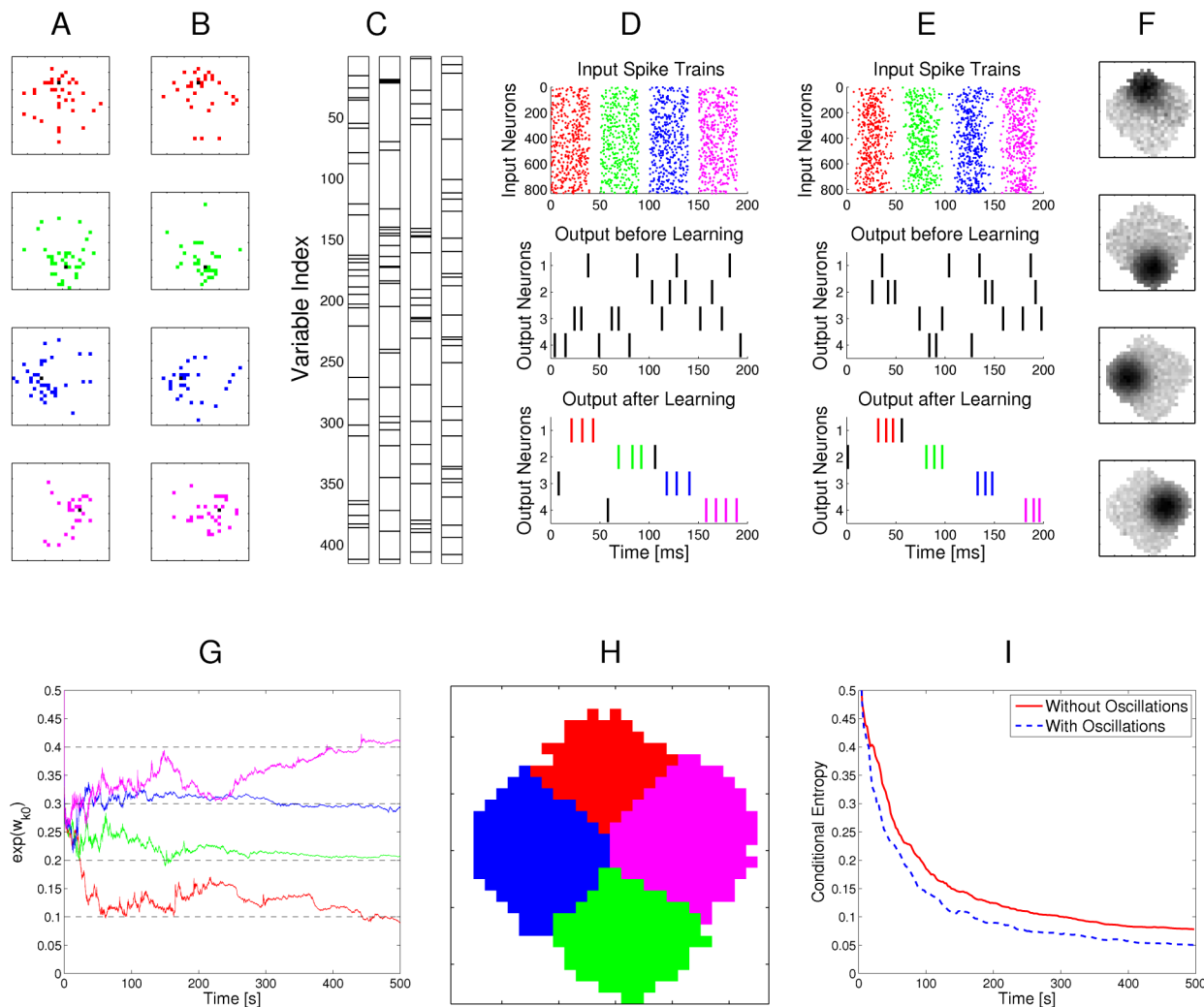


Figure 3. Example for the emergence of Bayesian computation through STDP and adaptation of neural excitability. **A, B:** Visualization of hidden structure in the spike inputs \mathbf{y} shown in D, E: Each row in panels A and B shows two results of drawing pixels from the same Gauss distribution over a 28×28 pixel array. Four different Gauss distributions were used in the four rows, and the location of their center represents the latent variable behind the structure of the input spike train. **C:** Transformation of the four 2D images in B into four linear arrays, resulting from random projections from 2D locations to 1D indices. Black lines indicate active pixels, and pixels that were active in less than 4% of all images were removed before the transformation (these pixels are white in panel H). By the random projection, both the 2D structure of the underlying pixel array and the value of the latent variable are hidden when the binary 1D vector is encoded through population coding into the spike trains \mathbf{y} that the neural circuit receives. **D:** Top row: Spike trains from 832 input neurons that result from the four linear patterns shown in panel C (color of spikes indicates which of the four hidden processes had generated the underlying 2D pattern, after 50 ms another 2D pattern is encoded). The middle and bottom row show the spike output of the four output neurons at the beginning and after 500 s of unsupervised learning with continuous spike inputs (every 50 ms another 2D pattern was randomly drawn from one of the 4 different Gauss distributions, with different prior probabilities of 0.1, 0.2, 0.3, and 0.4). Color of spikes indicates the emergent specialization of the four output neurons on the four hidden processes for input generation. Black spikes indicate incorrect guesses of hidden cause. **E:** Same as D, but with a superimposed 20 Hz oscillation on the firing rates of input neurons and membrane potentials of the output neurons. Fewer error spikes occur in the output, and output spikes are more precisely timed. **F:** Internal models (weight vectors \mathbf{w}) of output neurons z_1, \dots, z_4 after learning (pixel array). **G:** Autonomous learning of priors $p(k) \approx e^{w_{k0}}$, that takes place simultaneously with the learning of internal models. **H:** Average “winner” among the four output neurons for a test example (generated with equal probability by any of the 4 Gaussians) when a particular pixel was drawn in this test example, indicating the impact of the learned priors on the output response. **I:** Emergent discrimination capability of the output neurons during learning (red curve). The dashed blue curve shows that a background oscillation as in E speeds up discrimination learning. Curves in G and I represent averages over 20 repetitions of the learning experiment. doi:10.1371/journal.pcbi.1003037.g003

$p^*(\mathbf{y})$. A mathematically equivalent formulation is the maximization of the expected likelihood $L(\mathbf{w}) = E_{p^*}[\log p(\mathbf{y}|\mathbf{w})]$ of the inputs \mathbf{y} , drawn from $p^*(\mathbf{y})$. The parametrization of the generative probabilistic model $p(\mathbf{y}, k|\mathbf{w})$ is highly redundant, i.e. for every \mathbf{w} there is a continuous manifold of \mathbf{w}' , that all define identical generative distributions $p(\mathbf{y}, k|\mathbf{w}')$ in Eq. (24). There is, however,

exactly one \mathbf{w}' in this sub-manifold of the weight space that fulfills the normalization conditions in Eq. (10). By imposing the normalization conditions as constraints to the maximization problem, we can thus find unique local maxima (see “Details to Learning the parameters of the probability model by EM” in Methods).

The most common way to solve such unsupervised learning problems with hidden variables is the mathematical framework of Expectation Maximization (EM). In its standard form, the EM algorithm is a batch learning mechanism, in which a fixed, finite set of T instances of input vectors $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(T)}$ is given, and the task is to find the parameter vector \mathbf{w} that maximizes the log-likelihood $L(\mathbf{w}) = \sum_{l=1}^T \log p(\mathbf{y}^{(l)}|\mathbf{w})$ of these T instances to be generated as independent samples by the model $p(\mathbf{y}|\mathbf{w})$.

Starting from a random initialization for \mathbf{w} , the algorithm iterates between E-steps and M-steps. In the E-steps, the current parameter vector \mathbf{w} is used to find the posterior distributions of the latent variables $k^{(1)}, \dots, k^{(T)}$, each given by $p(k^{(l)}|\mathbf{y}^{(l)}, \mathbf{w})$.

In the M-steps a new parameter vector \mathbf{w}^{new} is computed, which maximizes the expected value of the complete-data log-likelihood function, subject to the normalization constraints in Eq. (10). The analytical solution for this M-step (compare [32]) is given by

$$\begin{aligned} w_{ki}^{\text{new}} &:= \log \frac{\sum_{l=1}^T y_i^{(l)} p(k|\mathbf{y}^{(l)}, \mathbf{w})}{\sum_{l=1}^T p(k|\mathbf{y}^{(l)}, \mathbf{w})} \text{ and} \\ w_{k0}^{\text{new}} &:= \log \frac{\sum_{l=1}^T p(k|\mathbf{y}^{(l)}, \mathbf{w})}{T}. \end{aligned} \quad (12)$$

The iterated application of this update procedure is guaranteed to converge to a (local) maximum of $L(\mathbf{w})$ [31]. It is obvious that \mathbf{w}^{new} fulfills the desired normalization conditions in Eq. (10) after every update.

Although the above deterministic algorithm requires that the same set of T training examples is re-used for every EM iteration, similar results also hold valid for online learning scenarios. In an online setup new samples $\mathbf{y}(l) \propto p^*(\mathbf{y})$ are drawn from the input distribution at every iteration, which is closer to realistic neural network learning settings. Instead of analytically computing the expected value of the complete-data log-likelihood function, a Monte-Carlo estimate is computed using the samples $k^{(l)}$, drawn according to their posterior distribution $p(k|\mathbf{y}^{(l)}, \mathbf{w})$. Even though additional stochastic fluctuations are introduced due to the stochastic sampling process, this stochastic EM algorithm will also converge to a stable result in the limit of infinite iterations, if the number of samples T is increased with every iteration [51].

In order to simplify the further notation we introduce the augmented input distribution $p_{\mathbf{w}}^*(\mathbf{y}, \mathbf{z})$ from which we can sample pairs $\langle \mathbf{y}, \mathbf{z} \rangle$ and define

$$p_{\mathbf{w}}^*(\mathbf{y}, \mathbf{z}) = p(\mathbf{z}|\mathbf{y}, \mathbf{w}) p^*(\mathbf{y}). \quad (13)$$

Sampling pairs $\langle \mathbf{y}^{(l)}, \mathbf{z}^{(l)} \rangle$ with $l = 1, \dots, T$ from $p_{\mathbf{w}}^*(\mathbf{y}, \mathbf{z})$ corresponds to online sampling of inputs, combined with a stochastic E-step. The subsequent M-step

$$w_{ki}^{\text{new}} := \log \frac{\sum_{l=1}^T y_i^{(l)} z_k^{(l)}}{\sum_{l=1}^T z_k^{(l)}}, \quad w_{k0}^{\text{new}} := \log \frac{\sum_{l=1}^T z_k^{(l)}}{T} \quad (14)$$

essentially computes averages over all T samples: $\exp(w_{k0}^{\text{new}})$ is the average of the variable z_k ; $\exp(w_{ki}^{\text{new}})$ is a conditional average of y_i taken over those instances in which z_k is 1.

The expected value of the new weight vector after one iteration, i.e., the sampling E-step and the averaging M-step, can be expressed in a very compact form based on the augmented input distribution as

$$E_{p_{\mathbf{w}}^*}[w_{ki}^{\text{new}}] = \log p_{\mathbf{w}}^*(y_i = 1 | z_k = 1) \quad E_{p_{\mathbf{w}}^*}[w_{k0}^{\text{new}}] = \log p_{\mathbf{w}}^*(z_k = 1). \quad (15)$$

A necessary condition for a point convergence of the iterative algorithm is a stable equilibrium point, i.e. a value \mathbf{w} at which the expectation of the next update \mathbf{w}^{new} is identical to \mathbf{w} . Thus we arrive at the following necessary implicit condition for potential convergence points of this stochastic algorithm.

$$w_{ki} = \log p_{\mathbf{w}}^*(y_i = 1 | z_k = 1) \quad w_{k0} = \log p_{\mathbf{w}}^*(z_k = 1). \quad (16)$$

This very intuitive implicit “solution” is the motivation for relating the function of the simple STDP learning rule (solid red line in Fig. 2) in the neural circuit shown in Fig. 1A to the framework of EM.

Spike-based Expectation Maximization. In order to establish a mathematically rigorous link between the STDP rule in Fig. 2 in the spike-based WTA circuit and stochastic online EM we identify the functionality of both the E- and the M-steps with the learning behavior of the spiking WTA-circuit with STDP.

In a biologically plausible neural network setup, one cannot assume that observations are stored and computations necessary for learning are deferred until a suitable sample size has been reached. Instead, we relate STDP learning to online learning algorithms in the spirit of Robbins-Monro stochastic approximations, in which updates are performed after every observed input.

At an arbitrary point in time t^l at which any one neuron z_k of the WTA circuit fires, the posterior $p(k|\mathbf{y}(t), \mathbf{w})$ according to Eq. (4) gives the probability that the spike at time t^l has originated from the neuron with index k . The pair $\langle \mathbf{y}(t), k \rangle$ can therefore be seen as a sample from the augmented input distribution $p_{\mathbf{w}}^*(\mathbf{y}, k)$. Hence, we can conclude that the generation of a spike by the WTA circuit corresponds to the generation of samples $\langle \mathbf{y}, k \rangle$ during the E-step. There are additional conditions on the inhibition signal $I(t)$ that have to be met in order to generate unbiased samples $\mathbf{y}(t^l)$ from the input distribution $p^*(\mathbf{y})$. These are discussed in depth in the section “Role of the Inhibition”, but for now let us assume that these conditions are fulfilled.

The generation of a spike in the postsynaptic neuron z_k triggers an STDP update according to Eq. (5) in all synapses from incoming presynaptic neurons y_i , represented by weights w_{ki} . We next show that the biologically plausible STDP rule in Eq. (5) (see also Fig. 2) together with the rule in Eq. (7) can be derived as approximating the M-step in stochastic online EM.

The update in Eq. (14) suggests that every synapse w_{ki} collects the activation statistics of its input y_i (the presynaptic neuron), given that its output z_k (the postsynaptic neuron) fires. These statistics can be gathered online from samples of the augmented input distribution $p_{\mathbf{w}}^*(\mathbf{y}, \mathbf{z})$.

From this statistical perspective each weight can be interpreted as $w_{ki} = \log \frac{a_{ki}}{N_{ki}}$, where a_{ki} and N_{ki} are two local virtual counters in each synapse. a_{ki} represents the number of the events $\langle y_i = 1, z_k = 1 \rangle$ and N_{ki} represents the number of the events $\langle z_k = 1 \rangle$, i.e. the postsynaptic spikes. Even though all virtual counters N_{ki} within one neuron z_k count the same postsynaptic spikes, it is easier to think of one individual such counter for every synapse. If we interpret the factor $\frac{1}{N_{ki}}$ as a local learning rate η_{ki} , we can derive Eq. (5) (see Methods) as the spike-event triggered stochastic online learning rule $w_{ki}^{\text{new}} = w_{ki} + \eta_{ki} z_k (y_i e^{-w_{ki}} - 1)$ that approximates in the synapse w_{ki} the log of the running average of $y_i(t^l)$ at the spiking times of neuron z_k . The update formula shows

that w_{ki} is only changed, if the postsynaptic neuron z_k fires, whereas spike events of other neurons $\langle z_{k'} = 1 \rangle$ with $k' \neq k$ are irrelevant for the statistics of w_{ki} . Thus the learning rule is purely local for every synapse w_{ki} ; it only has to observe its own pre- and postsynaptic signals. Additionally we show in the Methods section “Adaptive learning rates with Variance tracking” a very efficient heuristic how the learning rate ηa_{ki} can be estimated locally.

Analogously we can derive the working mechanism of the update rule in Eq. (7) as updates of the log of a fraction at the respective points in time.

The simple STDP rules in Eq. (5) and Eq. (7) thus approximate the M-step in a formal generative probabilistic model with local, biologically plausible computations. It remains to be shown that these STDP rules actually drive the weights \mathbf{w} to converge to the target points in Eq. (16) of the stochastic EM algorithm.

We can conclude from the equilibrium conditions of the STDP rule in Eq. (6) that convergence can only occur at the desired local maxima of the likelihood $L(\mathbf{w})$ subject to the normalization constraints. However, it remains to be shown that the update algorithm converges at all and that there are no limit cycles.

Proof of convergence. Even though we successfully identified the learning behavior of the simple STDP rule (Fig. 2) in the circuit model with the E- and the M-steps of the EM algorithm, this is not yet sufficient for a complete proof of convergence for the whole learning system. Not only are the single updates just approximations to the M-step, these approximations, in addition, violate the normalization conditions in Eq. (10). Although the system - as we will show - converges towards normalized solutions, there is always a stochastic fluctuation around the normalization conditions. One can therefore not simply argue that Eq. (5) implements a stochastic version of the generalized EM algorithm; instead, we have to resort to the theory of stochastic approximation algorithms as presented in [52]. Under some technical assumptions (see Methods) we can state

Theorem 1: *The algorithm in Eq. (5,7) updates \mathbf{w} in a way that it converges with probability 1 to the set of local maxima of the likelihood function $L(\mathbf{w}) = E_p[\log p(\mathbf{y}|\mathbf{w})]$, subject to the normalization constraints in Eq. (10).*

The detailed proof, which is presented in Methods, shows that the expected trajectory of the weight vector \mathbf{w} is determined by two driving forces. The first one is a normalization force which drives \mathbf{w} from every arbitrary point towards the regime where \mathbf{w} is normalized. The second force is the real learning force that drives \mathbf{w} to a desired maximum of $L(\mathbf{w})$. However, this interpretation of the learning force is valid only if \mathbf{w} is sufficiently close to normalized.

The role of the inhibition

We have previously shown that the output spikes of the WTA circuit represent samples from the posterior distribution in Eq. (11), which only depends on the ratios between the membrane potentials $u_k(t)$. The rate at which these samples are produced is the overall firing rate $R(t)$ of the WTA circuit and can be controlled by modifying the common inhibition $I(t)$ of the neurons z_k .

Although any time-varying output firing rate $R(t)$ produces correct samples from the posterior distribution in Eq. (11) of \mathbf{z} , for learning we also require that the input patterns $\mathbf{y}(t)$ observed at the spike times are unbiased samples from the true input distribution $p^*(\mathbf{y})$. If this is violated, some patterns coincide with a higher $R(t)$, and thus have a stronger influence on the learned synaptic weights. In Methods we formally show that $R(t)$ acts as a multiplicative weighting of the current input $\mathbf{y}(t)$, and so the generative model will learn a slightly distorted input distribution.

An unbiased set of samples can be obtained if $R(t)$ is independent of the current input activation $\mathbf{y}(t)$, e.g. if $R(t) = R$ is constant. This could in theory be achieved if we let $I(t)$ depend on the current values of the membrane potentials $u_k(t)$, and set $I(t) = -\log R + \log \sum_{k=1}^K e^{u_k(t)}$. Such an immediate inhibition is commonly assumed in rate-based soft-WTA models, but it seems implausible to compute this in a spiking neuronal network, where only spikes can be observed, but not the presynaptic membrane potentials.

However, our results show that a perfectly constant firing rate is not a prerequisite for convergence to the right probabilistic model. Indeed we can show that it is sufficient that $R(t)$ and $\mathbf{y}(t)$ are stochastically independent, i.e. $R(t)$ is not correlated to the appearance of any specific value of $\mathbf{y}(t)$. Still this might be difficult to achieve since the firing rate $R(t)$ is functionally linked to the input $\mathbf{y}(t)$ by $R(t) = e^{-I(t)} Z p(\mathbf{y}(t)|\mathbf{w})$, but it clarifies the role of the inhibition $I(t)$ as de-correlating $R(t)$ from the input \mathbf{y} , at least in the long run.

One possible biologically plausible mechanism for such a decorrelation of $R(t)$ and $\mathbf{y}(t)$ is an inhibitory feedback from a population of neurons that is itself excited by the neurons z_k . Such WTA competition through lateral inhibition has been studied extensively in the literature [16,33]. In the implementation used for the experiments in this paper every spike from the z -neurons causes an immediate very strong inhibition signal that lasts longer than the refractory period of the spiking neuron. This strong inhibition decays exponentially and is overlaid by a noise signal with high variability that follows an Ornstein-Uhlenbeck process (see “Inhibition Model in Computer Simulations” in Methods). This will render the time of the next spike of the system almost independent of the value of $p(\mathbf{y}(t)|\mathbf{w})$.

It should also be mentioned that a slight correlation between $R(t)$ and $p(\mathbf{y}(t)|\mathbf{w})$ may be desirable, and $I(t)$ might also be externally modulated (for example through attention, or neuromodulators such as Acetylcholin), as an instrument of selective input learning. This might lead e.g. to slightly higher firing rates for well-known inputs (high $p(\mathbf{y}(t)|\mathbf{w})$), or salient inputs, as opposed to reduced rates for unknown arbitrary inputs. In general, however, combining online learning with a sampling rate $R(t)$ that is correlated to $p(\mathbf{y}|\mathbf{w})$ may lead to strange artifacts and might even prohibit the convergence of the system due to positive feedback effects. A thorough analysis of such effects and of possible learning mechanisms that cope with positive feedback effects is the topic of future research.

Our theoretical analysis sheds new light on the requirements for inhibition in spiking WTA-like circuits to support learning and Bayesian computation. Inhibition does not only cause competition between the excitatory neurons, but also regulates the overall firing rate $R(t)$ of the WTA circuit. Variability in $R(t)$ does not influence the performance of the circuit, as long as there is no systematic dependence between the input and $R(t)$.

Continuous-time interpretation with realistically shaped EPSPs

In our previous analysis we have assumed a simplified non-additive step-function model for the EPSP. This allowed us to describe all input evidence within the last time window of length σ by one binary vector $\mathbf{y}(t)$, but required us to assume that no two neurons within the same group G_j fired within that period. We will now give an intuitive explanation to show that this restriction can be dropped and present an interpretation for additive biologically plausibly shaped EPSPs as inference in a generative model.

The postsynaptic activation $\tilde{y}_i(t)$ under an additive EPSPs is given by the convolution

$$\tilde{y}_i(t) = \sum_j K(t - t'_i), \quad (17)$$

where K describes an arbitrarily shaped kernel, e.g. an α -shaped EPSP function which is the difference of two exponential functions (see [37]) with different time constants. We use 1 ms for the rise and 15 ms for the decay in our simulations. $\tilde{y}_i(t)$ replaces $y_i(t)$ in Eq. (2) in the computation of the membrane potential $u_k(t)$ of our model neurons. We can still understand the firing of neurons in the WTA circuit according to the relative firing probabilities $q_k(t)$ in Eq. (4) as Bayesian inference. To see this, we imagine an extension of the generative probabilistic model $p(\mathbf{x}, k | \theta)$ in Fig. 1B, which contains multiple instances of \mathbf{x} , exactly one for every input spike from all input neurons y_i . For a fixed common hidden cause k , all instances of \mathbf{x} are conditionally independent of each other, and have the same conditional distributions for each x_j (see Methods for the full derivation of the extended probabilistic model). According to the definition in Eq. (8) of the population code every input spike represents evidence that x_j in an instance \mathbf{x} should take on a certain value. Since every spike contributes only to one instance, any finite input spike pattern can be interpreted as valid evidence for multiple instances of inputs \mathbf{x} .

The inference of a single hidden cause k in such extended graphical model from multiple instances of evidence is relatively straightforward: due to the conditional independence of different instances, we can compute the input likelihood for any hidden cause simply as the product of likelihoods for every single evidence. Inference thus reduces to counting how often every possible evidence occurred in all instances \mathbf{x} , which means counting the number of spikes of every y_i . Since single likelihoods are implicitly encoded in the synaptic weights w_{ki} by the relationship $w_{ki} = \log p(v_i = 1 | k, \mathbf{w})$, we can thus compute the complete input likelihood by adding up step-function like EPSPs with amplitudes corresponding to w_{ki} . This yields correct results, even if one input neuron spikes multiple times.

In the above model, the timing of spikes does not play a role. If we want to assign more weight to recent evidence, we can define a heuristic modification of the extended graphical model, in which contributions from spikes to the complete input log-likelihood are linearly interpolated in time, and multiple pieces of evidence simply accumulate. This is exactly what is computed in \tilde{y}_i in Eq. (17), where the shape of the kernel $K(t - t')$ defines how the contribution of an input spike at time t' evolves over time. Defining \tilde{y}_i as the weight for the evidence of the assignment of x_j to value $v(i)$, it is easy to see (and shown in detail in Methods) that the instantaneous output distribution $q_k(t)$ represents the result of inference over causes k , given the time-weighted evidences of all previous input spikes, where the weighting is done by the EPSP-function $K(t)$. Note that this evidence weighting mechanism is not equivalent to the much more complex mechanism for inference in presence of uncertain evidence, which would require more elaborate architectures than our feed-forward WTA-circuit. In our case, past evidence does not become uncertain, but just less important for the inference of the instantaneous hidden cause k .

We can analogously generalize the spike-triggered learning rule in Eq. (5) for continuous-valued input activations $\tilde{y}_i(t)$ according to Eq. (17):

$$\Delta w_{ki}(t) = \tilde{y}_i(t) \cdot c \cdot e^{-w_{ki}} - 1. \quad (18)$$

The update of every weight w_{ki} is triggered when neuron z_k , i.e. the postsynaptic neuron, fires a spike. The shape of the LTP part of the STDP curve is determined by the shape of the EPSP, defined by the kernel function $K(t)$. The positive part of the update in Eq. (18) is weighted by the value of $\tilde{y}_i(t)$ at the time of firing the postsynaptic spike. Negative updates are performed if $\tilde{y}_i(t)$ is close to zero, which indicates that no presynaptic spikes were observed recently. The complex version of the STDP curve (blue dashed curve in Fig. 1B), which resembles more closely to the experimentally found STDP curves, results from the use of biologically plausible α -shaped EPSPs. In this case, the LTP window of the weight update decays with time, following the shape of the α -function. This form of synaptic plasticity was used in all our experiments. If EPSPs accumulate due to high input stimulation frequencies, the resulting shape of the STDP curve becomes even more similar to previously observed experimental data, which is investigated in detail in the following section.

The question remains, how this extension of the model and the heuristics for time-dependent weighting of spike contributions affect the previously derived theoretical properties. Although the convergence proof does not hold anymore under such general conditions we can expect (and show in our Experiments) that the network will still show the principal behavior of EM under fairly general assumptions on the input: we have to assume that the instantaneous spike rate of every input group G_j is not dependent on the value of x_j that it currently encodes, which means that the total input spike rate must not depend on the hidden cause k . Note that this assumption on every input group is identical to the desired output behavior of the WTA circuit according to the conditions on the inhibition as derived earlier. This opens up the possibility of building networks of recursively or hierarchically connected WTA circuits. Note also that the grouping of inputs into different G_j is only a notational convenience. The neurons in the WTA circuit do not have to know which inputs are from the same group, neither for inference nor for learning, and can thus treat all input neurons equally.

Relationship to experimental data on synaptic plasticity

In biological STDP experiments that induce pairs of pre- and post-synaptic spikes at different time delays, it has been observed that the shape of the plasticity curve changes as a function of the repetition frequency for those spike pairs [40]. The observed effect is that at very low frequencies no change or only LTD occurs, a “classical” STDP window with timing-dependent LTD and LTP is observed at intermediate frequencies around 20 Hz, and at high frequencies of 40 Hz or above only LTP is observed, independently of which spikes comes first.

Although our theoretical model does not explicitly include a stimulation-frequency dependent term like other STDP models (e.g. [53]), we can study empirically the effect of a modification of the frequency of spike-pairing. We simulate this for a single synapse, at which we force pre- and post-synaptic spikes with varying time differences $\Delta t = t_{\text{post}} - t_{\text{pre}}$, and at fixed stimulation frequencies f of either 1 Hz, 20 Hz, or 40 Hz. Modeling EPSPs as α -kernels with time constants of 1 ms for the rise and 15 ms for the decay, we obtain the low-pass filtered signals \tilde{y}_i as in Eq. (17), which grow as EPSPs start to overlap at higher stimulation frequencies. At the time of a post-synaptic spike we compute the synaptic update according to the rule in Eq. (18), but keep both the weight and the learning rate fixed (at $w_{ki} = 3.5, c = e^{-5}, \eta = 0.5$) to distinguish timing-dependent from weight-dependent effects.

In Fig. 4A we observe that, as expected, at low stimulation frequencies (1 Hz) the standard shape of the complex STDP rule in Eq. (18) from Fig. 2 is recovered, since there is no influence from

previous spikes. The shift towards pure LTD that is observed in biology [40] would require an additional term that depends on postsynaptic firing rates like in [53], and is a topic of future research. However, note that in biology this shift to LTD was observed only in paired recordings, neglecting the cooperative effect of other synapses, and other studies have also reported LTP at low stimulation frequencies [43]. At higher stimulation frequencies (20 Hz in Fig. 4B) the EPSPs from different pre-synaptic spikes start to overlap, which results in larger \tilde{y}_i compared with isolated pre-synaptic spikes. We also see that the LTD part of the STDP window becomes timing-dependent (due to overlapping EPSPs), and thus the shape of the STDP curve becomes similar to standard models of STDP and observed biological data [43,54]. For even higher stimulation frequencies the STDP window shifts more and more towards LTP (see Fig. 4B and C). This is in good accordance with observations in biology [40]. Also in agreement with biological data, the minimum of the update occurs around $\Delta t = 0$, because there the new α -kernel EPSP is not yet effective, and the activation due to previous spikes has decayed maximally.

Another effect that is observed in hippocampal synapses when two neurons are stimulated with bursts, is that the magnitude of LTP is determined mostly by the amount of overlap between the pre- and post-synaptic bursts, rather than the exact timing of spikes [55]. In Fig. 4D we simulated this protocol with our continuous-time SEM rule for different onset time-differences of the bursts, and accumulated the synaptic weight updates in response to 50 Hz bursts of 5 pre-synaptic and 4 post-synaptic spikes. We performed this experiment for the same onset time differences used in Fig. 3 of [55], and found qualitatively similar results. For long time-differences, when EPSPs have mostly decayed, we observed an LTD effect, which was not observed in biology, but can be attributed to differences in synaptic time constants between biology and simulation.

These results suggest that our STDP rule derived from theoretical principles exhibits several of the key properties of synaptic plasticity observed in nature, depending on the encoding of inputs. This is quite remarkable, since these properties are not explicitly part of our learning rule, but rather emerge from a simpler rule with strong theoretical guarantees. Other phenomenological [56,57] or mechanistic models of STDP [58] also show some of these characteristics, but come without such theoretical

properties. The functional consequence of reproducing such key biological characteristics of STDP is that our new learning rule also exhibits most of the key functional properties of STDP, like e.g. strengthening synapses of inputs that are causally involved in firing the postsynaptic neuron, while pruning the connections that do not causally contribute to postsynaptic firing [10,13]. At low and intermediate firing rates our rule also shifts the onset of postsynaptic firing towards the start of repeated spike patterns [49,50,59], while depressing synapses that only become active for a pattern following the one for which the post-synaptic neuron is responsive. If patterns change quickly, then the stronger depression for presynaptic spikes with small Δt in Fig. 4B enhances the capability of the WTA to discriminate such patterns. With simultaneous high frequency stimulation (Fig. 4C and D) we observe that only LTP occurs, which is due to the decay of EPSPs not being fast enough to allow depression. In this scenario, the learning rule is less sensitive to timing, and rather becomes a classical Hebbian measure of correlations between pre- and post-synaptic firing rates. However, since inputs are encoded in a population code we can assume that the same neuron is not continuously active throughout, and so even at high firing rates for active input neurons, the synapses that are inactive during postsynaptic firing will still be depressed, which means that convergence to an equilibrium value is still possible for all synapses.

It is a topic of future research which effects observed in biology can be reproduced with more complex variations of the spike-based EM rule that are also dependent on postsynaptic firing rates, or whether existing phenomenological models of STDP can be interpreted in the probabilistic EM framework. In fact, initial experiments have shown that several variations of the spike-based EM rule can lead to qualitatively similar empirical results for the learned models in tasks where the input spike trains are Poisson at average or high rates over an extended time window (such as in Fig. 3). These variations include weight-dependent STDP rules that are inversed in time, symmetrical in time, or have both spike timing-dependent LTD and LTP. Such rules can converge towards the same equilibrium values as the typical causal STDP rule. However, they will behave differently if inputs are encoded through spatio-temporal spike patterns (as in Example 4: Detection of Spatio-Temporal Spike Patterns). Further variations

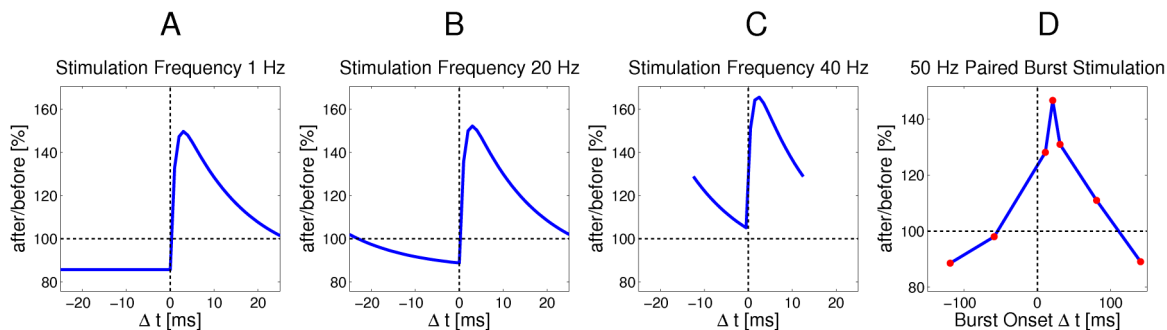


Figure 4. Relationship between the continuous-time SEM model and experimental data on synaptic plasticity. A–C: The effect of the continuous-time plasticity rule in Eq. (18) at a single synapse for different stimulation frequencies and different time-differences between pre- and post-synaptic spike pairs. Only time-intervals without overlapping pairs are shown. **A:** For very low stimulation frequencies (1 Hz) the standard shape of the complex learning rule from Fig. 2 is recovered. **B:** At a stimulation frequency of 20 Hz the plasticity curve shifts more towards LTP, and depression is no longer time independent, due to overlapping EPSPs. **C:** At high stimulation frequencies of 40 Hz or above, the STDP curve shifts towards only LTP, and thus becomes similar to a rate-based Hebbian learning rule. **D:** Cumulative effect of pre- and post-synaptic burst stimulation (50 Hz bursts of 5 pre-synaptic and 4 post-synaptic spikes) with different onset delays of -120, -60, 10, 20, 30, 80 and 140 ms (time difference between the onsets of the post- and pre-synaptic bursts). As in [55], the amount of overlap between bursts determines the magnitude of LTP, rather than the exact temporal order of spikes.

doi:10.1371/journal.pcbi.1003037.g004

can include short-term plasticity effects for pre-synaptic spikes, as observed and modeled in [60], which induce a stimulation-frequency dependent reduction of the learning rate, and could thus serve as a stabilization mechanism.

Spike-timing dependent LTD

Current models of STDP typically assume a “double-exponential” decaying shape of the STDP curve, which was first used in [54] to fit experimental data. This is functionally different from the shape of the complex STDP curve in Fig. 2 and Eq. (5), where the LTD part is realized by a constant timing-independent offset.

Although not explicitly covered by the previously presented theory of SEM, the same analytical tools can be used to explain functional consequences of timing-dependent LTD in our framework. Analogous to our approach for the standard SEM learning rule, we develop (in Methods) an extension of the simple step-function STDP rule from Fig. 2 with timing-dependent LTD, which is easier to analyze. We then generalize these results towards arbitrarily shaped STDP curves. The crucial result is that as long as the spike-timing dependent LTD rule retains the characteristic inversely-exponential weight-dependent relationship between the strengths of LTP and LTD that was introduced for standard SEM in Eq. (5), an equilibrium property similar to Eq. (6) still holds (see Methods for details). Precisely speaking, the new equilibrium will be at the difference between the logarithms of the average presynaptic spiking probabilities *before* and *after* the postsynaptic spike. This shows that spike-timing dependent LTD also yields synaptic weights that can be interpreted in terms of log-probabilities, which can thus be used for inference.

The new rule emphasizes contrasts between the current input pattern and the immediately following activity. Still, the results of the new learning rule and the original rule from Eq. (5) in our experiments are qualitatively similar. This can be explained from a stochastic learning perspective: at any point in time the relative spiking probabilities of excitatory neurons in the WTA circuit in Eq. (4) depend causally on the weighted sums of preceding presynaptic activities $\tilde{y}_i(t)$. However, they clearly do not depend on future presynaptic activity. Thus, the postsynaptic neuron will learn through SEM to fire for increasingly similar stochastic realizations of presynaptic input $\tilde{y}_i(t)$, whereas the presynaptic activity pattern following a postsynaptic spike will become more variable. In the extreme case where patterns are short and separated by noise, there will be no big difference between input patterns following firing of any of the WTA neurons, and so their relevance for the competition will become negligible.

Experimental evidence shows that the time constants of the LTP learning window are usually smaller than the time constants of the LTD window ([40,60]), which will further enhance the specificity of the LTP learning as opposed to the LTD part that computes the average over a longer window.

Note that the exponential weight dependence of the learning rule implies a certain robustness towards linearly scaling LTP or LTD strengths, which only leads to a constant offset of the weights. Assuming that the offset is the same for all synapses, this does not affect firing probabilities of neurons in a WTA circuit (see Methods “Weight offsets and positive weights”).

Example 2: Learning of probabilistic models for orientation selectivity

We demonstrated in this computer experiment the emergence of orientation selective cells z_k through STDP in the WTA circuit of Fig. 1A when the spike inputs encode isolated bars in arbitrary orientations. Input images were generated by the following process: Orientations were sampled from a uniform distribution,

and lines of 7 pixels width were drawn in a 28×28 pixel array. We added noise to the stimuli by flipping every pixel with a 10% chance, see Fig. 5A. Finally, a circular mask was applied to the images to avoid artifacts from image corners. Spikes trains \mathbf{y} were encoded according to the same population coding principle described in the previous example Fig. 3, in this case using a Poisson firing rate of 20 Hz for active units.

After training with STDP for 200 s, presenting 4000 different images, the projection of the learned weight vectors back into the 2D input space (Fig. 5B) shows the emergence of 10 models with different orientations, which cover the possible range of orientations almost uniformly. When we plot the strongest responding neuron as a function of orientation (Fig. 5C, D), measured by the activity in response to 360 noise-free images of oriented bars in 1° steps, we can see no structure in the response before learning (Fig. 5C). However, after unsupervised learning, panel D clearly shows the emergence of continuous, uniformly spaced regions in which one of the z_k neurons fires predominantly. This can also be seen in the firing behavior in response to the input spike trains in Fig. 5E, which result from the example images in panel A. Fig. 5F shows that the output neurons initially fire randomly in response to the input, and many different z_k neurons are active for one image. In contrast, the responses after learning in panel G are much sparser, and only occasionally multiple neurons are active for one input image, which is the case when the angle of the input image is in between the preferred angles of two output neurons, and therefore multiple models have a non-zero probability of firing.

In our experiment the visual input consisted of noisy images of isolated bars, which illustrates learning of a probabilistic model in which a continuous hidden cause (the orientation angle) is represented by a population of neurons, and also provides a simple model for the development of orientation selectivity. It has previously been demonstrated that similar Gabor-like receptive field structures can be learned with a sparse-coding approach using patches of natural images as inputs [61]. The scenario considered here is thus substantially simplified, since we do not present natural but isolated stimuli. However, it is worth noting that experimental studies have shown that (in mice and ferret) orientation selectivity, but not e.g. direction selectivity, exists in V1 neurons even before eye opening [62,63]. This initial orientation selectivity develops from innate mechanisms and from internally generated inputs during this phase [63], e.g. retinal waves, which have different, and very likely simpler statistics than natural stimuli. Our model shows that a WTA circuit could learn orientation selectivity from such simple bar-like inputs, but does not provide an alternative explanation to the results of studies like [61] using natural image stimuli. Although beyond the scope of this paper, we expect that later shaping of selectivity through exposure to natural visual experience would not alter the receptive fields by much, since the neurons have been primed to spike (and thereby trigger plasticity) only in response to a restricted class of local features.

Example 3: Emergent discrimination of handwritten digits through STDP

Spike-based EM is a quite powerful learning principle, as we demonstrate in Fig. 6 through an application to a computational task that is substantially more difficult than previously considered tasks for networks of spiking neurons: We show that a simple network of spiking neurons can learn without any supervision to discriminate handwritten digits from the MNIST benchmark dataset [64] consisting of 70,000 samples (30 are shown in Fig. 6A). This is one of the most frequently used benchmark tasks in machine learning. It has mostly been used to evaluate supervised

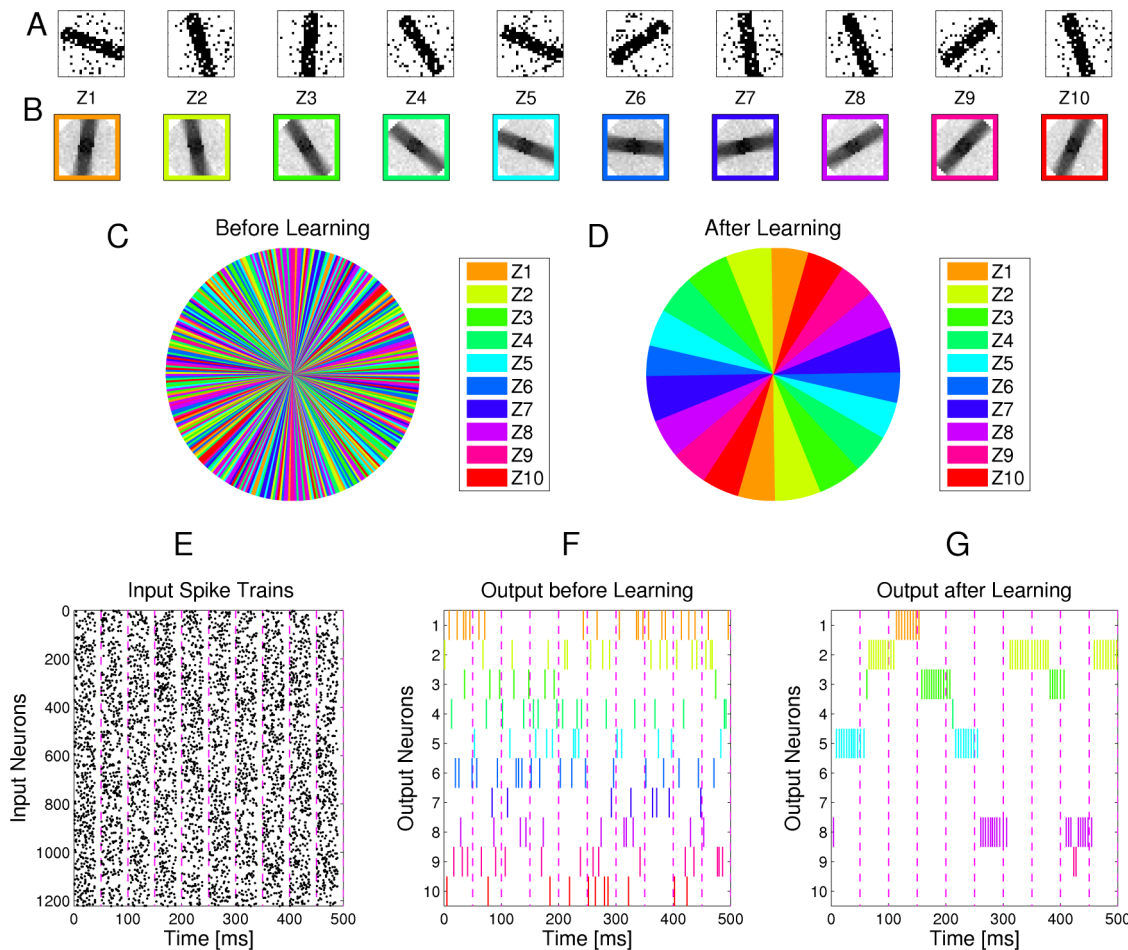


Figure 5. Emergence of orientation selective cells for visual input consisting of oriented bars with random orientations. **A** Examples of 28×28 -pixel input images with oriented bars and additional background noise. **B** Internal models (weight vectors of output neurons z_k) that are learned through STDP after the presentation of 4000 input images (each encoded by spike trains for 50 ms, as in Fig. 3). **C, D** Plot of the most active neuron for 360 images of bars with orientations from 0 to 360° in 1° steps. Colors correspond to the colors of z_k neurons in B. Before training (**C**), the $K = 10$ output neurons fire without any apparent pattern. After training (**D**) they specialize on different orientations and cover the range of possible angles approximately uniformly. **E**: Spike train encoding of the 10 samples in A. **F, G**: Spike trains produced by the $K = 10$ output neurons in response to these samples before and after learning with STDP for 200 s. Colors of the spikes indicate the identity of the output neuron, according to the color code in B.

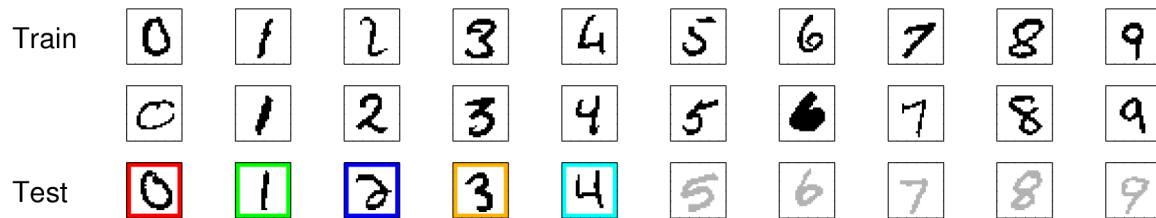
doi:10.1371/journal.pcbi.1003037.g005

or semi-supervised machine learning algorithms [27,65], or to evaluate unsupervised feature learning approaches [66,67]. Although the MNIST dataset contains labels (the intended digit) for each sample of a handwritten digit, we deleted these labels when presenting the dataset to the neural circuit of Fig. 1A, thereby forcing the $K = 100$ neurons on the output layer to self-organize in a completely unsupervised fashion. Each sample of a handwritten digit was encoded by 708 spike trains over 40 ms (and 10 ms periods without firing between digits to avoid overlap of EPSPs between images), similarly as for the task of Fig. 3. Each pixel was represented by two input neurons y_i , one of which produced a Poisson spike train at 40 Hz during these 40 ms. This yielded usually at most one or two spikes during this time window, demonstrating that the network learns and computes with information that is encoded through spikes, rather than firing rates. After 500 s of unsupervised learning by STDP almost all of the output neurons fired more sparsely, and primarily for handwritten samples of just one of the digits (see Fig. 6E).

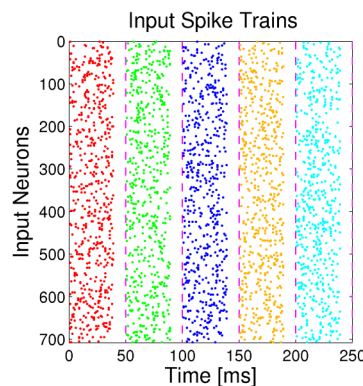
The application to the MNIST dataset had been chosen to illustrate the power of SEM in complex tasks. MNIST is one of

the most popular benchmarks in machine learning, and state-of-the-art methods achieve classification error rates well below 1%. The model learned by SEM can in principle also be used for classification, by assigning each neuron to the class for which it fires most strongly. However, since this is an unsupervised method, not optimized for classification but for learning a generative model, the performance is necessarily worse. We achieve an error rate of 19.86% on the 10-digit task on a previously unseen test set. This compares favorably to the 21% error that we obtained with a standard machine learning approach that directly learned the mixture-of-multinomials graphical model in Fig. 1B with a batch EM algorithm. This control experiment was not constrained by a neural network architecture or biologically plausible learning, but instead mathematically optimized the parameters of the model in up to 200 iterations over the whole training set. The batch method achieves a final conditional entropy of 0.1068, which is slightly better than the 0.1375 final result of the SEM approach, and shows that better performance on the classification task does not necessarily mean better unsupervised model learning.

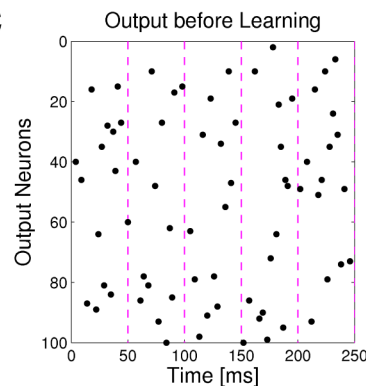
A



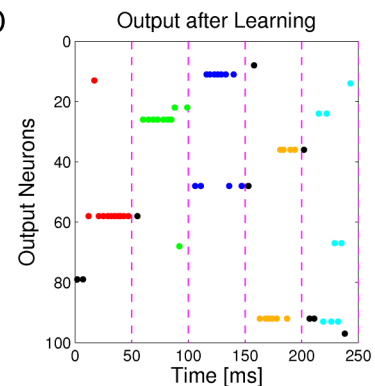
B



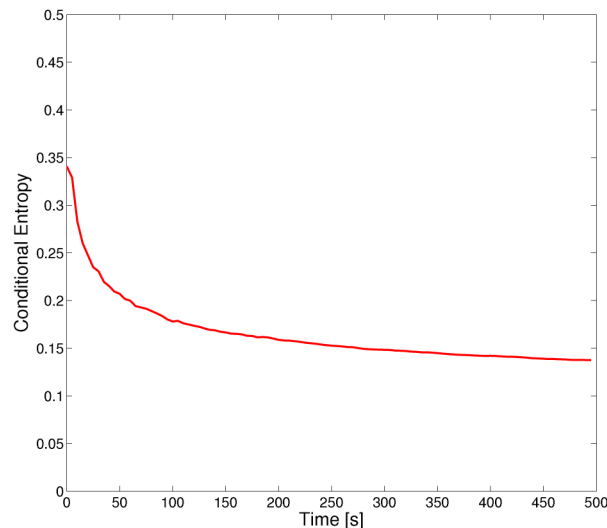
C



D



E



F



Figure 6. Emergent discrimination of handwritten digits through STDP. **A:** Examples of digits from the MNIST dataset. The third and fourth row contain test examples that had not been shown during learning via STDP. **B:** Spike train encoding of the first 5 samples in the third row of A. Colors illustrate the different classes of digits. **C, D:** Spike trains produced by the $K = 100$ output neurons before and after learning with STDP for 500 s. Colored spikes indicate that the class of the input and the class for which the neuron is mostly selective (based on human classification of its generative model shown in F) agree, otherwise spikes are black. **E:** Temporal evolution of the self-organization process of the 100 output neurons (for the complex version of STDP-curve shown in Fig. 1B), measured by the conditional entropy of digit labels under the learned models at different time points. **F:** Internal models generated by STDP for the 100 output neurons after 500 s. The network had not received any information about the number of different digits that exist and the colors for different ways of writing the first 5 digits were assigned by the human supervisor. On the basis of this assignment the test samples in row 3 of panel A had been recognized correctly.
doi:10.1371/journal.pcbi.1003037.g006

Example 4: Detection of Spatio-Temporal Spike Patterns

Our final application demonstrates that the modules for Bayesian computation that emerge in WTA circuits through STDP can not only explain the emergence of feature maps in

primary sensory cortices like in Fig. 5, but could also be viewed as generic computational units in generic microcircuits throughout the cortex. Such generic microcircuit receives spike inputs from many sources, and it would provide a very useful computational

operation on these if it could autonomously detect repeatedly occurring spatio-temporal patterns within this high-dimensional input stream, and report their occurrence through a self-organizing sparse coding scheme to other microcircuits. We have created such input streams with occasionally repeated embedded spike patterns for the computer experiment reported in Fig. 7. Fig. 7D demonstrates that sparse output codes for the 5 embedded spike patterns emerge after applying STDP in a WTA circuit for 200 s to such input stream. Furthermore, we show in the Supplement that these sparse output codes generalize (even without any further training) to time-warped versions of these spike patterns.

Even though our underlying probabilistic generative model (Fig. 1B) does not include time-dependent terms, the circuit in this example performs inference over time. The reason for this is that synapses that were active when a neuron fired become reinforced by STDP, and therefore make the neuron more likely to fire again when a similar spatial pattern is observed. Since we use EPSPs that smoothly decay over time, one neuron still sees a trace of previous input spikes as it fires again, and thus different spatial patterns within one reoccurring spatio-temporal pattern are recognized by the same neuron. The maximum length for such patterns is

determined by the time constants of EPSPs. With our parameters (1 ms rise, 15 ms decay time constant) we were able to recognize spike patterns up to 50–100 ms. For longer spatio-temporal patterns, different neurons become responsive to different parts of the pattern. The neuron that responds mostly to noise in Figs. 7D did not learn a specific spatial pattern, and therefore wins by default when none of the specialized neurons responds. Similar effects have previously been described [59,68], but for different neuron models, classical STDP curves, and not in the context of probabilistic inference.

For this kind of task, where also the exact timing of spikes in the patterns matters (which is not necessarily the case in the examples in Figs. 3, 5, and 6, where input neurons generate Poisson spike trains with different rates), we found that the shape of the STDP kernel plays a larger role. For example, a time-inverted version of the SEM rule, where pre-before-post firing causes LTD instead of LTP, cannot learn this kind of task, because once a neuron has learned to fire for a sub-pattern of the input, its firing onset is shifted back in time, rather than forward in time, which happens with standard SEM, but also with classical STDP [50,59]. Instead, with a time-inverted SEM rule, different neurons would learn to fire stronger for the offsets of different patterns.

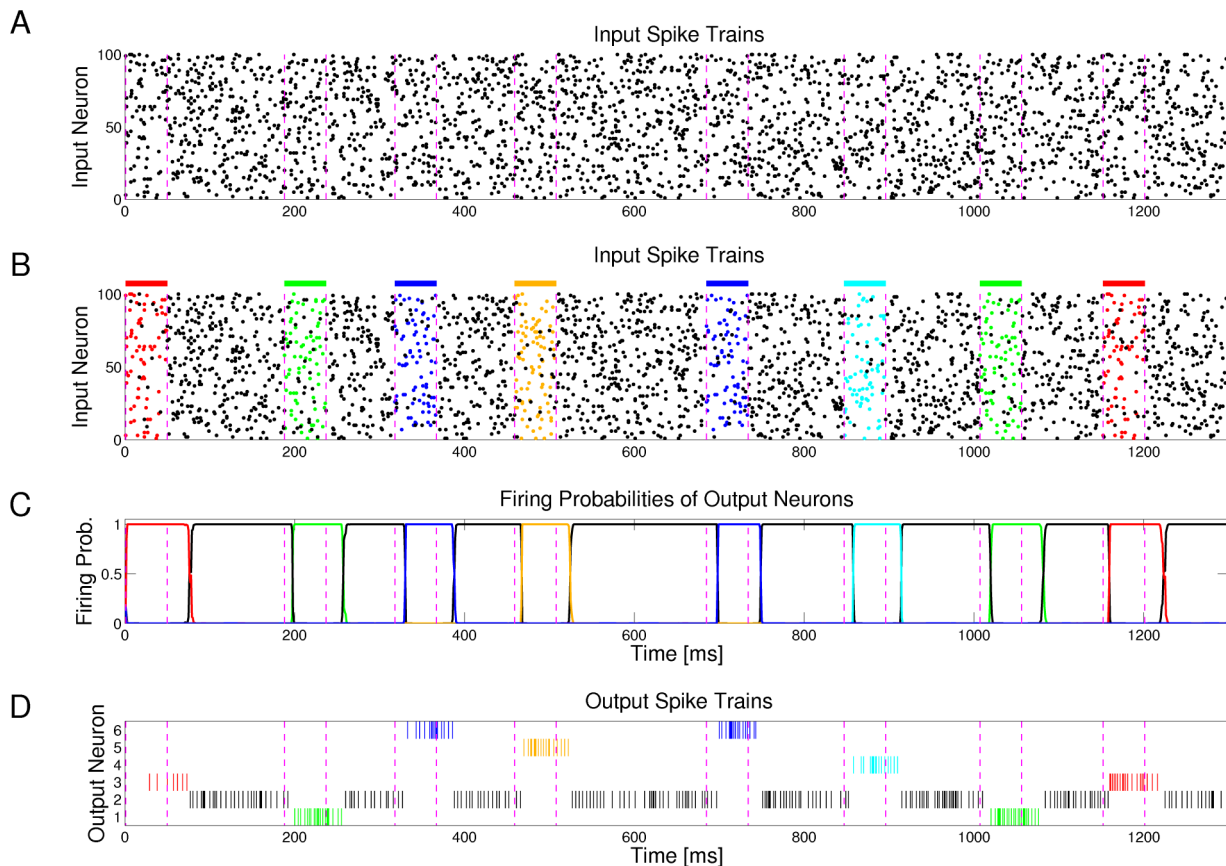


Figure 7. Output neurons self-organize via STDP to detect and represent spatio-temporal spike patterns. **A:** Sample of the Poisson input spike trains at 20 Hz (only 100 of the 500 input channels are shown). Dashed vertical lines mark time segments of 50 ms length where spatio-temporal spike patterns are embedded into noise. **B:** Same spike input as in A, but spikes belonging to five repeating spatio-temporal patterns (frozen Poisson spike patterns at 15 Hz) are marked in five different colors. These spike patterns are superimposed by noise (Poisson spike trains at 20 Hz) for intervals of randomly varying time lengths, and interrupted by segments of pure noise of the same statistics (Poisson spike trains at 20 Hz) for intervals of randomly varying time lengths. **C, D:** Firing probabilities and spike outputs of 6 output neurons (z-neurons in Fig. 1A) for the spike input shown in A, after applying STDP for 200 s to continuous spike trains of the same structure (without any supervision or reward). These 6 output neurons have self-organized so that 5 of them specialize on one of the 5 spatio-temporal patterns. One of the 6 output neurons (firing probability and spikes marked in black) only responds to the noise between these patterns. The spike trains in A represent test inputs, that had never been shown during learning. doi:10.1371/journal.pcbi.1003037.g007

Such emergent compression of high-dimensional spike inputs into sparse low-dimensional spike outputs could be used to merge information from multiple sensory modalities, as well as from internal sources (memory, predictions, expectations, etc.), and to report the co-occurrence of salient events to multiple other brain areas. This operation would be useful from the computational perspective no matter in which cortical area it is carried out. Furthermore, the computational modules that we have analyzed can easily be connected to form networks of such modules, since their outputs are encoded in the same way as their inputs: through probabilistic spiking populations that encode for abstract multinomial variables. Hence the principles for the emergence of Bayesian computation in local microcircuits that we have exhibited could potentially also explain the self-organization of distributed computations in large networks of such microcircuits.

Discussion

We have shown that STDP induces a powerful unsupervised learning principle in networks of spiking neurons with lateral inhibition: spike-based Expectation Maximization. Each application of STDP can be seen as a move in the direction of the M-step in a stochastic online EM algorithm that strives to maximize the log-likelihood $\log p(\mathbf{y}|\mathbf{w})$ of the spike input \mathbf{y} . This is equivalent to the minimization of the Kullback-Leibler divergence between the true distribution $p^*(\mathbf{y})$ of spike inputs, and the generative model $p(\mathbf{y}|\mathbf{w})$ that is implicitly represented by the WTA circuit from the Bayesian perspective. This theoretically founded principle guarantees that iterative applications of STDP to different spike inputs do not induce a meaningless meandering of the synaptic weights \mathbf{w} through weight space, but rather convergence to at least a local optimum in the fitting of the model to the distribution $p^*(\mathbf{y})$ of high-dimensional spike inputs \mathbf{y} . This generation of an internal model through STDP provides the primary component for the self-organization of Bayesian computation. We have shown that the other component, the prior, results from a simple rule for use-dependent adaptation of neuronal excitability. As a consequence, the firing of a neuron z_k in a stochastic WTA circuit (Fig. 1A) can be viewed as sampling from the posterior distribution of hidden causes for high-dimensional spike inputs \mathbf{y} (and simultaneously as the E-step in the context of online EM): A prior (encoded by the thresholds w_{k0} of the neurons z_k) is multiplied with a likelihood (encoded through an implicit generative distribution defined by the weights w_{k1}, \dots, w_{kn} of these neurons z_k), to yield through the firing probabilities of the neurons z_k a representation of the posterior distribution of hidden causes for the current spike input \mathbf{y} . The multiplications and the divisive normalization that are necessary for this model are carried out by the linear neurons in the log-scale. This result is then transformed into an instantaneous firing rate, assuming an exponential relationship between rate and the membrane potential [38]. It is important that the neurons z_k fire stochastically, i.e., that there exists substantial trial-to-trial variability, since otherwise they could not represent a probability distribution. Altogether our models support the view that probability distributions, rather than deterministic neural codes, are the primary units of information in the brain, and that computational operations are carried out on probabilities, rather than on deterministic bits of information.

Following the “probabilistic turn” in cognitive science [3,4,69] and related hypotheses in computational neuroscience [1,2,5], probabilistic inference has become very successful in explaining behavioral data on human reasoning and other brain functions. Yet, it has remained an important open problem how networks of spiking neurons can learn to implement those probabilistic

inference operations and probabilistic data structures. The soft WTA model presented in this article provides an answer for the case of Bayesian inference and learning in a simple graphical model, where a single hidden cause has to be inferred from bottom-up input. Although this is not yet a mechanism for learning to perform general Bayesian inference in arbitrary graphical models, it clearly is a first step into that direction. Importantly, the encoding of posterior distributions through spiking activity of the neurons z_k in a WTA circuit is perfectly compatible with the assumed input encoding from external variables x_j into spiking activity in \mathbf{y} . Thus, the interpretation of spikes from output neurons z_k as samples of the posterior distributions over hidden variables in principle allows for using these spikes as input for performing further probabilistic inference.

This compatibility of input and output codes means that SEM modules could potentially be hierarchically and/or recurrently coupled in order to serve as inputs of one another, although it remains to be shown how this coupling affects the dynamics of learning and inference. Future research will therefore address the important questions whether interconnected networks of modules for Bayesian computation that emerge through STDP can provide the primitive building blocks for probabilistic models of cortical computation. Previous studies [23,25] have shown that interconnected networks of WTA modules are indeed computationally very powerful. In particular, [24,25] have recently shown how recurrently connected neurons can be designed to perform neural sampling, an approach in which time-independent probability distributions can be represented through spiking activity in recurrent neural networks. The question how salient random variables come to be represented by the firing activity of neurons has remained open. This paper shows that such representations may emerge autonomously through STDP.

A prediction for networks of hierarchically coupled SEM modules would be that more and more abstract hidden causes can be learned in higher layers such as it has been demonstrated in machine learning approaches using Deep Belief Networks [66] and more recently in Deep Boltzmann Machines (DBM) [70]. This effect would correspond to the emergence of abstract feature selectivity in higher visual areas of primates (e.g. face-selective cells in IT, [71]). The hierarchical structure, however, that would result from such deeply organized SEM-modules is more reminiscent of a Deep Sum-Product Network [72], a recently presented new architecture, which has a much simpler learning dynamics but arguably a similar expressive power as DBM. In addition, with a consistent input encoding, associations between different sensory modalities could be formed by connecting inputs from different low-level or high-level sources to a single SEM.

Importantly, while the discussion above focused only on the representation of complex stimuli by neurons encoding abstract hidden causes, SEM can also be an important mechanism for fast and reliable reinforcement learning or decision making under uncertainty. Preprocessing via single or multiple SEM circuits provides an abstraction of the state of the organism, which is much lower-dimensional than the complete stream of individual sensory signals. Learning a behavioral strategy by reading out such behaviorally relevant high-level state signals and mapping them into actions could therefore speed up learning by reducing the state space. In previous studies [21,73] we have shown how optimal strategies can be learned very fast by simple local learning rules for reinforcement learning or categorization, if a preprocessing of input signals based on probabilistic dependencies is performed. SEM would be a suitable unsupervised mechanism for learning such preprocessing networks for decision making.

We also have shown that SEM is a very powerful principle that endows networks of spiking neurons to solve complex tasks of practical relevance (see e.g. Fig. 6), and as we have shown, their unsupervised learning performance is within the range of conventional machine learning approaches. Furthermore, this could be demonstrated for computations on spike inputs with an input dimension of about 1000 presynaptic neurons y_1, \dots, y_n , a number that approaches the typical dimension of the spike input that a cortical neuron receives. A very satisfactory aspect is that this high computational performance can be achieved by networks of spiking neurons that learn completely autonomously by STDP, without any postulated teacher or other guidance. This could benefit the field of neuromorphic engineering [74–76], which develops dedicated massively parallel and very efficient hardware for emulating spiking neural networks and suitable plasticity rules. The link between spiking neuron models and plasticity rules and established machine learning concepts provides a novel way of installing well-understood Bayesian inference and learning mechanisms on neuromorphic hardware. First steps towards implementing SEM-like rules in different types of neuromorphic hardware have been taken.

Prior related work

A first model for competitive Hebbian learning paradigm in non-spiking networks of neurons had been introduced in [77]. They analyzed a Hebbian learning rule in a hard WTA network and showed that there may exist equilibrium states, in which the average change of all weight values vanishes for a given set of input patterns. They showed that in these cases the weights adopt values that are proportional to the conditional probability of the presynaptic neuron being active given that the postsynaptic unit wins (rather than the log of this conditional probability, as in our framework). [78] showed that the use of a soft competition instead of a hard winner assignment and corresponding average weight updates lead to an exact gradient ascent on the log-likelihood function of a generative model of a mixture of Gaussians. However, these learning rules had not yet been analyzed in the context of EM.

Stochastic approximation algorithms for expectation maximization [31] were first considered in [79], incremental and on-line EM algorithms with soft-max competition in [80–82]. A proof of the stochastic approximation convergence for on-line EM in exponential family models with hidden variables was shown in [29]. They developed a sophisticated schedule for the learning rate in this much more general model, but did not yet consider individual learning rates for different weights.

[54] initiated the investigation of STDP in the context of unsupervised competitive Hebbian learning and demonstrated that correlations of input spike trains can be learned in this way. They also showed that this leads to a competition between the synapses for the control of the timing of the postsynaptic action potential. A similar competition can also be observed during learning in our model, since our learning rule automatically drives the weights towards satisfying the normalization conditions in Eq. (10).

[83] present a network and learning model that is designed to perform Independent Component Analysis (ICA) with spiking neurons through STDP and intrinsic plasticity. The mixture model of independent components can also be formulated as a generative model, and the goal of ICA is to find the optimal parameters of the mixing matrix. It has been shown that also this problem can be solved by a variant of Expectation Maximization [84], so there is some similarity to the identification of hidden causes in our model.

Recently, computer experiments in [85,86] have used STDP in the context of WTA circuits to achieve a clustering of input patterns. Their STDP rules implements linear updates, independent of the current weight values, mixed with a homeostasis rule to keep the sum of all weights constant and every weight between 0 and 1. This leads to weights that are roughly proportional to the probability of the presynaptic neuron's firing given that the postsynaptic neuron fires afterwards. The competition between the output neurons is carried out as hard-max. In [85] the 4 output neurons learn to differentiate the 4 presented patterns and smoothly interpolate new rotated input patterns, whereas in [86] 48 neurons learn to differentiate characters in a small pixel raster. [86] uses a STDP rule where both LTP and LTD are modeled as exponentially dependent on the time difference. However, the very specific experimental setting with synchronous regular firing of the input neurons makes it difficult to generalize their result to more general input spike trains. No theoretical analysis is provided in [85] or [86], but their experimental results can be explained by our SEM approach. Instead of adding up logs of conditional probabilities and performing the competition on the exponential of the sums, they sum up the conditional probabilities directly and use this sum of probabilities for the competition. This can be seen as a linear approximation of SEM, especially under the additional normalization conditions that they impose by homeostasis rules.

It has previously been shown that spike patterns embedded in noise can be detected by STDP [49,50,59]. Competitive pattern learning through STDP has recently been studied in [68]. They simulate a deterministic version of a winner-take-all circuit consisting of a fixed number of neurons, all listening to the same spiking input lines and connected to each other with a strong inhibition. The STDP learning rule that they propose is additive and weight-independent. Just like our results, they also observe that different neurons specialize on different fixed repeated input pattern, even though the repeated patterns are embedded in spiking noise such that the mean activity of all inputs remains the same throughout the learning phase. Additionally they show that within each pattern the responsible neuron tries to detect the start of the pattern. In contrast to our approach they do not give any analysis of convergence guarantees, nor does their model try to build a generative probabilistic model of the input distribution.

[87–89] investigated the possibility to carry out Bayesian probabilistic computations in recurrent networks of spiking neurons, both using probabilistic population codes. They showed that the ongoing dynamics of belief propagation in temporal Bayesian models can be represented and inferred by such networks, but they do not exhibit any neuronal plausible learning mechanism. [7] presented another approach to Bayesian inference using probabilistic population codes, also without any learning result.

An interesting complementary approach is presented in [90,91], where a single neuron is modeled as hidden Markov model with two possible states. This approach has the advantage, that the instantaneous synaptic input does not immediately decide the output state, but only incrementally influences the probability for switching the state. The weights and the temporal behavior can be learned online using local statistics. The downside of this approach is that this hidden Markov model can have only two states. In contrast, the SEM approach can be applied to networks with any number of output neurons.

In [92] it was shown that a suitable rule for supervised spike-based learning (the Tempotron learning rule) can be used to train a network to recognize spatio-temporal spike patterns. This discriminative learning scheme enables the recognizing neuron to focus on the most discriminative segment of the pattern. In

contrast, our generative unsupervised learning scheme drives the recognizing neuron to generalize and spike many times during the whole pattern, and thus learns the spatial average activity pattern. The conductance based approach of [92] differs drastically from our method (and the results shown in the Supplement) insofar as here only STDP was used (focusing on average spatial patterns), no supervision was involved, and the time-warped input pattern had never been shown during training.

An alternative approach to implement the learning of generative probabilistic models in spiking neuronal networks is given in [93,94]. Both approaches are based on the idea to model a sequence of spikes in a Hidden-Markov-Model-like probabilistic model and learn the model parameters through different variants of EM, in which a sequence of spikes represents one single sample of the model's distribution. Due to the explicit incorporation of inference over time, these models are more powerful than ours and thus require non-trivial, non-local learning mechanisms.

Experimentally testable predictions of the proposed model

Our analysis has shown, that STDP supports the creation of internal models and implements spike-based EM if changes of synaptic weights depend in a particular way on the current value of the weight: Weight potentiation depends in an inversely exponential manner on the current weight (see Eq. (5)). This rule for weight potentiation (see Fig. 8A) is consistent with all published data on this dependence: Fig. 5 in [43] and Fig. 5C in [40] for STDP, as well as Fig. 10 in [95] and Fig. 1 in [96] for other protocols for LTP induction. One needs to say, however, that these data exhibit a large trial-to-trial variability, so that it is hard to infer precise quantitative laws from them. On the other hand, the applications of STDP that we have examined in Fig. 3–7 work almost equally well if the actual weight increase varies by up to 100% from the weight increase proposed by our STDP rule (see open circles in Fig. 8A). The resulting distribution of weight increases matches qualitatively the above mentioned experimental data quite well.

The prediction of our model for the dependence of the amount of weight depression on the current weight is drastically different: Even though we make the strong simplification that the depression part of the STDP rule is independent of the time difference between pre- and postsynaptic spike, the formulation in Eq. (5) makes the assumption, that the amount of the depression should

be independent of the current weight value. It is this contrast between an exponential dependency for LTP and a constant LTD which makes the weight converge to the logarithm of the conditional presynaptic firing probability in Eq. (6). In experiments this dependency has been investigated in-vitro [40]. There it has been found that the *percentage* of weight depression under STDP is independent of the current weight, which implies that the amount of depression is linear in the current weight value. This seems to contradict the presented learning rule. However, the key property that is needed for the desired equilibrium condition is the ratio between LTP and LTD. So the equilibrium proof in Eq. (28) remains unchanged if Δw_{ki} is multiplied (for potentiation and depression) by some arbitrary function $f(w_{ki})$ of the current weight value. Choosing for example $f(w_{ki}) = w_{ki}$ yields a depression whose percentage is independent of the initial value, which would be consistent with the above mentioned in-vitro data [40]. The resulting dependence for potentiation is plotted in Fig. 8B. Since this curve is very similar to that of Fig. 8A, the above mentioned experimental data for potentiation are too noisy to provide a clear vote for one of these two curves. Thus more experimental data are needed for determining the dependence of weight potentiation on the initial weight. Whereas the relevance of this dependency had previously not been noted, our analysis suggests that such a contrast it is in fact essential for the capability of STDP to create internal models for high-dimensional spike inputs.

Our analysis has shown, that if the excitability of neurons is also adaptive, with a rule as in Eq. (7) that is somewhat analogous to that for synaptic plasticity, then neurons can also learn appropriate priors for Bayesian computation. Several experimental studies have already confirmed, that the intrinsic excitability of neurons does in fact increase when they are more frequently activated [45], see [97], [14] and [39] for reviews. But a quantitative study, which relates the resulting change in intrinsic excitability to its initial value, is missing.

Our model proposes that pyramidal neurons in cortical microcircuits are organized into stochastic WTA circuits, that together represent a probability distribution. This organization is achieved by a suitably regulated common inhibitory signal, where the inhibition follows the excitation very closely. Such instantaneous balance between excitation and inhibition was described by [34]. A resulting prediction of the WTA structure is that the firing activity of these neurons is highly de-correlated due to the inhibitory competition. In contrast to previous experimental

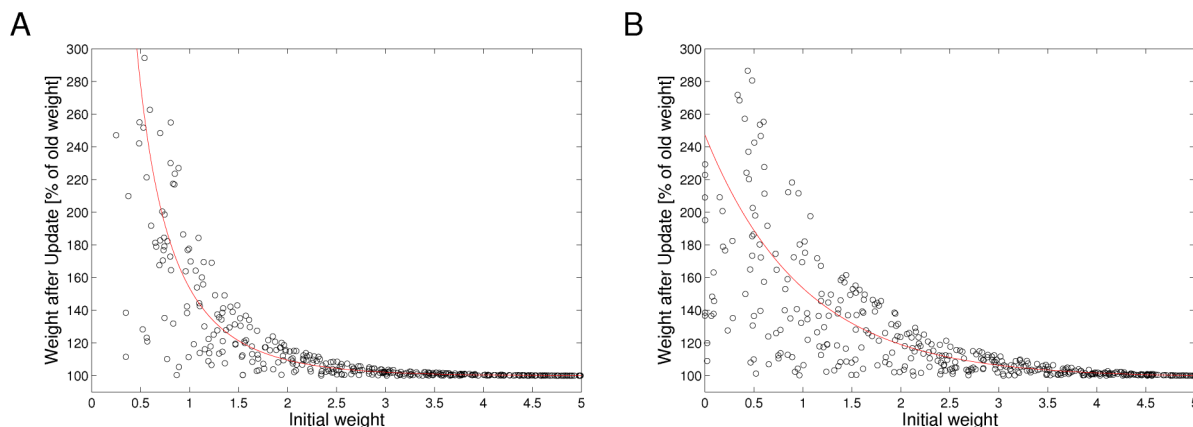


Figure 8. Ideal dependence of weight potentiation under STDP on the initial value of the weight (solid lines). Open circles represent results of samples from this ideal curve with 100% noise, that can be used in the previously discussed computer experiments with almost no loss in performance. **A:** Dependence of weight potentiation on initial weight according to the STDP rule in Eq. (5). **B:** Same with an additional factor w . doi:10.1371/journal.pcbi.1003037.g008

results, that reported higher correlations, it has recently been confirmed in [35] for the visual cortex of awake monkey that nearby neurons, even though they share common input show extremely low correlations.

Another prediction is that neural firing activity especially for awake animals subject to natural stimuli is quite sparse, since only those neurons fire whose internal model matches their spike input. A number of experimental studies confirm this predictions (see [98] for a review). Our model also predicts, that the neural firing response to stimuli exhibits a fairly high trial-to-trial variability, as is typical for drawing repeated samples from a posterior distribution (unless the posterior probability is close to 0 or 1). A fairly high trial-to-trial variability is a common feature of most recordings of neuronal responses (see e.g. [99], Fig. 1B in [100]; a review is provided in [101]). In addition, our model predicts that this trial-to-trial variability decreases for repeatedly occurring natural stimuli (especially if this occurs during attention) and discrimination capability improves for these stimuli, since the internal models of neurons are becoming better fitted to their spike input during these repetitions (“sharpening of tuning”), yielding posterior probabilities closer to 1 or 0 for these stimuli. These predictions are consistent with a number of experimental data related to perceptual learning [102,103], and with the evolution of neuronal responses to natural scenes that were shown repeatedly in conjunction with nucleus basalis stimulation [104].

In addition our model predicts that if the distribution of sensory inputs changes, the organization of codes for such sensory inputs also changes. More frequently occurring sensory stimuli will be encoded with a finer resolution (see [105] for a review of related experimental data). Furthermore in the case of sensory deprivation (see [106]) our model predicts that neurons that used to encode stimuli which no longer occur will start to participate in the encoding of other stimuli.

We have shown in Fig. 3 that an underlying background oscillation on neurons that provide input to a WTA circuit speeds up the learning process, and produces more precise responses after learning. This result predicts that cortical areas that collaborate on a common computational task, especially under attention, exhibit some coherence in their LFP. This has already been shown for neurons in close proximity [107] but also for neurons in different cortical areas [108,109].

If one views the modules for Bayesian computation that we have analyzed in this article as building blocks for larger cortical networks, these networks exhibit a fundamental difference to networks of neurons: Whereas a neuron needs a sufficiently strong excitatory drive in order to reach its firing threshold, the output neurons z of a stochastic WTA circuit according to our model in Eq. (3) are firing already on their own - even without any excitatory drive from the input neuron y (due to assumed background synaptic inputs; modeled in our simulations by an Ornstein-Uhlenbeck process, as suggested by in-vivo data [110]). Rather, the role of the input from the y -neurons is to modulate which of the neurons in the WTA circuit fire. One consequence of this characteristic feature is that even relatively few presynaptic neurons y can have a strong impact on the firing of the z -neurons, provided the z -neurons have learned (via STDP) that these y -neurons provide salient information about the hidden cause for the total input \mathbf{y} from all presynaptic neurons. This consequence is consistent with the surprisingly weak input from the LGN to area V1 [16,111,112]. It is also consistent with the recently found exponential distance rule for the connection strength between cortical areas [112]. This rule implies that the connection strength between distal cortical areas, say between primary visual cortex and PFC, is surprisingly weak. Our model suggests that these weak connections can nevertheless support

coherent brain computation and memory traces that are spread out over many, also distal, cortical areas.

Apart from these predictions regarding aspects of brain computation on the microscale and macroscale, a primary prediction of our model is that complex computations in cortical networks of neurons - including very efficient and near optimal processing of uncertain information - are established and maintained through STDP, on the basis of genetically encoded stereotypical connection patterns (WTA circuits) in cortical microcircuits.

Methods

According to our input model, every external multinomial variable x_j , with $j = 1, \dots, m$ is encoded through a group G_j of neurons y_i , with $i \in G_j$. The generative model $p(\mathbf{x}|\theta)$ from Fig. 1B is implicitly encoded in the WTA circuit of Fig. 1A with K excitatory neurons z_k by:

$$p(\mathbf{y}|\mathbf{w}) = \frac{1}{Z} \cdot \sum_{k=1}^K \left[e^{w_{k0}} \cdot \prod_{j=1}^m \prod_{i \in G_j} e^{w_{ki} \cdot \{x_j = v(i)\}} \right] \quad (19)$$

$$= \frac{1}{Z} \sum_{k=1}^K e^{w_{k0} + \sum_{i=1}^m w_{ki} \cdot y_i}$$

where $\{x_j = v(i)\}$ is the binary indicator function of x_j taking on value $v(i)$. In the generative model $p(\mathbf{y}|\mathbf{w})$ we define the binary variables y_i and set $y_i = 1$ if y_i represents the value $v(i)$ of the multinomial variable x_j (with j s.t. $i \in G_j$) and $x_j = v(i)$, otherwise $y_i = 0$. The sets G_j represent a partition of $\{1, \dots, n\}$, thus $\prod_{j=1}^m \prod_{i \in G_j} e^{w_{ki} y_i}$ and the form $\prod_{i=1}^n e^{w_{ki} y_i}$ used in Eq. (9) are equivalent expressions. The value of the normalization constant Z can be calculated explicitly as

$$Z = \sum_{k=1}^K e^{w_{k0}} \prod_{j=1}^m Z_{kj}, \quad \text{with} \quad Z_{kj} = \sum_{i \in G_j} e^{w_{ki}}. \quad (20)$$

This generative model can be rewritten as a mixture distribution with parameters π_k and μ_{ki} :

$$p(\mathbf{y}|\mathbf{w}) = \sum_{k=1}^K p(\mathbf{y}, k|\mathbf{w}) = \sum_{k=1}^K \left[\pi_k \cdot \prod_{i=1}^n \mu_{ki}^{y_i} \right], \quad (21)$$

$$\pi_k = p(k|\mathbf{w}) = e^{w_{k0}} \frac{\prod_{j=1}^m Z_{kj}}{Z} \quad (22)$$

$$\mu_{ki} = p(y_i = 1|k, \mathbf{w}) = \frac{e^{w_{ki}}}{Z_{kj}} \quad \text{with } i \in G_j. \quad (23)$$

In order to show how the constants Z_{kj} cancel out we write the full joint distribution of \mathbf{y} and the “hidden cause” k as the product of the prior $p(k|\mathbf{w})$ and the likelihood $p(\mathbf{y}|k, \mathbf{w})$:

$$p(\mathbf{y}, k|\mathbf{w}) = p(k|\mathbf{w}) \cdot p(\mathbf{y}|k, \mathbf{w}) \quad (24)$$

$$= e^{w_{k0}} \frac{\prod_{j=1}^m Z_{kj}}{Z} \cdot \prod_{i=1}^n \left(\frac{e^{w_{ki}}}{Z_{kj}} \right)^{y_i} \quad (25)$$

with j such that $i \in G_j$

$$= e^{w_{k0}} \frac{\prod_{j=1}^m Z_{kj}}{Z} \cdot \prod_{j=1}^m \left(\frac{1}{Z_{kj}} \prod_{i \in G_j} (e^{w_{ki}})^{y_i} \right) \quad (26)$$

$$= \frac{1}{Z} e^{w_{k0}} \cdot \prod_{i=1}^n e^{w_{ki} y_i}. \quad (27)$$

Under the normalization conditions in Eq. (10) the parameters of the mixture distribution simplify to $\mu_{ki} = e^{w_{ki}}$ and $\pi_k = e^{w_{k0}}$, since all $Z_{kj} = 1$ and $Z = 1$.

The generative model in Eq. (24) is well defined only for vectors \mathbf{y} , such that there is exactly one “1” entry per group G_j . However, in the network model with rectangular, renewable EPSPs, there are time intervals where $\mathbf{y}(t)$ may violate this condition, if the interval between two input spikes is longer than σ . It is obvious from Eq. (24) that this has the effect of dropping all factors representing x_j , since this results in an exponent of 0. Under proper normalization conditions (or at least if all Z_{kj} have identical values), this drop of an entire input group in the calculation of the posterior in Eq. (11) is identical to performing inference with unknown x_j (see ‘Impact of missing input values’). Eq. (11) holds as long as there are no two input spikes from different neurons within the same group closer than σ , which we have assumed for the simple input model with rectangular, renewable EPSPs.

Equilibrium condition

We will now show that all equilibria of the stochastic update rule in Eq. (5) and Eq. (7), i.e., all points where $E_{p^*}[\Delta \mathbf{w}] = \mathbf{0}$, exactly match the implicit solution conditions in Eq. (46), and vice versa:

$$\begin{aligned} E[\Delta w_{ki}] = \mathbf{0} &\Leftrightarrow p_{\mathbf{w}}^*(y_i = 1 | z_k = 1)(e^{-w_{ki}} - 1) - p_{\mathbf{w}}^*(y_i = 0 | z_k = 1) = 0 \\ &\Leftrightarrow p_{\mathbf{w}}^*(y_i = 1 | z_k = 1)(e^{-w_{ki}} - 1) + \\ &\quad p_{\mathbf{w}}^*(y_i = 1 | z_k = 1) - 1 = 0 \\ &\Leftrightarrow p_{\mathbf{w}}^*(y_i = 1 | z_k = 1)e^{-w_{ki}} = 1 \\ &\Leftrightarrow w_{ki} = \log p_{\mathbf{w}}^*(y_i = 1 | z_k = 1). \end{aligned} \quad (28)$$

Analogously, one can show that $E_{p^*}[\Delta w_{k0}] = 0 \Leftrightarrow w_{k0} = \log p_{\mathbf{w}}^*(z_k = 1)$. Note that this result implies that the learning rule in Eq. (5) and Eq. (7) has no equilibrium points outside the normalization conditions in Eq. (10), since all equilibrium points fulfill the implicit solutions condition in Eq. (46) and these in turn fulfill the normalization conditions.

Details to Learning the parameters of the probability model by EM

In this section we will analyze the theoretical basis for learning the parameters \mathbf{w} of the generative probability model $p(\mathbf{y}, k | \mathbf{w})$ given in Eq. (9) from a machine learning perspective. In contrast to the intuitive explanation of the Results section which was based on Expectation Maximization we will now derive an implicit analytical solution for a (locally) optimal weight vector \mathbf{w} , and rewrite this solution in terms of log probabilities. We will later use this derivation in order to show that the stochastic online learning rule provably converges towards this solution.

For an exact definition of the learning problem, we assume that the input is given by a stream of vectors \mathbf{y} , in which every \mathbf{y} is

drawn independently from the input distribution $p^*(\mathbf{y})$. In principle, this stream of \mathbf{y} 's corresponds to the samples $\mathbf{y}(t_1), \mathbf{y}(t_2), \dots$ that are observed at the spike times t_1, t_2, \dots of the circuit. However, in order to simplify the proofs in this and subsequent sections, we will neglect any possible temporal correlation between successive samples.

The learning task is to find parameter values \mathbf{w} , such that the marginal $p(\mathbf{y} | \mathbf{w})$ of the model distribution $p(\mathbf{y}, k | \mathbf{w})$ approximates the actual input distribution $p^*(\mathbf{y})$ as accurately as possible. This is equivalent to minimizing the Kullback-Leibler divergence between the two distributions:

$$\begin{aligned} \text{KL}(p^*(\mathbf{y}) || p(\mathbf{y} | \mathbf{w})) &= \sum_{\mathbf{y}} p^*(\mathbf{y}) \log \frac{p^*(\mathbf{y})}{p(\mathbf{y} | \mathbf{w})} \\ &= -H_{p^*}(\mathbf{y}) - E_{p^*}[\log p(\mathbf{y} | \mathbf{w})], \end{aligned} \quad (29)$$

where $H_{p^*}(\mathbf{y})$ is the (constant) entropy of the input distribution $p^*(\mathbf{y})$, and $E_{p^*}[\cdot]$ denotes the expectation over \mathbf{y} , according to the distribution $p^*(\mathbf{y})$. Since $H_{p^*}(\mathbf{y})$ is constant, minimizing the right hand side of Eq. (29) is equivalent to maximizing the expected log likelihood $L(\mathbf{w}) = E_{p^*}[\log p(\mathbf{y} | \mathbf{w})]$.

There are many different parametrizations \mathbf{w} that define identical generative distributions $p(\mathbf{y}, k | \mathbf{w})$ in Eq. (24). There is, however, exactly one \mathbf{w}' in this sub-manifold of the weight space that fulfills the normalization conditions in Eq. (10).

We thus redefine the goal of learning more precisely as the constrained maximization problem

$$\max_{\mathbf{w}} L(\mathbf{w}) \quad (30)$$

$$\text{subject to } \sum_{k=1}^K e^{w_{k0}} = 1 \quad \text{and} \quad \sum_{i \in G_j} e^{w_{ki}} = 1 \quad \text{for all } k, j. \quad (31)$$

This maximization problem never has a unique solution \mathbf{w} , because any permutation of the values of k and their corresponding weights leads to different joint distributions $p(\mathbf{y}, k | \mathbf{w})$, all of them having identical marginals $p(\mathbf{y} | \mathbf{w})$. The local maxima of Eq. (30) can be found using the Lagrange multiplier method.

Note that we do at no time enforce normalization of \mathbf{w} during the learning process, nor do we require normalized initialization of \mathbf{w} . Instead, we will show that the learning rule in Eq. (5,7) automatically drives \mathbf{w} towards a local maximum, in which the normalization conditions are fulfilled.

Under the constraints in Eq. (31) the normalization constant Z in Eq. (21) equals 1, thus $L(\mathbf{w})$ simplifies to $E_{p^*}[\log \sum_{k=1}^K e^{u_k}]$ - with $u_k = w_{k0} + \sum_{i=1}^n w_{ki} y_i$ - and we can define a Lagrangian function $\tilde{L}(\mathbf{w}, \boldsymbol{\lambda})$ for the maximization problem in Eq. (30,31) by

$$\begin{aligned} \tilde{L}(\mathbf{w}, \boldsymbol{\lambda}) &= E_{p^*}[\log \sum_{k=1}^K e^{u_k}] - \lambda_0 \left(1 - \sum_{k=1}^K e^{w_{k0}} \right) - \\ &\quad \sum_{k=1}^K \sum_{j=1}^m \lambda_{kj} \left(1 - \sum_{i \in G_j} e^{w_{ki}} \right). \end{aligned} \quad (32)$$

Setting the derivatives to zero we arrive at the following set of equations in \mathbf{w} and $\boldsymbol{\lambda}$:

$$\forall k : \quad \frac{\partial \tilde{L}}{\partial w_{k0}} = E_{p^*} \left[\frac{e^{u_k}}{\sum_{l=1}^K e^{u_l}} \right] - \lambda_0 e^{w_{k0}} = 0 \quad (33)$$

$$\forall k, i : \quad \frac{\partial \tilde{L}}{\partial w_{ki}} = E_{p^*} \left[y_i \frac{e^{u_k}}{\sum_{l=1}^K e^{u_l}} \right] - \lambda_{kj} e^{w_{ki}} = 0. \quad (34)$$

Summing over those equations that have the same multiplier λ_{kj} or λ_0 , resp., leads to

$$\sum_{k=1}^K \frac{\partial \tilde{L}}{\partial w_{k0}} = \sum_{k=1}^K E_{p^*} [p(k|\mathbf{y}, \mathbf{w})] - \lambda_0 \sum_{k=1}^K e^{w_{k0}} = 0 \quad (35)$$

$$\forall k, j : \sum_{i \in G_j} \frac{\partial \tilde{L}}{\partial w_{ki}} = \sum_{i \in G_j} E_{p^*} [y_i p(k|\mathbf{y}, \mathbf{w})] - \lambda_{kj} \sum_{i \in G_j} e^{w_{ki}} = 0, \quad (36)$$

where $p(k|\mathbf{y}, \mathbf{w})$ is the shorthand notation for the equivalent expression $\frac{e^{u_k}}{\sum_{l=1}^K e^{u_l}}$. The identity $\sum_{k=1}^K E_{p^*} [p(k|\mathbf{y}, \mathbf{w})] = 1$, the identity $\sum_{i \in G_j} E_{p^*} [y_i p(k|\mathbf{y}, \mathbf{w})] = E_{p^*} [p(k|\mathbf{y}, \mathbf{w}) \sum_{i \in G_j} y_i]$ the fact that $\sum_{i \in G_j} y_i = 1$, which follows from the definition of population encoding, and the constraints in Eq. (31) are used in order to derive the explicit solution for the Lagrange multipliers

$$\lambda_0 = 1 \quad \text{and} \quad \forall k, j : \lambda_{kj} = E_{p^*} [p(k|\mathbf{y}, \mathbf{w})], \quad (37)$$

in dependence of \mathbf{w} . We insert this solution for λ into the gradient Eq. (33,34) and get

$$E_{p^*} [p(k|\mathbf{y}, \mathbf{w})] - e^{w_{k0}} = 0 \quad (38)$$

$$E_{p^*} [y_i p(k|\mathbf{y}, \mathbf{w})] - E_{p^*} [p(k|\mathbf{y}, \mathbf{w})] e^{w_{ki}} = 0,$$

from which we derive an implicit solution for \mathbf{w} :

$$\begin{aligned} w_{k0} &= \log E_{p^*} [p(k|\mathbf{y}, \mathbf{w})] \\ w_{ki} &= \log \frac{E_{p^*} [y_i p(k|\mathbf{y}, \mathbf{w})]}{E_{p^*} [p(k|\mathbf{y}, \mathbf{w})]}. \end{aligned} \quad (39)$$

It is easily verified that all fixed points of this implicit solution satisfy the normalization constraints:

$$\sum_{k=1}^K e^{w_{k0}} = \sum_{k=1}^K E_{p^*} [p(k|\mathbf{y}, \mathbf{w})] = E_{p^*} \left[\sum_{k=1}^K p(k|\mathbf{y}, \mathbf{w}) \right] = 1 \quad (40)$$

$$\sum_{i \in G_j} e^{w_{ki}} = \sum_{i \in G_j} \frac{E_{p^*} [y_i p(k|\mathbf{y}, \mathbf{w})]}{E_{p^*} [p(k|\mathbf{y}, \mathbf{w})]} = \frac{E_{p^*} [p(k|\mathbf{y}, \mathbf{w}) \sum_{i \in G_j} y_i]}{E_{p^*} [p(k|\mathbf{y}, \mathbf{w})]} = 1. \quad (41)$$

Finally, in order to simplify the notation we use the augmented input distribution $p_{\mathbf{w}}^*(\mathbf{y}, \mathbf{z})$. The expectations in Eq. (39) nicely

evaluate to

$$\begin{aligned} E_{p^*} [p(z_k = 1|\mathbf{y}, \mathbf{w})] &= \sum_{\mathbf{y}} p^*(\mathbf{y}) p(z_k = 1|\mathbf{y}, \mathbf{w}) \\ &= \sum_{\mathbf{y}} p_{\mathbf{w}}^*(\mathbf{y}, z_k = 1) = \end{aligned} \quad (42)$$

$$= p_{\mathbf{w}}^*(z_k = 1) \quad \text{and} \quad (43)$$

$$\begin{aligned} E_{p^*} [y_i p(z_k = 1|\mathbf{y}, \mathbf{w})] &= \sum_{\mathbf{y}} p^*(\mathbf{y}) y_i p(z_k = 1|\mathbf{y}, \mathbf{w}) \\ &= \sum_{\mathbf{y}} y_i p_{\mathbf{w}}^*(\mathbf{y}, z_k = 1) = \end{aligned} \quad (44)$$

$$= p_{\mathbf{w}}^*(y_i = 1, z_k = 1), \quad (45)$$

which allows us to rewrite the implicit solution in a very intuitive form as:

$$w_{k0} = \log p_{\mathbf{w}}^*(z_k = 1) \quad w_{ki} = \log p_{\mathbf{w}}^*(y_i = 1|z_k = 1). \quad (46)$$

Any weight vector \mathbf{w} that fulfills Eq. (46) is either a (local) maximum, a saddle point or a (local) minimum of the log likelihood function L under the normalization constraints.

An obvious numerical approach to solve this fixed point equation is the repeated application of Eq. (39). According to the derivations in the Results section this corresponds exactly to the Expectation Maximization algorithm. But every single iteration asks for the evaluation of expectations with respect to the input distribution $p^*(\mathbf{y})$, which theoretically requires infinite time in an online learning setup.

Details to Spike-based Expectation Maximization

We derive the update rule in Eq. (5) from the statistical perspective that each weight can be interpreted as $w_{ki} = \log \frac{a_{ki}}{N_{ki}}$, where a_{ki} and N_{ki} correspond to counters of the events $\langle y_i = 1, z_k = 1 \rangle$ and $\langle z_k = 1 \rangle$. Every new event $\langle y_i, z_k \rangle$ leads to a weight update

$$w_{ki}^{new} = \log \frac{a_{ki} + y_i z_k}{N_{ki} + z_k} = \quad (47)$$

$$= \log \frac{a_{ki}}{N_{ki}} \left(1 + \frac{1}{N_{ki}} \frac{N_{ki}}{a_i} y_i z_k \right) \left(1 + \frac{1}{N_{ki}} z_k \right)^{-1} \quad (48)$$

$$\begin{aligned} &= w_{ki} + \log \left(1 + \frac{1}{N_{ki}} e^{-w_{ki}} y_i z_k \right) \\ &\quad - \log \left(1 + \frac{1}{N_{ki}} z_k \right) \end{aligned} \quad (49)$$

$$\approx w_{ki} + \frac{1}{N_{ki}} z_k (e^{-w_{ki}} y_i - 1), \quad (50)$$

where the log-function is linearly approximated around 1 as $\log(1+x) \approx x$. The factor $\frac{1}{N_{ki}}$ is understood as learning rate η_{ki} in the additive update rule $w_{ki}^{new} = w_{ki} + \eta_{ki} \Delta w_{ki}$. If $z_k = 0$, i.e. if there

is no postsynaptic spike, the update $\Delta w_{ki}=0$. In the case of a postsynaptic spike, i.e., $z_k=1$, the update $\Delta w_{ki}=0$ decomposes in the two cases $y_i=1$ and $y_i=0$ as it is stated explicit in Eq. (5).

As a side note, we observe that by viewing our STDP rule as an approximation to counting statistics, the learning rate $\eta_{ki} = \frac{1}{N_{ki}}$ can be understood as the inverse of the equivalent sample size from which the statistics was gathered. If the above rule is used with a small constant learning rate we will get a close approximation to an exponentially decaying average. If the learning rate decays like $\frac{1}{N_{ki}}$ we will get an approximation to an online updated average, where all samples are equally weighted. We will come back to a regulation mechanism for the learning rate in the section ‘Variance Tracking’.

Details to Proof of convergence

In this section we give the proof of Theorem 1. Formally, we define the sequences $\mathbf{w}^{(t)}$, $\Delta \mathbf{w}^{(t)}$, $\mathbf{y}^{(t)}$, $\mathbf{z}^{(t)}$ and $\eta^{(t)}$ for $t=0,1,\dots,\infty$: For all t we assume that $\mathbf{y}^{(t)}$ is drawn independently from $p^*(\mathbf{y})$. The value of $\mathbf{z}^{(t)}$ is drawn from the posterior distribution of the model $p(\mathbf{z}|\mathbf{y}^{(t)}, \mathbf{w}^{(t)})$ (see Eq. (11)), given the input $\mathbf{y}^{(t)}$ and the current model parameters $\mathbf{w}^{(t)}$. The weight updates $\Delta w_{ki}^{(t)}$, and $\Delta w_{k0}^{(t)}$ are calculated according to Eq. (5) and (7) with $c=1$. The sequence of weight vectors $\mathbf{w}^{(t)}$ is determined by the randomly initialized vector $\mathbf{w}^{(0)}$, and by the iteration equation

$$\mathbf{w}^{(t+1)} = \Pi(\mathbf{w}^{(t)} + \eta^{(t)} \Delta \mathbf{w}^{(t)}). \quad (51)$$

The projection function Π represents a coordinate-wise clipping of $\mathbf{w}^{(t+1)}$ to a hyper-rectangle B such that

$$-w_{\min} \leq w_{ki}^{(t+1)} \leq 0 \quad \text{and} \quad -w_{\min} \leq w_{k0}^{(t+1)} \leq 0. \quad (52)$$

The bound w_{\min} is assumed to be chosen so that all (finite) maxima of L are inside of B . For the sequence of learning rates $\eta^{(t)}$ we assume that

$$\sum_{t=1}^{\infty} \eta^{(t)} = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} (\eta^{(t)})^2 < \infty. \quad (53)$$

Under these assumptions we can now restate the theorem formally:

Theorem 1: *The sequence $\mathbf{w}^{(t)}$ converges with probability 1 to the set S_B of all points within the hyper-rectangle B that fulfill the equilibrium conditions in Eq. (6). The stable convergence points among S_B are the (local) maxima of L , subject to the normalization constraints in Eq. (10).*

The iterative application of the learning rule in Eq. (5) and (7) is indeed a stochastic approximation algorithm for learning a (locally) optimal parameter vector \mathbf{w} . We resort to the theory of stochastic approximation algorithms as presented in [52] and use the method of the “mean limit” ordinary differential equation (ODE). The goal is to show that the sequence of the weight vector $\mathbf{w}^{(t)}$ under the stochastic learning rule in Eq. (5) and (7) converges to one of the local maxima of Eq. (30) with probability one, i.e., the probability to observe a non-converging realization of this sequence is zero. The location of the local maximum to which a single sequence of $\mathbf{w}^{(t)}$ converges depends on the starting point $\mathbf{w}^{(0)}$ as well as on the concrete realization of the stochastic noise sequence. We will not discuss the effect of this stochasticity in more

detail, except for stating that a stochastic approximation algorithm is usually less prone to get stuck in small local maxima than its deterministic version. The stochastic noise introduces perturbations that decrease slowly over time, which has an effect that is comparable to simulated annealing.

We will use the basic convergence theorem of [52] to establish the convergence of the sequence $\mathbf{w}^{(t)}$ to the limit set of the mean limit ODE. Then it remains to show that this limit set is identical to the desired set of all equilibrium points and thus, particularly, does not contain limit cycles.

Proof: In the notation of [52], the mean update of the stochastic algorithm in Eq. (51) is $\bar{\mathbf{g}}(\mathbf{w}) = \mathbb{E}_{p^*}[\Delta \mathbf{w}^{(t)}]$. The bounds B imply that $\mathbb{E}_{p^*}[\|\Delta \mathbf{w}^{(t)}\|] < \infty$ for all t and $\sup_t \mathbb{E}_{p^*}[(\Delta \mathbf{w}^{(t)})^2] < \infty$.

For any set A we define $F(A)$ as the positive limit set of the mean limit ODE $\dot{\mathbf{w}}(s) = \bar{\mathbf{g}}(\mathbf{w}(s))$ for all initial conditions $\mathbf{w}(0) \in A$:

$$F(A) = \lim_{s \rightarrow \infty} \bigcup_{\mathbf{w} \in A} \{\mathbf{w}(s'), s' \geq s : \mathbf{w}(0) = \mathbf{w}\}. \quad (54)$$

According to Theorem 3.1 in Chapter 5 of [52], the sequence $\mathbf{w}^{(t)}$ under the algorithm in Eq. (51) converges for all start conditions $\mathbf{w}^{(0)} \in B$ to the limit set $F(B)$ with probability one in the sense that

$$\lim_{t \rightarrow \infty} \min_{\mathbf{w} \in F(B)} |\mathbf{w}^{(t)} - \mathbf{w}| = 0. \quad (55)$$

We will now show that the limit set $F(B)$ of $\dot{\mathbf{w}} = \bar{\mathbf{g}}(\mathbf{w})$ is identical to the set of stationary points $S_B = \{\mathbf{w} \in B : \bar{\mathbf{g}}(\mathbf{w}) = 0\}$ and does not contain limit cycles. It is obvious that S_B is a subset of $F(B)$ since for all initial conditions $\mathbf{w}(0) \in S_B$ the trajectory of $\dot{\mathbf{w}}(s) = \bar{\mathbf{g}}(\mathbf{w}(s))$ fulfills $\mathbf{w}(s) \equiv \mathbf{w}(0)$ for all s . Thus it remains to be shown that there are no other points in F_B (like e.g. limit cycles).

We split the argument into two parts. In the first part we will show that for $s \rightarrow \infty$ all trajectories of $\dot{\mathbf{w}}(s) = \bar{\mathbf{g}}(\mathbf{w}(s))$ converge asymptotically to the manifold H defined by the normalization constraints 31. This leads to the conclusion that $F(B \setminus H) \subset F(B \cap H)$. In the second part we will show that all trajectories within H converge to the stationary points S_B , i.e., $F(B \cap H) = S_B$. Both parts together yield the desired result that S_B are the only limit points of the ODE $\dot{\mathbf{w}}(s) = \bar{\mathbf{g}}(\mathbf{w}(s))$.

The first part we start by defining the set of functions $h_0(\mathbf{w})$ and $h_{kj}(\mathbf{w})$ for all k, j to represent the deviation of the current \mathbf{w} from each of the normalization constraints 31, i.e.,

$$h_0(\mathbf{w}) = \sum_{k=1}^K e^{w_{k0}} - 1 \quad h_{kj}(\mathbf{w}) = \sum_{i \in G_j} e^{w_{ki}} - 1. \quad (56)$$

The manifold H is the set of all points \mathbf{w} where $h_0(\mathbf{w}) = 0$ and $h_{kj}(\mathbf{w}) = 0$ for all k, j . Furthermore, we calculate the gradient vectors $\frac{\partial h_0}{\partial \mathbf{w}}$ and $\frac{\partial h_{kj}}{\partial \mathbf{w}}$ for each of these functions with respect to the argument \mathbf{w} . Note that many entries of these gradient vectors are 0, since every single function $h_{kj}(\mathbf{w})$ and $h_0(\mathbf{w})$ only depends on a few entries of its argument \mathbf{w} . The nonzero entries of these gradients are

$$\frac{\partial h_0}{\partial w_{k0}} = e^{w_{k0}} \quad \forall i \in G_j : \frac{\partial h_{kj}}{\partial w_{ki}} = e^{w_{ki}}. \quad (57)$$

We can now show that the trajectory of $\dot{\mathbf{w}}(s) = \bar{\mathbf{g}}(\mathbf{w}(s))$ in any point $\mathbf{w}(s)$ always points in direction of decreasing absolute values for all deviations $h_0()$ and $h_{kj}()$:

$$\bar{\mathbf{g}}(\mathbf{w}) \cdot \frac{\partial h_0}{\partial \mathbf{w}} = \sum_{k=1}^K (\mathbb{E}_{p^*}[z_k] e^{-w_{k0}} - 1) e^{w_{k0}} = 1 - \sum_{k=1}^K e^{w_{k0}} \quad (58)$$

$$= -h_0(\mathbf{w}) \quad (59)$$

$$\begin{aligned} \bar{\mathbf{g}}(\mathbf{w}) \cdot \frac{\partial h_{kj}}{\partial \mathbf{w}} &= \sum_{i \in G_j} (\mathbb{E}_{p^*}[y_i z_k] e^{-w_{ki}} - \mathbb{E}_{p^*}[z_k]) e^{w_{ki}} \\ &= p(k|\mathbf{y}, \mathbf{w}) (1 - \sum_{i \in G_j} e^{w_{ki}}) \end{aligned} \quad (60)$$

$$= -p(k|\mathbf{y}, \mathbf{w}) h_{kj}(\mathbf{w}) \quad (61)$$

This shows that $\lim_{s \rightarrow \infty} h_{kj}(\mathbf{w}(s)) = 0$ for all k, j and $\lim_{s \rightarrow \infty} h_0(\mathbf{w}(s)) = 0$. This implies that the limit set of all trajectories with initial conditions outside H is contained in H , or more formally $F(B \setminus H) \subseteq B \cap H$. Note that the continuity and the boundedness of $\bar{\mathbf{g}}(\mathbf{w})$ on B implies $F(F(A)) = F(A)$ and $F(A_1) \subseteq F(A_2)$ if $A_1 \subseteq A_2$ for all $A, A_1, A_2 \subseteq B$. Therefore we can now conclude as the result of the first part

$$F(B \setminus H) = F(F(B \setminus H)) \subseteq F(B \cap H), \quad (62)$$

i.e. the limit set of all trajectories starting outside the manifold of normalized weights is contained in the limit set of all trajectories starting within the normalization constraints. The equations (61) also prove that any trajectory with initial condition $\mathbf{w}(0) \in H$ stays within H , since all components of $\bar{\mathbf{g}}(\mathbf{w}(s))$ with directions orthogonal to the tangent space of H in $\mathbf{w}(s)$ are 0 for all s , thus $\bar{\mathbf{g}}(\mathbf{w}(s))$ is in the tangent space H in $\mathbf{w}(s)$.

This immediately leads to the second part of the proof, which is based on the gradient $\frac{\partial \tilde{L}}{\partial \mathbf{w}}$ of the Lagrangian \tilde{L} as given in Eq. (33, 34). For any $\mathbf{w} \in H$ let $P(\mathbf{w})$ be the linear projection matrix that orthogonally projects any vector \mathbf{a} into the tangent space of H in \mathbf{w} . The projection $P(\mathbf{w}) \cdot \frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}}$ of the gradient of \tilde{L} at any $\mathbf{w} \in H$ points towards the strongest increase of the value of the objective function L under the constraints of the normalization conditions. Thus, the value of L increases in the direction of any vector within the tangent space of H in \mathbf{w} that has a positive scalar product with $P(\mathbf{w}) \cdot \frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}}$. As $\bar{\mathbf{g}}(\mathbf{w})$ is a tangent vector of H in \mathbf{w} for all $\mathbf{w} \in H$, the orthogonal component $\frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}} - P(\mathbf{w}) \cdot \frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}}$ of the gradient is orthogonal to $\bar{\mathbf{g}}(\mathbf{w})$. Thus, the value of the scalar product with the projected gradient $\bar{\mathbf{g}}(\mathbf{w}) \cdot (P(\mathbf{w}) \cdot \frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}})$ is identical to the value of the scalar product with the gradient itself $\bar{\mathbf{g}}(\mathbf{w}) \cdot \frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}}$:

$$\begin{aligned} \bar{\mathbf{g}}(\mathbf{w}) \cdot \frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}} &= \sum_{k=1}^K \frac{\partial \tilde{L}}{\partial w_{k0}} (\mathbb{E}_{p^*}[z_k] e^{-w_{k0}} - 1) \\ &\quad + \sum_{i,k} \frac{\partial \tilde{L}}{\partial w_{ki}} (\mathbb{E}_{p^*}[z_k y_i] e^{-w_{ki}} - \mathbb{E}_{p^*}[z_k]) \end{aligned} \quad (63)$$

$$\begin{aligned} &= \sum_{k=1}^K e^{-w_{k0}} (\mathbb{E}_{p^*}[p(z=k|\mathbf{y}, \mathbf{w}) - e^{w_{k0}}])^2 + \\ &\quad + \sum_{i,k} e^{-w_{ki}} (\mathbb{E}_{p^*}[(y_i - e^{w_{ki}}) p(z=k|\mathbf{y}, \mathbf{w})])^2 \geq 0, \end{aligned}$$

with equality if and only if $\bar{\mathbf{g}}(\mathbf{w}) = 0$, which is equivalent to $\frac{\partial \tilde{L}(\mathbf{w})}{\partial \mathbf{w}} = 0$. This shows that all trajectories with initial condition $\mathbf{w}(0) \in H$ stay within H forever and converge to the set of stationary points S_H , i.e. $F(B \cap H) = S_H$. Combining the results of both parts as

$$F(B) = F(B \setminus H) \cup F(B \cap H) = F(B \cap H) = S_B \quad (64)$$

establishes the stochastic convergences of any sequence $\mathbf{w}^{(t)}$ to the set S_B with probability one.

Weight offsets and positive weights. All weights w_{ki} in the theoretical model are logs of probabilities and therefore always have negative values. Through a simple transformation we can shift all weights into the positive range in order to be able to use positive weights only, which is the common assumption for excitatory connections in biologically inspired neural network models. We will now show that setting the parameter c in Eq. (5) different from 1 leads to a linear shift of the resulting weight values by $\log c$, without changing the functionality of the Spike-based EM algorithm.

Firstly, we observe that the application of the update rule in Eq. (5) with $c > 1$ on a shifted weight $w'_{ki} = w_{ki} + \log c$ is identical to the application of the update rule with $c = 1$ on the original weight w_{ki} , since

$$ce^{\overbrace{w_{ki} + \log c}^{w'_{ki}}} - 1 = e^{-w_{ki}} - 1. \quad (65)$$

Secondly, we see that the relative firing rate $q_k(t)$ of neuron z_k remains unchanged if all weights are subject to the same offset $\log c$, since

$$q_k(t) = \frac{e^{w_{k0} + \sum_{i=1}^n (w_{ki} + \log c) y_i}}{\sum_{k'=1}^K e^{w_{k'0} + \sum_{i=1}^n (w_{k'i} + \log c) y_i}} \quad (66)$$

$$= \frac{\left(e^{\log c \sum_{i=1}^n y_i} \right) e^{w_{k0} + \sum_{i=1}^n w_{ki} y_i}}{\sum_{k'=1}^K \left(e^{\log c \sum_{i=1}^n y_i} \right) e^{w_{k'0} + \sum_{i=1}^n w_{k'i} y_i}} \quad (67)$$

$$= \frac{e^{w_{k0} + \sum_{i=1}^n w_{ki} y_i}}{\sum_{k'=1}^K e^{w_{k'0} + \sum_{i=1}^n w_{k'i} y_i}} \quad (68)$$

In contrast, the overall firing rate $R(t)$ increases by the factor $e^{\log c \sum_{i=1}^n y_i}$. By our definition of the population coding for \mathbf{y} , this factor equals $e^{m \log c}$, where m is the number of original input variables \mathbf{x} . An increase of the inhibitory signal $I(t)$ by $m \log c$ can therefore compensate the increase of overall firing rate. Using this

shifted representation, a single excitatory synapse can take on values in the range $[0, \log c]$, corresponding to probabilities in the range $[\frac{1}{c}, 1]$.

Similarly the consideration holds valid that it is mathematically equivalent whether the depression of the excitability w_{k0} in Eq. (7) is modeled either as an effect of lateral spiking activity or as a constant decay, independent of the circuit activity. In the first case, w_{k0} converges to the relative spiking probability of the k -th neuron such that the sum of all w_{k0} is indeed 1 as described by our theory. In the second case, the w_{k0} really describe absolute firing rates in some time scale defined by the decay constant. In the logarithmic scale of w_{k0} this is nothing else than a constant offset and thus cancels down in Eq. (68).

Impact of missing input values. The proof of theorem 1 assumes that every sample $\mathbf{y}^{(t)}$ gathered online is a binary vector which contains exactly one entry with value 1 in every group G_j . This value indicates the value of the abstract variable x_j that is encoded by this group. As long as the spikes from the input neurons are closely enough in time, this condition will be fulfilled for every activation vector $\mathbf{y}(t)$. For the cases in which the value of the abstract variable x_j changes, the first spike from group G_j has to appear exactly at that point in time at which the rectangular EPSP for the previous value vanishes, i.e., σ ms after the last preceding spike.

We will now break up this strong restriction of the provable theory and analyze the results that are to be expected, if we allow for interspike intervals longer than σ . We interpret the resulting “gaps” in the information about the value of an input group as missing value in the sense of Bayesian inference.

We had already addressed the issue of such missing values, resulting from presynaptic neurons that do not spike within the integration time window of an output neuron z_k , in the discussion of Fig. 3.

A profound analysis of the correct handling of missing data in EM can be found in [48]. Their analysis implies that the correct learning action would be to leave all weights w_{ki} in the group G_j unchanged, if the value of the external variable x_j is missing, i.e., if all corresponding y_i 's are 0. However, in this case the STDP rule in Eq. (5) reduces these weights by η . This leads to a modification of the analysis of the equilibrium condition (28):

$$\begin{aligned} E[\Delta w_{ki}] &= 0 \Leftrightarrow (1-r) \\ & (p^*(y_i=1|z_k=1)\eta(e^{-w_{ki}}-1) - p^*(y_i=0|z_k=1)\eta) \\ & - r\eta = 0 \\ & \Leftrightarrow w_{ki} = \log p^*(y_i=1|z_k=1) + \log(1-r), \end{aligned} \quad (69)$$

where r is the probability that i belongs to a group G_j in which the value of x_j is unknown. We assume that the probability for such a missing value event is independent of the (true) value of the abstract variable x_j and we assume further that the probability of such missing value events is the same for all groups G_j and thus conclude that this offset of $\log(1-r)$ is expected to be the same for all weights. It can easily be verified, that such an offset does not change the resulting probabilities of the competition in the inference according to Eq. (68).

Adaptive learning rates with Variance Tracking. In our experiments we used an adaptation of the variance tracking heuristic from [73] for an adaptive control of learning rates. If we assume that the consecutive values of the weights represent independent samples of their true stochastic distribution at the current learning rate, then this observed distribution is the log of a beta-distribution defined by the parameters a_{ki} and N_{ki} that were

used in Eq. (50) to define the update of w_{ki} from sufficient statistics. Analytically (see supplement) this distribution has the first and second moments

$$E[w_{ki}] \approx \log \frac{a_{ki}}{N_{ki}} \quad \text{and} \quad E[w_{ki}^2] \approx E[w_{ki}]^2 + \frac{1}{a_{ki}} + \frac{1}{N_{ki}}. \quad (70)$$

From the first equation we estimate $\frac{1}{a_{ki}} = \frac{e^{-E[w_{ki}]}}{N_{ki}}$. This leads to a heuristic estimate for the (inverse of the) current sample size based on the empirically observed variance $E[w_{ki}^2] - E[w_{ki}]^2$:

$$\eta_{ki}^{new} = \frac{1}{N_{ki}} = \frac{E[w_{ki}^2] - E[w_{ki}]^2}{e^{-E[w_{ki}]} + 1}. \quad (71)$$

The empirical estimates of these first two moments can be gathered online by exponentially decaying averages using the same learning rate η_{ki} . Even though the assumption of independent samples for the estimates of the moments is not met, one can argue about two cases: In case of a stationary evolution of the weight, the strong dependence of consecutive samples typically leads to an underestimation of the variance. This in turn leads to a decrease of the learning rate which is the desired effect of a stationary evolution. In case of a directed evolution of the weight the variance will at least indicate the amount of the current gradient of the evolution despite the strong dependence and thus keep the learning rate high enough to support fast convergence towards the asymptote of the gradient.

An adaptive learning rate such as in Eq. (71) facilitates a spontaneous reorganization of the internal models encoded by the weight vectors of the output neurons z_k in case that the input distribution $p^*(\mathbf{y})$ changes (see Fig. S1 in Text S1).

Details to Role of the Inhibition

Biased sampling problem. In this section we analyze the influence of the instantaneous output firing rate $R(t)$ of the learning circuit and derive the analytical result that the output rate $R(t)$ plays the role of a multiplicative weighting of samples during learning. We show how a theoretically optimal inhibition signal can compensate this effect and describe how this compensation is approximated in our experiments.

We start with the assumption that the input signal $\mathbf{y}(t)$ can be described by some stationary stochastic process. An empirical estimate of its stationary distribution can be obtained by measuring the relative duration of presentation of every different discrete value \mathbf{y} in a time window of length T . The accuracy of this empirical estimate of the input distribution can be increased by using a longer time window T , such that in the limit of an infinitely large time window the estimate will converge to the true stationary input distribution of \mathbf{y} , denoted by $p^*(\mathbf{y})$:

$$p^*(\mathbf{y}) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \delta(\mathbf{y} - \mathbf{y}(t)) dt, \quad (72)$$

where δ is a vectorized version of the Kronecker Delta with $\delta(0)=1$ and $\delta(\mathbf{x})=0$, if $\mathbf{x} \neq 0$.

However, even though the WTA-circuit receives this time-continuous input stream $\mathbf{y}(t)$, the spike-triggered STDP rule in Eq. (5) and (7) updates the model parameters - i.e. the synaptic weights - only at those time points where one of the output neurons spikes. We denote by $p_S^*(\mathbf{y})$ the (empirical) distribution that is obtained from the observations of $\mathbf{y}(t)$ at the first S spike events $t_1^f, t_2^f, \dots, t_S^f$:

$$p_S^*(\mathbf{y}) = \frac{1}{S} \sum_{s=1}^S \delta(\mathbf{y} - \mathbf{y}(t_s^f)). \quad (73)$$

The distribution $p_S^*(\mathbf{y})$ that is seen by the learning rule in Eq. (5) depends not only on the time-continuous input stream $\mathbf{y}(t)$, but also on the concrete spike times t_s^f of the circuit. The output spikes thus serve as trigger events at which the continuous input signal is *sampled*.

The spike times $t_1^f, t_2^f, \dots, t_S^f$ and the total number of spikes S of the whole circuit within a time window of length T are distributed according to an inhomogeneous Poisson process with the instantaneous rate $R(t)$. For any stochastic realization of S and t_s^f in the time interval 0 to T , we can derive the expectation of the function $p_S^*(\mathbf{y})$ by taking the limit for $T \rightarrow \infty$ and call this the expected empirical distribution $p_{R(t)}^*(\mathbf{y})$. Thus

$$p_{R(t)}^*(\mathbf{y}) = \lim_{T \rightarrow \infty} \mathbb{E}_{S, t_1^f, \dots, t_S^f} \left[\frac{1}{S} \sum_{s=1}^S \delta(\mathbf{y} - \mathbf{y}(t_s^f)) \right] \quad (74)$$

$$= \lim_{T \rightarrow \infty} \mathbb{E}_S \left[\mathbb{E}_{t_1^f, \dots, t_S^f} \left[\frac{1}{S} \sum_{s=1}^S \delta(\mathbf{y} - \mathbf{y}(t_s^f)) \mid S \right] \right], \quad (75)$$

where we divided the expectation into two parts. Firstly we take the expectation over the total number S of spikes, secondly we take the expectation over the spike times t_1, \dots, t_S , given S . We now make use of the fact that for any inhomogeneous Poisson process $R(t)$, conditioned on the total number of events S within a certain time window T , the event times t_1^f, \dots, t_S^f are distributed as order statistics of S unordered independent samples t_1^f, \dots, t_S^f from the probability density $\frac{R(t')}{\int_0^T R(t) dt}$. The expectation $\mathbb{E}_{t_s^f} [f(t_s^f) | S]$ over an arbitrary function $f()$ is the integral $\int_0^T \frac{R(t')}{\int_0^T R(t) dt} f(t') dt'$, independent of the event number s , thus

$$p_{R(t)}^*(\mathbf{y}) = \lim_{T \rightarrow \infty} \mathbb{E}_S \left[\frac{1}{S} \sum_{s=1}^S \mathbb{E}_{t_s^f} [\delta(\mathbf{y} - \mathbf{y}(t_s^f)) | S] \right] \quad (76)$$

$$= \lim_{T \rightarrow \infty} \mathbb{E}_S \left[\frac{1}{S} S \int_0^T \frac{R(t')}{\int_0^T R(t) dt} \delta(\mathbf{y} - \mathbf{y}(t')) dt' \right] \quad (77)$$

$$= \lim_{T \rightarrow \infty} \mathbb{E}_S \left[\int_0^T \frac{R(t')}{\int_0^T R(t) dt} \delta(\mathbf{y} - \mathbf{y}(t')) dt' \right]. \quad (78)$$

Since the remaining term within the expectation operator \mathbb{E}_S is independent of S we obtain the final result

$$p_{R(t)}^*(\mathbf{y}) = \lim_{T \rightarrow \infty} \frac{1}{\int_0^T R(t) dt} \int_0^T R(t) \delta(\mathbf{y} - \mathbf{y}(t)) dt. \quad (79)$$

This shows that the output rate $R(t)$ acts as a multiplicative weighting of the contribution of the current input $\mathbf{y}(t)$ to the

expected empirical distribution $p_{R(t)}^*(\mathbf{y})$, which is learned in the limit of $t \rightarrow \infty$ by the simple STDP rule in Eq. (5) and (7).

It turns out that the condition of a constant rate $R(t)$ is by far stronger than necessary. In fact, it is easy to see from a comparison of Eq. (72) and Eq. (79), that $p_{R(t)}^*(\mathbf{y}) = p^*(\mathbf{y})$ for all values of \mathbf{y} if and only if the relative weight for the input value \mathbf{y} , which is $\frac{\int_0^T R(t) \delta(\mathbf{y} - \mathbf{y}(t)) dt}{\int_0^T \delta(\mathbf{y} - \mathbf{y}(t)) dt}$, is independent of \mathbf{y} in the limit $T \rightarrow \infty$. This is certainly true if $R(t)$ and $\mathbf{y}(t)$ are stochastically independent, i.e. $R(t)$ is not correlated to the occurrence of any specific value of \mathbf{y} .

Inhibition model in computer simulations. In our computer simulation the inhibition is implemented by adding a strongly negative impulse to the membrane potential of all z -neurons whenever one of them fires, which decays with a time constant of 5 ms back to its resting value. In addition, a noise term $v(t)$ is added to the membrane potential $u_k(t)$ that models background synaptic inputs through an Ornstein-Uhlenbeck (OU) process (as proposed in [110] for modeling in-vivo conditions) and causes stochastic firing. For each experiment, all parameters for the inhibition model are listed in “Simulation Parameters” in the Supplementary Material.

Details to Continuous-Time Interpretation with Realistically Shaped EPSPs

Let the external input vector \mathbf{x} consist of multiple discrete-valued functions in time $x_j(t)$, and let us assume that for every input x_j there exists an independent Poisson sampling process with rate r_j which generates spike times for the group of neurons y_i with $i \in G_j$. At every spike time t_j^f there is exactly one neuron in the group that fires a spike, and this is the neuron that is associated with the value $x_j(t_j^f)$. First, we analyze additive step-function EPSPs, i.e. the postsynaptic activation $\tilde{y}_i(t)$ is given by the convolution in Eq. (17) where K is a step-function kernel with $K(t) = 1$ for $0 < t < \sigma$ for a fixed EPSP-duration σ and $K(t) = 0$ otherwise. In order to understand the resulting distribution $q_k(t)$ in Eq. (4) as Bayesian inference we extend our underlying generative probabilistic model $p(\mathbf{x}, k | \theta)$ such that it contains multiple instances of the variable vector \mathbf{x} , called $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(L)}$, where L is the total number of spikes from all input neurons y_i within the time window $[t - \sigma, t]$. We can see every spike as a single event in continuous time. The full probabilistic model is defined as

$$p(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(L)}, k | \mathbf{w}) = p(k | \mathbf{w}) \prod_{l=1}^L p(\mathbf{x}^{(l)} | k, \mathbf{w}), \quad (80)$$

which defines that the multiple instances are modeled as being conditionally independent of each other, given k . Let the vectors $\hat{\mathbf{y}}^{(l)}$ describe the corresponding spike “patterns” in which every binary vector $\hat{\mathbf{y}}^{(l)}$ has exactly one 1 entry $\hat{y}_{i^{(l)}}^{(l)} = 1$. All other values are zero, thus it represents exactly one evidence for $\mathbf{x}^{(l)}$, i.e. $x_j^{(l)} = v(i^{(l)})$, with j , s.t. $i^{(l)} \in G_j$, according to the decoding in Eq. (8).

Due to the conditional independences in the probabilistic model every such evidence, i.e. every spike, contributes one factor $p(\hat{y}_i^{(l)} = 1 | k, \mathbf{w})$ to the likelihood term in the inference of the hidden node k . The inference is expressed as

$$p(k|\tilde{\mathbf{y}}^{(1)}, \dots, \tilde{\mathbf{y}}^{(L)}, \mathbf{w}) = \frac{\underbrace{\sum_{k'=1}^K e^{w_{k'0}} \prod_{i=1}^n (e^{w_{k'i}})^{\sum_{l=1}^L \tilde{y}_i^{(l)}}}_{p(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(L)}|\mathbf{w})} \cdot \underbrace{\prod_{i=1}^n (e^{w_{ki}})^{\sum_{l=1}^L \tilde{y}_i^{(l)}}}_{\text{likelihood } p(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(L)}|k, \mathbf{w})}}{e^{w_{k0}} \prod_{i=1}^n (e^{w_{ki}})^{\sum_{l=1}^L \tilde{y}_i^{(l)}}} \quad (81)$$

The identity $\tilde{y}_i(t) = \sum_{l=1}^L \tilde{y}_i^{(l)}$ reveals that the above posterior distribution is realized by the relative spike probability $q_k(t)$ of the network model according to Eq. (4), where $\tilde{\mathbf{y}}(t)$ replaces $\mathbf{y}(t)$ in the computation of the membrane potential $u_k(t)$. Due to the step function $K(t)$ the result of the convolution in $\tilde{y}_i(t)$ equals the number of spikes within the time window $[t-\sigma, t]$ from neuron y_i . The factor $e^{w_{ki}}$, which has the meaning $p(y_i=1|k, \mathbf{w})$ in the network model, is multiplied $\tilde{y}_i(t)$ times to the likelihood.

The above discrete probabilistic model gives an interpretation only for integer values of $\tilde{y}_i(t)$, i.e. for functions K such that $K(t)$ is 0 or any positive integer at any time t . For an interpretation of arbitrarily shaped EPSPs $K(t)$ - especially for continuously decaying functions - in the context of our probabilistic model, we now extend this weighting mechanism from integer valued weights to real valued weights by a linear interpolation of the likelihood in the log-space.

The obvious restrictions on the EPSP function $K(t)$ are that it is non-negative, zero for $t < 0$, and $\int_0^\infty K(t)dt < \infty$, in order to avoid acausal or nondecaying behavior, and unboundedly growing postsynaptic potentials at constant input rates. We assume the normalization $\max K(t) = 1$. Let again $t^{(1)}, \dots, t^{(l)}, \dots$ be the times of the past spiking events and $i^{(1)}, \dots, i^{(l)}, \dots$ be the indices of the corresponding input neurons. The output distribution $q_k(t)$ can be written as

$$q_k(t) = \frac{e^{w_{k0}} \prod_{l=1}^\infty (e^{w_{ki^{(l)}}})^{K(t-t^{(l)})}}{\sum_{k'=1}^K e^{w_{k'0}} \prod_{l=1}^\infty (e^{w_{k'i^{(l)}}})^{K(t-t^{(l)})}}, \quad (82)$$

which nicely illustrates that every single past spike at time $t^{(l)}$ is seen as an evidence in the inference, but that evidence is weighted with a value $K(t-t^{(l)})$, which is between 0 and 1.

The analogous interpolation for continuous-valued input activations $\tilde{y}_i(t)$ yields the learning rule in Eq. (18), which is illustrated in Fig. 2 as the “Complex STDP rule” (blue dashed curve). The resulting shape of the LTP part of the STDP curve is determined by the EPSP shape defined by $K(t)$. The positive part of the update in Eq. (18) is weighted by the value of $\tilde{y}_i(t)$ at the time of firing the postsynaptic spike. Negative updates are performed if $\tilde{y}_i(t)$ is close to zero, which indicates that no presynaptic spikes were observed recently.

The proof of stochastic convergence does not explicitly assume that $\mathbf{y}^{(l)}$ is a binary vector, but is valid for any (positive) random variable vector $\tilde{\mathbf{y}}^{(l)}$ with finite variance. Further, the proof assumes the condition that in every group G_j the sum of the input activities $\tilde{y}_i^{(l)}$ is 1 at all times or at least at those points in time at which one z_k neuron of the WTA-circuit fires. The condition can be relaxed such that the sum per group does not have to be equal to 1 but to any arbitrary (positive) constant if the corresponding normalization constraint is adapted accordingly. Due to the decaying character of the EPSP shape, this sum will never stay constant, even for very regular input patterns. If we only assumed a constant average activation within a group, allowing for stochastic fluctuations

around the target value, it turns out that this condition alone is not enough. We need to further assume that these stochastic fluctuations in the sum of every input group G_j are stochastically independent of the circuit's response z_k . This assumption is intricate and may depend on the data and the learning progress itself, so it will usually not be exactly fulfilled. We can, however, argue that we are close to independence if at least the sum of activity in every group G_j is independent of the value of the underlying abstract variable x_j .

In our simulations we obtain the input activations $\tilde{y}_i(t)$ by simulating biologically realistic EPSPs at every synapse, using α -kernels with plausible time constants to model the contributions of single input spikes.

Details to Spike-timing dependent LTD

We formalize the presynaptic activity of neuron y_i after a postsynaptic spike at time t^f by v_i , s.t. $v_i = 1$ if there is a spike from neuron y_i within the time window $[t^f, t^f + \sigma]$ and $v_i = 0$ otherwise. This trace is used purely for mathematical analysis, and cannot be known to the postsynaptic neuron at time t^f , since the future input activity is unknown. Mechanistically, however, v_i can be implemented as a trace updated by postsynaptic firing, and utilized for plasticity at the time of presynaptic firing [113]. Let us now consider the STDP rule illustrated by the red curve in Fig. 9, where a depression of the synapse happens only if there is a presynaptic spike within the short time window of length σ after the postsynaptic spike, i.e. if $v_i = 1$. The application of this STDP-rule in our neuronal circuit is equivalent to the circuit-spike triggered update rule

$$\Delta w_{ki} = z_k (c \cdot e^{-w_{ki}} y_i - d \cdot v_i) \quad (83)$$

which replaces Eq. (5). In analogy to Eq. (6) the equilibrium of this new update rule can be derived as

$$E[\Delta w_{ki}] = 0 \Leftrightarrow p(y_i = 1, z_k = 1) c e^{-w_{ki}} - p(v_i = 1, z_k = 1) d = 0 \quad (84)$$

$$\Leftrightarrow w_{ki} = \log p^*(y_i = 1 | z_k = 1) - \log p^*(v_i = 1 | z_k = 1) + \log \frac{c}{d}, \quad (85)$$

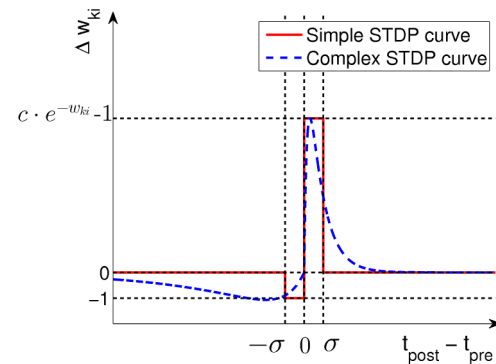


Figure 9. STDP learning curves with time-dependent LTD. Under the simple STDP model (red curve), weight-dependent LTP occurs only if the postsynaptic spike falls within a time window of length σ after the presynaptic spike, and LTD occurs in a time window of the same length, but for the opposite order of spikes. This can be extended to a more complex STDP rule (blue dashed curve), in which both LTP and LTD follow α -kernels with different time constants, typically with longer time-constants for LTD. doi:10.1371/journal.pcbi.1003037.g009

under the assumption that y_i, v_i and z_k are sampled from a stationary distribution $p^*(y_i, v_i, z_k)$. This shows that the synaptic weights can be interpreted as the log-likelihood ratio of the presynaptic neuron firing before instead of after the postsynaptic neuron. In other words, the neuron's synaptic weights learn the contrast between the current input pattern y_i that caused firing, and the following pattern of activity v_i . Note that any factor c (for LTP) or d (for LTD) only leads to a constant offset of the weight which - under the assumption that the offset is the same for all synapses - can be neglected due to the WTA circuit (see Methods "Weight offsets and positive weights").

Similarly to our analysis for the standard SEM rule, we can derive a continuous-time interpretation of the timing-dependent LTD rule. As we did in Eq. (17), we can define

$$\tilde{y}_i(t) = \sum_f K_P(t - t'_i) \quad \tilde{v}_i(t) = \sum_f K_D(t - t'_i), \quad (86)$$

where K_P is the same convolution kernel as in Eq. (17), and K_D is an arbitrary but time-inversed kernel, such that $K_D(t) = 0$ for positive t and $K_D(t) > 0$ for negative t . The value of v_i thus reflects a time-discounted sum of presynaptic activity immediately after the postsynaptic spike.

The complex STDP rule from Fig. 2, which models LTD as a constant time-independent depression, can be seen as an extreme case of the spike-timing dependent LTD rule. If K_D is a step function with $K_D(t) = \frac{1}{\sigma}$ in the interval $[-\sigma, 0]$ and 0 everywhere else, then v_i is just the average rate of presynaptic activity in the time interval $[t', t' + \sigma]$ following a postsynaptic spike. In the limit of $\sigma \rightarrow \infty$ this is equivalent to the overall spiking rate of the neuron y_i , which is proportional to the marginal $p(y_i)$ in the probabilistic model. Precisely, $v_i \rightarrow r p(y_i)$, where r is the base firing rate of an active input in our input encoding model. The equilibrium point of every weight w_{ki} becomes $\log p(y_i = 1 | z_k = 1) - \log p(y_i)$, neglecting the offsets induced by the constants c, d and r . It is easy to see that the probabilistic interpretation of the neuronal model from Eq. (4) is invariant under the transformation $w'_{ki} = w_{ki} - \log p(y_i)$, since

$$q_k(t) = \frac{e^{w_{k0} + \sum_{i=1}^n (w_{ki} - \log p(y_i)) y_i}}{\sum_{k'=1}^K e^{w_{k'0} + \sum_{i=1}^n (w_{k'i} - \log p(y_i)) y_i}} \quad (87)$$

$$= \frac{\left(e^{\sum_{i=1}^n y_i \log p(y_i)} \right) e^{w_{k0} + \sum_{i=1}^n w_{ki} y_i}}{\sum_{k'=1}^K \left(e^{\sum_{i=1}^n y_i \log p(y_i)} \right) e^{w_{k'0} + \sum_{i=1}^n w_{k'i} y_i}} \quad (88)$$

$$= \frac{e^{w_{k0} + \sum_{i=1}^n w_{ki} y_i}}{\sum_{k'=1}^K e^{w_{k'0} + \sum_{i=1}^n w_{k'i} y_i}}, \quad (89)$$

which proves that in our network model the complex STDP rule from Fig. 2 is equivalent to an offset-free STDP rule in the limit of an arbitrarily long window for LTD. In practice, of course, we can assume that the times between pre- and post-synaptic spikes are finite, and we have shown in Fig. 4 that as a result, very realistic shapes of STDP curves emerge at intermediate stimulation frequencies.

Supporting Information

Text S1 Supplement. Derivation of Variance tracking, Adaptation to changing input distributions, Invariance to Time-Warping, Simulation Parameters. (PDF)

Acknowledgments

We would like to thank Wulfram Gerstner for critical comments to an earlier version of this paper. MP would like to thank Rodney J. Douglas and Tobi Delbruck for their generous advice and support.

Author Contributions

Conceived and designed the experiments: BN MP WM. Performed the experiments: BN MP. Wrote the paper: BN MP LB WM.

References

- Rao RPN, Olshausen BA, Lewicki MS (2002) Probabilistic Models of the Brain. MIT Press.
- Doya K, Ishii S, Pouget A, Rao RPN (2007) Bayesian Brain: Probabilistic Approaches to Neural Coding. MIT-Press.
- Griffiths TL, Tenenbaum JB (2006) Optimal predictions in everyday cognition. *Psychological Science* 17: 767–773.
- Griffiths TL, Kemp C, Tenenbaum JB (2008) Bayesian models of cognition. In: Sun R, editor. *Handbook of Computational Cognitive Modeling*. Cambridge Univ. Press. chapter 3. p. 59100.
- Körding KP, Wolpert DM (2004) Bayesian integration in sensorimotor learning. *Nature* 427: 244–247.
- Fiser J, Berkes P, Orban G, Lengyel M (2010) Statistically optimal perception and learning: from behavior to neural representation. *Trends in Cogn Sciences* 14: 119–130.
- Ma WJ, Beck JM, Latham PE, Pouget A (2006) Bayesian inference with probabilistic population codes. *Nature Neuroscience* 9: 1432–1438.
- Dan Y, Poo M (2004) Spike timing-dependent plasticity of neural circuits. *Neuron* 44: 23–30.
- Feldman D (2012) The spike-timing dependence of plasticity. *Neuron* 75: 556–571.
- Song S, Abbott LF (2001) Cortical developing and remapping through spike timing-dependent plasticity. *Neuron* 32: 339–350.
- Kempster R, Gerstner W, van Hemmen JL (1999) Hebbian learning and spiking neurons. *Phys Rev E* 59: 4498–4514.
- Kempster R, Gerstner W, van Hemmen JL (2001) Intrinsic stabilization of output rates by spikebased Hebbian learning. *Neural Computation* 13: 2709–2741.
- Abbott LF, Nelson SB (2000) Synaptic plasticity: taming the beast. *Nature Neuroscience* 3: 1178–1183.
- Daoudal G, Debanne D (2003) Long-term plasticity of intrinsic excitability: learning rules and mechanisms. *Learn Mem* 10: 456–465.
- Grillner S, Graybiel A (2006) *Microcircuits: The Interface between Neurons and Global Brain Function*. MIT-Press.
- Douglas RJ, Martin KA (2004) Neuronal circuits of the neocortex. *Annual Review of Neuroscience* 27: 419–451.
- Hahnloser RHR, Sarpeshkar R, Mahowald MA, Douglas RJ, Seung HS (2000) Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* 405: 947–951.
- Carandini M, Heeger D (2012) Normalization as a canonical neural computation. *Nature Reviews Neuroscience* 13: 51–62.
- Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal Mach Intell* 20: 1254–1259.
- Nessler B, Pfeiffer M, Maass W (2009) STDP enables spiking neurons to detect hidden causes of their inputs. *Proceedings of NIPS Advances in Neural Information Processing Systems* 22: 1–9.
- Pfeiffer M, Nessler B, Douglas R, Maass W (2010) Reward-modulated Hebbian Learning of Decision Making. *Neural Computation* 22: 1399–1444.
- Maass W (2000) On the computational power of winner-take-all. *Neural Computation* 12: 2519–2535.
- Rutishauser U, Douglas R (2009) State-dependent computation using coupled recurrent networks. *Neural Computation* 21: 478–509.
- Buesing L, Bill J, Nessler B, Maass W (2011) Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. *PLoS Computational Biology* 7: e1002211.
- Pecevski D, Buesing L, Maass W (2011) Probabilistic Inference in General Graphical Models through Sampling in Stochastic Networks of Spiking Neurons. *PLoS Comput Biol* 7(12): e1002294.
- Hinton GE, Ghahramani Z (1997) Generative models for discovering sparse distributed representations. *Philos Trans R Soc Lond B Biol Sci* 352: 1177–1190.

27. Hinton GE, Osindero S, Teh YW (2006) A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation* 18: 1527–1554.
28. Keck C, Savin C, Lücke J (2012) Feedforward Inhibition and Synaptic Scaling - Two Sides of the Same Coin? *PLoS Computational Biology* 8: e1002432.
29. Sato M (1999) Fast learning of on-line EM algorithm. Technical report, ATR Human Information Processing Research Laboratories, Kyoto, Japan.
30. Sato M, Ishii S (2000) On-line EM Algorithm for the Normalized Gaussian Network. *Neural Computation* 12: 407–432.
31. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)* 39: 1–38.
32. Bishop CM (2006) *Pattern Recognition and Machine Learning*. New York: Springer.
33. Oster M, Douglas R, Liu S (2009) Computation with spikes in a winner-take-all network. *Neural Computation* 21: 2437–2465.
34. Okun M, Lampl I (2008) Instantaneous correlation of excitation and inhibition during ongoing and sensory-evoked activities. *Nature Neuroscience* 11: 535–537.
35. Ecker AS, Berens P, Keliris GA, Bethge M, Logothetis NK, et al. (2010) Decorrelated neuronal firing in cortical microcircuits. *Science* 327: 584–587.
36. Fino E, Yuste R (2011) Dense inhibitory connectivity in neocortex. *Neuron* 69: 1188–1203.
37. Gerstner W, Kistler WM (2002) *Spiking Neuron Models*. Cambridge: Cambridge University Press.
38. Jolivet R, Rauch A, Lüscher H, Gerstner W (2006) Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of Computational Neuroscience* 21: 35–49.
39. Caporale N, Dan Y (2008) Spike timing-dependent plasticity: a Hebbian learning rule. *Annual Review of Neuroscience* 31: 25–46.
40. Sjöström PJ, Turrigiano GG, Nelson S (2001) Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32: 1149–1164.
41. Toyozumi T, Pfister JP, Aihara K, Gerstner W (2005) Generalized Bienenstock-Cooper-Munro rule for spiking neurons that maximizes information transmission. *Proc Natl Acad Sci USA* 102: 5239–5244.
42. Pfister J, Toyozumi T, Barber D, Gerstner W (2006) Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Computation* 18: 1318–1348.
43. Bi G, Poo M (1998) Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience* 18: 10464–10472.
44. Hebb DO (1949) *The Organization of Behavior*. New York: Wiley.
45. Cudmore RH, Turrigiano GG (2004) Long-term potentiation of intrinsic excitability in LV visual cortical neurons. *Journal of Neurophysiology* 92: 341–348.
46. Turrigiano G (2011) Too many cooks? intrinsic and synaptic homeostatic mechanisms in cortical circuit refinement. *Annual Review of Neuroscience* 34: 89–103.
47. Habenschuss S, Bill J, Nessler B (2012) Homeostatic plasticity in Bayesian spiking networks as Expectation Maximization with posterior constraints. In: *Advances in Neural Information Processing Systems* 25, pp. 782–790.
48. Ghahramani Z, Michael MJ (1997) Mixture models for learning from incomplete data. In: *Computational Learning Theory and Natural Learning Systems: Volume IV: Making Learning Systems Practical*. MA, USA: MIT Press Cambridge, pp. 67–85.
49. Masquelier T, Hugues E, Deco G, Thorpe S (2009) Oscillations, Phase-of-Firing Coding, and Spike Timing-Dependent Plasticity: An Efficient Learning Scheme. *Journal of Neuroscience* 29: 13484–13493.
50. Gilson M, Masquelier T, Hugues E (2011) STDP Allows Fast Rate-Modulated Coding with Poisson-Like Spike Trains. *PLoS Computational Biology* 7: e1002231.
51. Jank W (2006) The EM algorithm, its randomized implementation and global optimization: Some challenges and opportunities for operations research. *Perspectives in Operations Research* : 367–392.
52. Kushner H, Yin G (2003) *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Verlag.
53. Gjorgjieva J, Clopath C, Audet J, Pfister JP (2011) A triplet spike-timing-dependent plasticity model generalizes the Bienenstock-Cooper-Munro rule to higher-order spatiotemporal correlations. *PNAS* 108: 19383–19388.
54. Song S, Miller KD, Abbott LF (2000) Competitive Hebbian learning through spike-timing dependent synaptic plasticity. *Nature Neuroscience* 3: 919–926.
55. Kobayashi K, Poo M (2004) Spike Train Timing-Dependent Associative Modification of Hippocampal CA3 Recurrent Synapses by Mossy Fibers. *Neuron* 41: 445–454.
56. Morrison A, Diesmann M, Gerstner W (2008) Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics* 98: 459–478.
57. Clopath C, Büsing L, Vasilaki E, Gerstner W (2010) Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nature Neuroscience* 13: 344–352.
58. Graupner M, Brunel N (2012) Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location. *PNAS* 109: 3991–3996.
59. Masquelier T, Guyonneau R, Thorpe S (2008) Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLoS ONE* 3: e1377.
60. Froemke RC, Dan Y (2002) Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature* 415: 433–438.
61. Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381: 607–609.
62. Li Y, Fitzpatrick D, White L (2006) The development of direction selectivity in ferret visual cortex requires early visual experience. *Nature neuroscience* 9: 676–681.
63. Espinosa JS, Stryker MP (2012) Development and Plasticity of the Primary Visual Cortex. *Neuron* 75: 230–249.
64. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86: 2278–2324.
65. Cireşan DC, Meier U, Gambardella LM, Schmidhuber J (2010) Deep, big, simple neural nets for handwritten digit recognition. *Neural computation* 22: 3207–20.
66. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313: 504–507.
67. Ranzato M, Huang F, Boureau YL, Lecun Y (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'07)*, pp. 1–8.
68. Masquelier T, Guyonneau R, Thorpe S (2009) Competitive STDP-based spike pattern learning. *Neural Computation* 21: 1259–1276.
69. Oaksford M, Chater N (2007) *Bayesian Rationality: The Probabilistic Approach to Human Reasoning*. Oxford University Press.
70. Salakhutdinov R, Hinton G (2009) Deep boltzmann machines. In: *Proceedings of the international conference on artificial intelligence and statistics*. Cambridge, MA: MIT Press, volume 5, pp. 448–455.
71. Desimone R (1991) Face-selective cells in the temporal cortex of monkeys. *Journal of Cognitive Neuroscience* 3: 1–8.
72. Poon H, Domingos P (2011) Sum-product networks: A new deep architecture. In: *Computer Vision Workshops (ICCV Workshops)*, 2011 IEEE International Conference on. IEEE, pp. 689–690.
73. Nessler B, Pfeiffer M, Maass W (2009) Hebbian learning of Bayes optimal decisions. In *Proc of NIPS 2008: Advances in Neural Information Processing Systems* 21.
74. Indiveri G, Linares-Barranco B, Hamilton T, van Schaik A, Etienne-Cummings R, et al. (2011) Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience* 5: 1–23.
75. Schemmel J, Brüderle D, Meier K, Ostendorf B (2007) Modeling synaptic plasticity within networks of highly accelerated I&F neurons. In: *International Symposium on Circuits and Systems, ISCAS 2007. IEEE*, pp. 3367–3370.
76. Jin X, Lujan M, Plana L, Davies S, Temple S, et al. (2010) Modeling spiking neural networks on SpiNNaker. *Computing in Science & Engineering* 12: 91–97.
77. Rumelhart DE, Zipser D (1988) Feature discovery by competitive learning. In: *Waltz D, Feldman JA, editors, Connectionist Models and Their Implications: Readings from Cognitive Science*. Ablex Publishing Corporation, pp. 205–242.
78. Nowlan SJ (1990) Maximum likelihood competitive learning. In: *Touretzky D, editor, Advances in Neural Information Processing Systems (NIPS)*, San Mateo, California: Morgan Kaufmann, volume 2, pp. 574–582.
79. Celeux G, Diebolt J (1985) The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Comput Statist Quater* 2: 73–82.
80. Nowlan SJ (1991) Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures. Technical Report CS-91-126, Carnegie Mellon University, Pittsburgh.
81. Jordan MI, Jacobs RA (1994) Hierarchical mixtures of experts and the algorithm. *Neural Computation* 6: 181–214.
82. Neal RM, Hinton GE (1998) A view of the EM algorithm that justifies incremental, sparse, and other variants. In: *Jordan MI, editor, Learning in Graphical Models*, Kluwer Academic Publishers, volume 89, pp. 355–370.
83. Savin C, Joshi P, Triesch J (2010) Independent Component Analysis in Spiking Neurons. *PLoS Computational Biology* 6: e1000757.
84. Dayan P, Abbott LF (2001) *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press.
85. Gupta A, Long LN (2009) Hebbian learning with winner take all for spiking neural networks. *IEEE International Joint Conference on Neural Networks*: 1189–1195.
86. Gupta A, Long LN (2007) Character recognition using spiking neural networks. *IJCNN* : 53–58.
87. Rao RP (2005) Hierarchical Bayesian inference in networks of spiking neurons. In: *Advances in Neural Information Processing Systems*. MIT Press, volume 17, pp. 1113–1120.
88. Rao RPN (2007) Neural models of Bayesian belief propagation. In: *Doya K, Ishii S, Pouget A, Rao RPN, editors, Bayesian Brain.*, Cambridge, MA: MIT-Press, pp. 239–267.
89. Zemel R, Huys QJM, Natarajan R, Dayan P (2005) Probabilistic computation in spiking populations. In: *Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference*, volume 17, pp. 1609–1616.
90. Deneve S (2008) Bayesian spiking neurons I: Inference. *Neural Computation* 20: 91–117.
91. Deneve S (2008) Bayesian spiking neurons II: Learning. *Neural Computation* 20: 118–145.

92. Gütig R, Sompolinsky H (2009) Time-warp invariant neuronal processing. *PLoS Biology* 7: e1000141.
93. Rezende DJ, Wierstra D, Gerstner W (2011) Variational learning for recurrent spiking networks. In: Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K, editors, *Advances in Neural Information Processing Systems* 24. pp. 136–144.
94. Brea J, Senn W, Pfister JP (2011) Sequence learning with hidden units in spiking neural networks. In: Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K, editors, *Advances in Neural Information Processing Systems* 24. pp. 1422–1430.
95. Liao D, Jones A, Malinow R (1992) Direct measurement of quantal changes underlying long-term potentiation in CA1 hippocampus. *Neuron* 9: 1089–1097.
96. Montgomery JM, Pavlidis P, Madison DV (2001) Pair recordings reveal all-silent synaptic connections and the postsynaptic expression of long-term potentiation. *Neuron* 29: 691–701.
97. Debanne D, Poo MM (2010) Spike-timing dependent plasticity beyond synapse – pre- and postsynaptic plasticity of intrinsic neuronal excitability. *Front Syn Neurosci* 2: 21.
98. Olshausen BA, Field DJ (2005) How close are we to understanding V1? *Neural Computation* 17: 1665–1699.
99. Kerr JND, de Kock CPJ, Greenberg DS, Bruno RM, Sakmann B, et al. (2007) Spatial organization of neuronal population responses in layer 2/3 of rat barrel cortex. *Journal of Neuroscience* 27: 13316–13328.
100. Nikolic D, Hauesler S, Singer W, Maass W (2009) Distributed fading memory for stimulus properties in the primary visual cortex. *PLoS Biology* 7: 1–19.
101. Faisal AA, Selen LPJ, Wolpert DM (2008) Noise in the nervous system. *Nature Reviews Neuroscience* 9: 292–303.
102. Gilbert CD, Sigman M, Crist RE (2001) The neural basis of perceptual learning. *Neuron* 31: 681–697.
103. Gilbert CD, Li W, Piech V (2009) Perceptual learning and adult cortical plasticity. *Journal of Physiology* 337: 2743–2751.
104. Goard M, Dan Y (2009) Basal forebrain activation enhances cortical coding of natural scenes. *Nature Neuroscience* 12: 1444–1449.
105. de Villers-Sidani E, Merzenich MM (2011) Lifelong plasticity in the rat auditory cortex: basic mechanisms and role of sensory experience. *Prog Brain Res* 191: 119–131.
106. Merabet LB, Pascual-Leone A (2010) Neural reorganization following sensory loss: the opportunity of change. *Nature Reviews Neuroscience* 11: 44–52.
107. Maldonado P, Babul C, Singer W, Rodriguez E, Berger D, et al. (2008) Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images. *Journal of Neurophysiology* 100: 1523–1532.
108. Uhlhaas PJ, Pipa G, Lima B, Melloni L, Neuenschwander S, et al. (2009) Neural synchrony in cortical networks: history, concept and current status. *Front Integr Neurosci* 3: 17.
109. Uhlhaas PJ, Roux F, Rodriguez E, Rotarska-Jagiela A, Singer W (2010) Neural synchrony and the development of cortical networks. *Trends in Cogn Sciences* 14: 72–80.
110. Destexhe A, Rudolph M, Fellous JM, Sejnowski TJ (2001) Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons. *Neuroscience* 107: 13–24.
111. Binzegger T, Douglas RJ, Martin KA (2007) Stereotypical bouton clustering of individual neurons in cat primary visual cortex. *Journal of Neuroscience* 27: 12242–12254.
112. Markov NT, Misery P, Falchier A, Lamy C, Vezoli J, et al. (2011) Weight Consistency Specifies Regularities of Macaque Cortical Networks. *Cerebral Cortex* 21: 1254–1272.
113. Schmiedt J, Albers C, Pawelzik K (2010) Spike timing-dependent plasticity as dynamic filter. In: *Advances in Neural Information Processing Systems* 23. pp. 2110–2118.