Independent Component Analysis in Spiking Neurons

Cristina Savin*, Prashant Joshi, Jochen Triesch

Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany

Abstract

Although models based on independent component analysis (ICA) have been successful in explaining various properties of sensory coding in the cortex, it remains unclear how networks of spiking neurons using realistic plasticity rules can realize such computation. Here, we propose a biologically plausible mechanism for ICA-like learning with spiking neurons. Our model combines spike-timing dependent plasticity and synaptic scaling with an intrinsic plasticity rule that regulates neuronal excitability to maximize information transmission. We show that a stochastically spiking neuron learns one independent component for inputs encoded either as rates or using spike-spike correlations. Furthermore, different independent components can be recovered, when the activity of different neurons is decorrelated by adaptive lateral inhibition.

Citation: Savin C, Joshi P, Triesch J (2010) Independent Component Analysis in Spiking Neurons. PLoS Comput Biol 6(4): e1000757. doi:10.1371/journal.pcbi.1000757

Editor: Abigail Morrison, RIKEN Brain Science Institute, Japan

Received October 22, 2009; **Accepted** March 23, 2010; **Published** April 22, 2010

Funding: The authors are supported by EC MEXT project PLICON and the German Federal Ministry of Education and Research (BMBF) within the Bernstein Focus: Neurotechnology through research grant 01GQ0840. 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: savin@fias.uni-frankfurt.de

Introduction

Independent component analysis is a well-known signal processing technique for extracting statistically independent components from high-dimensional data. For the brain, ICA-like processing could play an essential role in building efficient representations of sensory data [1–4]. However, although many algorithms have been proposed for solving the ICA problem [5], only few consider spiking neurons. Moreover, the existing spike-based models [6,7] do not answer the question how this type of learning can be realized in networks of spiking neurons using local, biologically plausible plasticity mechanisms (but see [8]).

Classic ICA algorithms often exploit the non-Gaussianity principle, which allows the ICA model to be estimated by maximizing some non-Gaussianity measure, such as kurtosis or negentropy [5]. A related representational principle is sparse coding, which has been used to explain various properties of V1 receptive fields [9]. Sparse coding states that only a small number of neurons are activated at the same time, or alternatively, that each individual unit is activated only rarely [10]. In the context of neural circuits, it offers a different interpretation of the goal of the ICA transform, from the perspective of metabolic efficiency. As spikes are energetically expensive, neurons have to operate under tight metabolic constraints [11], which affect the way information is encoded. Moreover, experimental evidence supports the idea that the activity of neurons in V1 is sparse. Close to exponential distributions of firing rates have been reported in various visual areas in response to natural scenes [12].

Interestingly, certain homeostatic mechanisms are thought to regulate the distribution of firing rates of a neuron [13]. These intrinsic plasticity (IP) mechanisms adjust ionic channel properties, inducing persistent changes in neuronal excitability [14]. They have been reported for a variety of systems, in brain slices and neuronal cultures [14,15] and they are generally thought to play a role in maintaining system homeostasis. Moreover, IP has been found to occur in behaving animals, in response to learning (see [14] for review).

From a computational perspective, it is believed that IP may maximize information transmission of a neuron, under certain metabolic constraints [13]. Additionally, we have previously shown for a rate neuron model that, when interacting with Hebbian synaptic plasticity, IP allows the discovery of heavy-tailed directions in the input [16]. Here, we extend these results for a network of spiking neurons. Specifically, we combine spike-timing dependent plasticity (STDP) [17–19], synaptic scaling [20] and an IP rule similar to [16], which tries to make the distribution of instantaneous neuronal firing rates close to exponential.

We show that IP and synaptic scaling complement STDP learning, allowing single spiking neurons to learn useful representations of their inputs for several ICA problems. First, we show that output sparsification by IP together with synaptic learning is sufficient for demixing two zero mean supergaussian sources, a classic formulation of ICA. When using biologically plausible inputs and STDP, complex tasks, such as Foldiák's bars problem [21], and learning oriented receptive fields for natural visual stimuli, can be tackled. Moreover, a population of neurons learns to extract several independent components if the activity of different neurons are decorrelated by adaptive lateral inhibition. When investigating the mechanisms how learning occurs in our model, we show that IP is necessary for learning, as it enforces a sparse output, guiding learning towards heavy-tailed directions in the input. Lastly, for specific STDP implementations, we show that IP shifts the threshold between potentiation and depression, similar to a sliding threshold for Bienenstock-Cooper-Munro (BCM) learning [22].

The underlying assumption behind our approach, implicit in all standard models of V1 receptive field development, is that both input and output information are encoded in rates. In this light, one may

Author Summary

How the brain learns to encode and represent sensory information has been a longstanding question in neuroscience. Computational theories predict that sensory neurons should reduce redundancies between their responses to a given stimulus set in order to maximize the amount of information they can encode. Specifically, a powerful set of learning algorithms called Independent Component Analysis (ICA) and related models, such as sparse coding, have emerged as a standard for learning efficient codes for sensory information. These algorithms have been able to successfully explain several aspects of sensory representations in the brain, such as the shape of receptive fields of neurons in primary visual cortex. Unfortunately, it remains unclear how networks of spiking neurons can implement this function and, even more difficult, how they can learn to do so using known forms of neuronal plasticity. This paper solves this problem by presenting a model of a network of spiking neurons that performs ICA-like learning in a biologically plausible fashion, by combining three different forms of neuronal plasticity. We demonstrate the model's effectiveness on several standard sensory learning problems. Our results highlight the importance of studying the interaction of different forms of neuronal plasticity for understanding learning processes in the brain.

think of our current work as a translation of the model in [16] to a spike-based version. However, the principles behind our model are more general than suggested by our work with rate neurons. We show that the same rule can be applied when inputs are encoded as spike-spike correlation patterns, where a rate-based model would fail.

Results

A schematic view of the learning rules is shown in Fig. 1. The stochastically spiking neuron [23] generates spikes as an inhomogeneous Poisson process, with mean expressed as a function of the total incoming current to the neuron $g(u)$, parametrized by variables r_0, u_0 and u_α. This transfer function is optimized by adapting the three parameters to make the distribution of instantaneous firing rates of the neuron approximatively exponential (for a complete mathematical formulation, see the Methods section). Additionally, Hebbian synaptic plasticity, implemented by nearest-neighbor STDP [24], changes incoming weights and a synaptic scaling mechanism keeps the sum of all incoming weights constant over time.

A simple demixing problem

To illustrate the basic mechanism behind our approach, we first ask if enforcing a sparse prior by IP and Hebbian learning can yield a valid ICA implementation for the classic problem of demixing two supergaussian independent sources. In the standard form of this problem, zero mean, unit variance inputs ensure that the covariance matrix is the identity, such that simple Hebbian learning with a linear unit (equivalent to principal component analysis) would not be able to exploit the input statistics and would just perform a random walk in the input space. This is, however, a purely mathematical formulation, and does not make much sense in the context of biological neurons. Inputs to real neurons are bounded and —in a rate-based encoding— all positive. Nonetheless, we chose this standard formulation to illustrate the basic underlying principles behind our model. Below, we will consider different spike-based encodings of the input and learning with STDP.

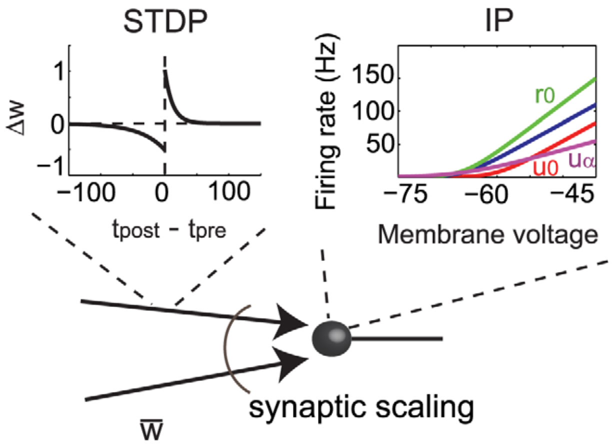

Figure 1. Overview of plasticity rules used for ICA-like learning. Synapse weights $\overline{w}$ are modified by nearest-neighbor STDP and synaptic scaling. Additionally, intrinsic plasticity changes the neuron's transfer function by adjusting three parameters r_0, u_0, and u_α. Different transfer functions show the effects of changing each of the three parameters individually relative to the default case depicted in blue. Namely, r_0 gives the slope of the curve, u_0 shifts the entire curve left or right, while u_α can be used for rescaling the membrane potential axis. Here, r_0 is increased by a factor of 1.5, u_0 by 5 mV, u_α by a factor of 1.2.
doi:10.1371/journal.pcbi.1000757.g001

As a special case of a demixing problem, we use two independent Laplacian distributed inputs, with unit variance: $p_{U_i}(u_i)=\frac{1}{\sqrt{2}}e^{-\sqrt{2}|u_i|}$, $p(u_1,u_2)=p_{U_1}(u_1)\cdot p_{U_2}(u_2)$. For the linear superposition, we use a rotation matrix A:

$$A=\begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}, \tag{1}$$

where α is the angle of rotation, resulting in a set of inputs $u'=Au$. Samples are drawn at each time step from the input distribution and are mapped into a total input to the neuron as $u_t=w_1u'_1+w_2u'_2$, with the weight vector w normalized). The neuron's transfer functions $g(u_t)$, the same as for our spiking model (Fig. 1), is adapted based on our IP rule, to make the distribution of firing rates exponential. For simplicity, here weights change by classic Hebbian learning: $\Delta w_i=\eta u'_i g(u_t)$, with η being the synaptic learning rate (see Methods for details). Similar results can be obtained when synaptic changes follow the BCM rule.

In Fig. 2A we show the evolution of synaptic weights for different starting conditions. As our IP rule adapts the neuron parameters to make the output distribution sparse (Fig. 2B,C), the weight vector aligns itself along the direction of one of the sources. With this simple model, we are able to demix a linear combination of two independent sources for different mixing matrices and different weights constraints (Fig. 2D), as any other single-unit implementation of ICA.

One neuron learns an independent component

After showing that combining IP and synaptic learning can solve a classical formulation of ICA, we focus on spike-based, biologically plausible inputs. In the following, STDP is used for implementing synaptic learning, while the IP and the synaptic scaling implementations remain the same.

Demixing with spikes. The demixing problem above can be solved also in a spike-based setting, after a few changes. First, the

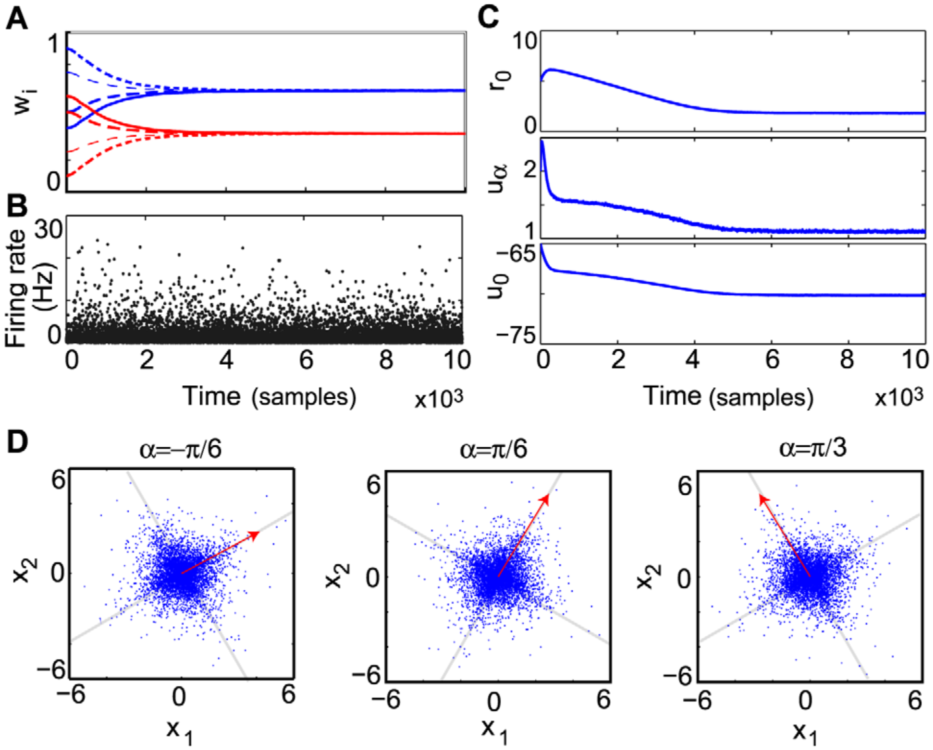

Figure 2. A demixing problem: two rotated Laplace directions. (A) Evolution of the weights (w_1 in blue, w_2 in red) for different initial conditions, with $\alpha = -\frac{\pi}{6}$, and L_1 weight normalization. (B) Evolution of the instantaneous firing rate g, sampled each 1000 ms, for the initial weights $w_1 = 0.4$, $w_2 = 0.6$. (C) Corresponding changes in transfer function parameters, with r_0 in Hz and u_0 and u_α in mV. (D) Final weight vector for different rotation angles α (in red). In the first example, normalization was done by $|w|_{L_1} = 1$ (the estimated rotation angle is $\tilde{\alpha} = \arctan(w_2/w_1) = 0.5215$, instead of the actual value 0.5236); for the others $|w|_{L_2} = 1$ was used. In all cases the final weight vector was scaled by a factor of 5, to improve visibility.
doi:10.1371/journal.pcbi.1000757.g002

positive and negative inputs have to be separated into on- and off-channels ($u_i^{+/-}$) and converted into Poisson spike trains of a certain duration T (see Methods), with the corresponding rates $u_i^{+/-}$ (note that the inputs are no longer white in this four-dimensional space). Secondly, to avoid the unbiological situation of having few very strong inputs, such that single presynaptic spikes always elicit a spike in the postsynaptic neuron, each channel consists of several (here, 25) synapses, with independent inputs having the same firing rate. Synapses are all positive and adapt by STDP, under the normalization $\sum w_i = 1$. A more detailed description of the parameters can be found in the Methods section.

The evolution of the weights is shown in Fig. 3A. The corresponding receptive field of the neuron can be obtained by projecting the weight vector back onto the original two-dimensional space (details of this procedure are described in the Methods and Text S1). As in the original formulation of the problem, the neuron receptive field slowly aligns itself along one of the independent components (Fig. 3B).

Foldiák's bars: input encoded as firing rates. As a second test case for our model, we consider Foldiák's well-known bars problem [21]. This is a classic non-linear ICA problem and is interesting from a biological perspective, as it mimics relevant nonlinearities, e.g. occlusions. In the classic formulation, for a two-dimensional input x, of size $N \times N$, a single bar must be learned after observing samples consisting of the nonlinear superposition of $2N$ possible individual bars (see Fig. 4A), each appearing independently, with probability $\frac{1}{2N}$ ($N = 10$). The superposition is non-linear, as the intersection of two bars has the same intensity as the other pixels in the bars (a binary OR).

In our implementation, the input vector is normalized and the value of each pixel $x_{i,j}$ is converted into a corresponding Poisson spike train of duration on the order of a fixation duration [25]. The details of the experimental setup are described in the Methods section. As the IP mechanism begins to take effect, making neuronal activity sparse, the receptive field of the neuron slowly adapts to one of the bars (Fig. 4B). This effect is robust for a wide range of parameters (see Text S2) and, as suggested by our previous results for rate neurons [16], does not critically depend on the particular implementation of the synaptic learning. We obtain similar results with an additive [26] and a simple triplet [27] model of STDP.

The input normalization makes a bar in an input sample containing a single IC stronger than a bar in a sample containing multiple ICs. This may suggest that a single component emerges by preferentially learning 'easy' examples, i.e. examples with a single bar. However, this is not the case and one bar is learned even when single bars never appear in the input, as in [28]. Specifically, we use a variant of the bars problem, in which the input always consists of 4 distinct bars, selected at random. In this case, the neuron correctly learns a single component. Moreover, similar results can be obtained for 2–5 bars (see Text S2), using the same set of parameters.

Foldiák's bars: input encoded by spike-spike correlations. In the previous experiments, input information was encoded as firing rates. In the following, we show that this

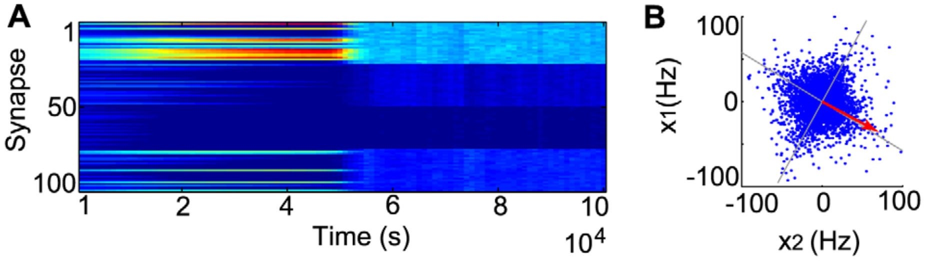

Figure 3. The demixing problem with inputs encoded as spike trains. (A) Evolution of the weights for a rotation angle $\alpha = \frac{\pi}{6}$. (B) Final corresponding weight vector in the original two-dimensional space. The final weight vector is scaled by a factor of 100, to improve visibility.
doi:10.1371/journal.pcbi.1000757.g003

stimulus encoding is not critical and the presence of spike-spike correlations is sufficient to learn an independent component, even in the absence of any presynaptic rate modulation. To demonstrate this, we consider a slight modification of the above bars problem, with input samples consisting of always two bars, each two pixel wide, with N distinct bars. This is a slightly more difficult task, as wide bars emphasize non-linearities due to overlap, but can be solved by our model with a rate encoding (see Text S2).

In this case, all inputs have the same mean firing rate and the information about whether a pixel belongs to a bar or not is encoded in correlation patterns between different inputs (see Methods). Specifically, background inputs are uncorrelated, while inputs belonging to bars are all pairwise correlated, with a correlation coefficient $C = 0.75$ (see Fig. 5B).

As for the rate-based encoding, the neuron is able to reliably learn a bar (Fig. 5C, D). Similar results were obtained for the version with always two bars, each one pixel wide, but with slower convergence. The fact that our approach also works for correlated inputs rests on the properties of STDP and IP. In the original rate formulation, strong inputs lead to higher firing in the postsynaptic neuron, causing the potentiation of the corresponding synapses. Similarly, if presynaptic inputs fire all with the same rate, correlated inputs are more successful in driving the neuron and hence their weights are preferentially strengthened [26]. Moreover, as before, IP enforces sparse post-synaptic responses, guiding learning towards a heavy-tailed direction in the input (a more formal analysis of the interaction between IP and STDP is presented below).

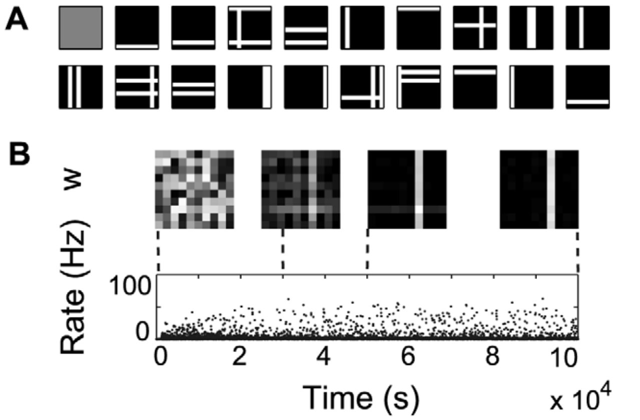

Figure 4. Learning a single independent component for the bars problem. (A) A set of randomly generated samples from the input distribution, (B) Evolution of the neuron's receptive field as the IP rule converges and instantaneous firing rate of the neuron. Each dot corresponds to the instantaneous firing rate ($g(u(t))\cdot R(t)$) sampled each 500 ms.
doi:10.1371/journal.pcbi.1000757.g004

Due to its stochastic nature, our neuron model is not particularly sensitive to input correlations, hence C cannot be too small. Stable receptive fields with a single 2 pixel wide bar are still obtained for lower correlation coefficients ($C = 0.5$), but with slower convergence. We expect better results with a deterministic neuron model, such as the leaky integrate-and-fire. An approximation of IP, based on moment matching could be used in this case. Additionally, STDP-induced competition (due to the relatively small number of inputs, in some cases one weight grows big enough to elicit a spike in the postsynaptic neuron) [26] enforces some constraints on the model parameters to ensure the stability of a solution with multiple non-zero weights. This could be done by increasing the size of the input, restricting the overall mean firing of the neuron μ or by slightly changing the STDP parameters (see Methods). These parameter changes do not affect learning with the rate-based encoding, however.

Natural scenes input. The third classical ICA problem we consider is the development of V1-like receptive fields for natural scenes [3]. Several computational studies have emphasized that simple cell receptive fields in V1 may be learned from the statistics of natural images by ICA or other similar component extraction algorithms [9,29,30]. We hypothesized that the same type of computation could be achieved in our spiking neuron model, by combining different forms of plasticity. Only a rate encoding was used for this problem, partly for computational reasons and partly because it is not immediately obvious how a correlation-based encoding would look like in this case.

We use a set of images from the van Hateren database [31], with standard preprocessing (see Methods). The rectified values of the resulting image patches are linearly mapped into a firing frequency for an on- and off-input population, as done for the bars. The STDP, IP and other simulation parameters are the same as before.

As shown in Fig. 6A, the receptive field of the neuron computed as the difference between the weights of the on- and the off- input populations (depicted in Fig. 6B) evolves to an oriented filter, similar to those obtained by other ICA learning procedures [9,29,30,32,33]. A similar receptive field can be obtained by reverse correlation from white noise stimuli. The least-mean-square-error fit to a Gabor wavelet [34] is shown in Fig. 6C. As vertical edges are usually over-represented in the input, the neuron will typically learn a vertical edge filter, with a phase shift depending on the initial conditions. The receptive field has low spatial frequency, but more localized solutions result for a neural population (see below).

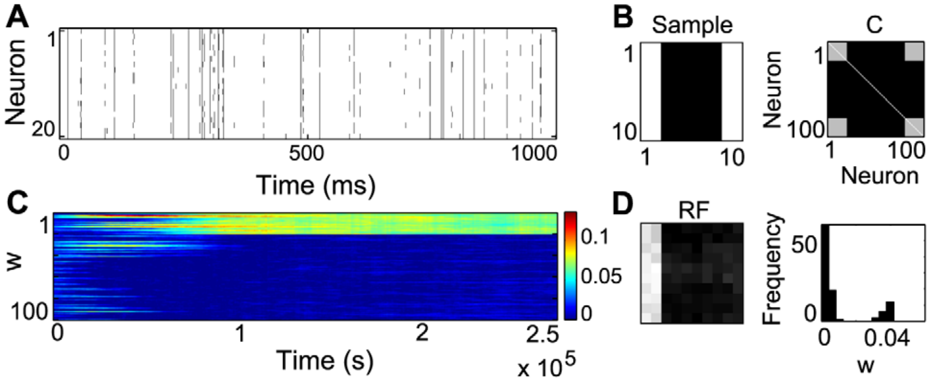

Figure 5. Bars in a correlation-based encoding. (A) Example of 20 spike trains with $C = 0.75$. (B) A sample containing two 2-pixel wide bars and the corresponding covariance matrix used for its encoding. (C) Evolution of the weights during learning. (D) Final receptive field and corresponding weights histogram.
doi:10.1371/journal.pcbi.1000757.g005

ICA in a neuron population

So far, learning has been restricted to a single neuron. For learning multiple independent components, we implement a neuron population in which the activities of different neurons are decorrelated by adaptive lateral inhibition. This approach is standardly used for feature extraction methods based on single-unit contrast functions [30]. Here, we consider a simple scheme for parallel (symmetrical) decorrelation. The all-to-all inhibitory weights (Fig. 7A) change by STDP and are subject to synaptic scaling, as done for the input synapses. We only use a rate-based encoding for this case, due to computational overhead, which also limits the size of networks we can simulate.

We consider a population of 10 neurons. In order to have a full basis set for the bars problem, we use 2 pixel wide bars. For this case, our learning procedure is able to recover the original basis (Fig. 7B). As lateral inhibition begins to take effect, the average correlation coefficient between the responses of different neurons in the population decreases (Fig. 7C), making the final inhibitory weights unspecific (Fig. 7D). As decorrelation is not a sufficient condition for independence, we show that, simultaneously, the normalized mutual information decreases (see Methods for details). Using the same network for the image patches, we obtain oriented, localized receptive fields (Fig. 7E).

Due to the adaptive nature of IP, the balance between excitation and inhibition does not need to be tightly controlled, allowing for robustness to changes in parameters. However, the inhibition strength influences the time required for convergence (the stronger the inhibition, the longer it takes for the system to reach a stable state). A more important constraint is that the adaptation of inhibitory connections needs to be faster than that of feedforward connections to allow for efficient decorrelation (see Methods for parameters).

Is IP necessary for learning?

We wondered what the role of IP is in this learning procedure. Does IP simply find an optimal nonlinearity for the neuron's transfer function, given the input, something that could be computed offline (as for InfoMax [29]), or is the interaction between IP and STDP critical for learning? To answer this question, we go back to Foldiák's bars. We repeat our first bars experiment (Fig. 4) for a fixed gain function, given by the parameters obtained after learning ($r_0 = 23.8$ Hz, $u_0 = -66.4$ mV, $u_\alpha = 1.1$ mV). In this case, the receptive field does not evolve to an IC (Fig. 8). This suggests that ICA-like computation relies on the interplay between weight changes and the corresponding readjustment of neuronal excitability, which forces the output to be sparse. Note that this result holds for simulation times significantly larger than in the experiment before, where a bar emerged after $5 \cdot 10^4$ s, suggesting that, even if the neuron would eventually learn a bar, it would take significantly longer to do so.

We could assume that the neuron failed to learn a bar for the fixed transfer function just because the postsynaptic firing was too low, slowing down learning. Hence, it may be that a simpler rule, regulating just the mean firing rate of the neuron, would suffice to learn an IC. To test this hypothesis, we construct an alternative IP rule, which adjusts just r_0 to preserve the average firing rate of the

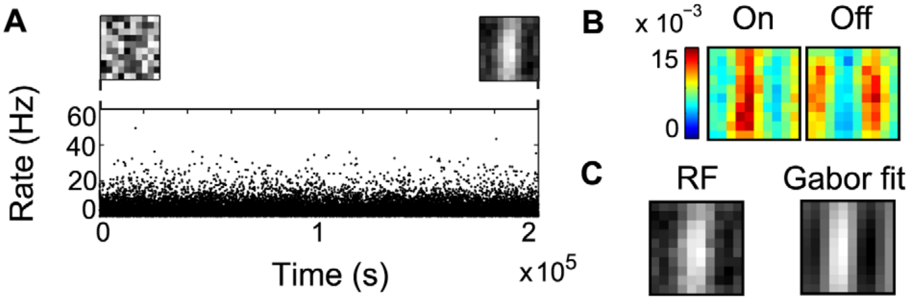

Figure 6. Learning a Gabor-like receptive field. (A) Evolution of the neuronal activity during learning, (B) Learned weights corresponding to the inputs from the on and off populations, (C) The receptive field learned by the neuron, and its l.m.s. Gabor fit.
doi:10.1371/journal.pcbi.1000757.g006

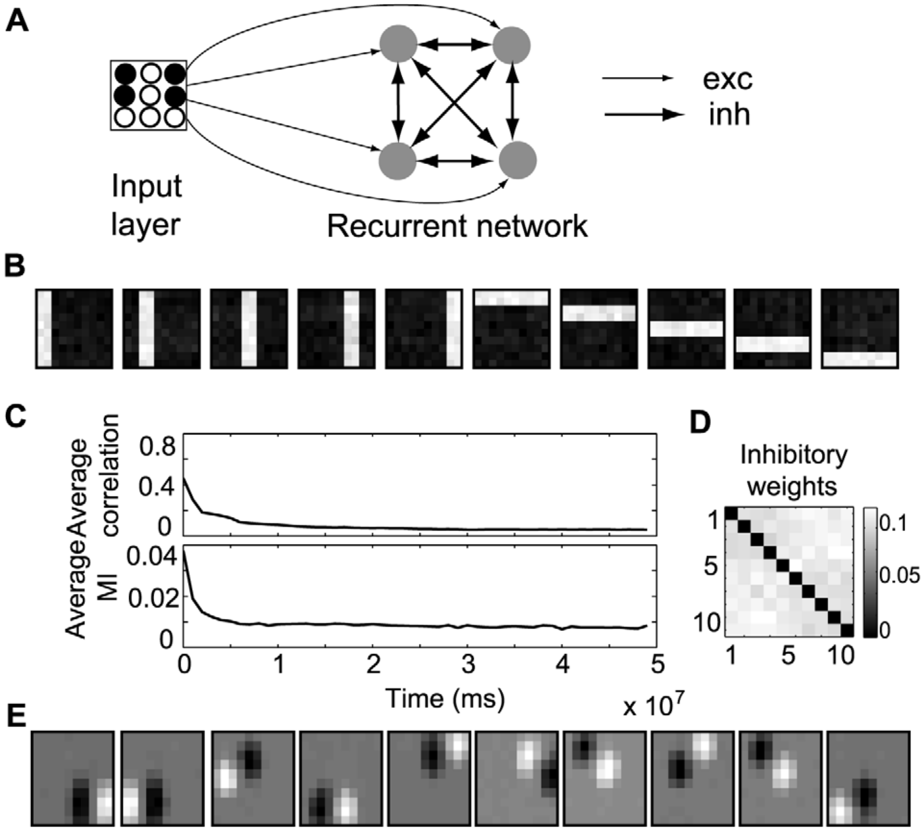

Figure 7. Learning multiple ICs. (A) Overview of the network structure: all neurons receive signals from the input layer and are recurrently connected by all-to-all inhibitory synapses, (B) A set of receptive fields learned for the bars problem, (C) Evolution of the mean correlation coefficient and mutual information in time, both computed by dividing the neuron output in bins of width 1000 s and estimating C and MI^* for each bin, (D) Learned inhibitory lateral connections, (E) A set of receptive fields learned for natural image patches.
doi:10.1371/journal.pcbi.1000757.g007

neuron (see Methods). With the same setup as before and the new IP rule, no bar is learned and the output distribution is Gaussian, with a small standard deviation around the target value μ (Fig. 9A). However, after additional parameter tuning, a bar can sometimes be learned, as shown in Fig. 9B. In this case, the final output distribution is highly kurtotic, due to the receptive field. The outcome depends on the variance of the total input, which has to be large enough to start the learning process (variance was regulated by the parameter w_{tot}, see Methods). Most importantly, this dependence on model parameters shows that regulating the mean of the output distribution is not sufficient for reliably learning a bar and higher order moments need to be considered as well.

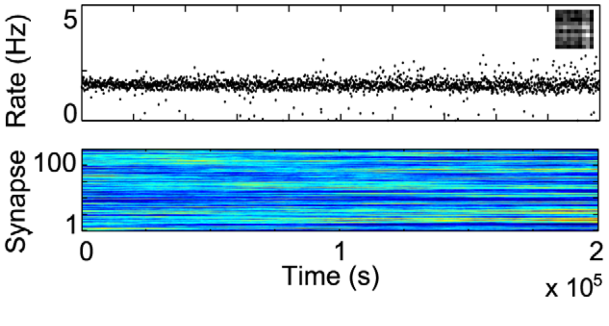

Figure 8. IP is critical for learning. Evolution of the receptive field for a neuron with a fixed gain function, given by the final parameters obtained after learning in the previous bars experiment. A bar cannot be learned in this case.
doi:10.1371/journal.pcbi.1000757.g008

Interaction between IP and STDP

A good starting point for elucidating the mechanism by which the interaction between STDP and IP facilitates the discovery of an independent component is our initial problem of a single unit receiving a two dimensional input. We have previously shown in simulations that for a bounded, whitened, two dimensional input the weight vector tends to rotate towards the heavy-tailed direction in the input [16]. Here, we extend these results both analytically and in simulations. Our analysis focuses on the theoretical formulation of zero mean, unit variance inputs used for the demixing problem before and is restricted to expected changes in weights given the input and output firing rates, ignoring the time of individual spikes.

We report here only the main results of these experiments, while a detailed description is provided as supplemental information (see Text S3). Firstly, for conveniently selected pairs of input distributions, it is possible to show analytically that the weight vector rotates towards the heavy-tailed direction in the input, under the assumption that IP adaptation is faster than synaptic learning (previously demonstrated numerically in [16]). Secondly, due to the IP rule, weight changes mostly occur on the tail of the output distribution and are significantly larger for the heavy-tailed input. Namely, IP focuses learning to the heavy tailed direction in

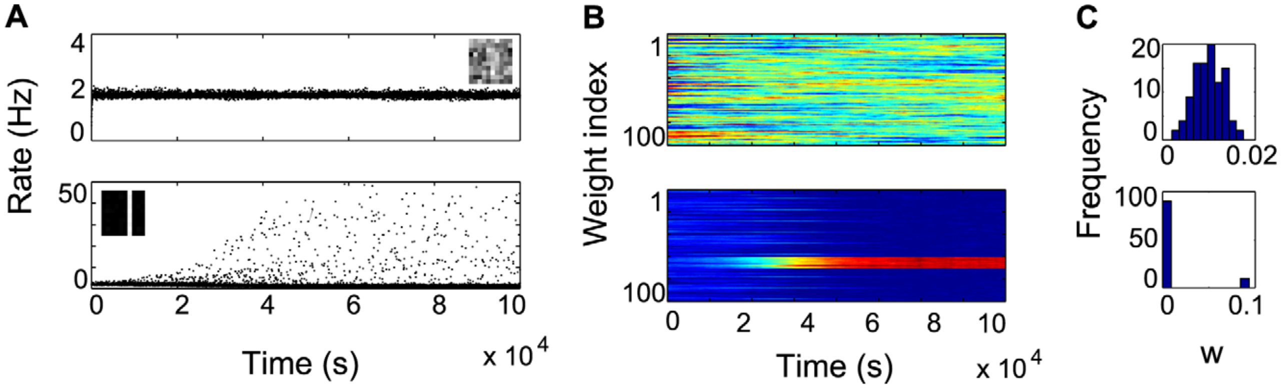

Figure 9. Mean firing constraint is not sufficient for reliable learning. (A) Evolution of neuron activation for a neuron with a gain function regulated by a simplified IP rule, which adjusts r_0 to maintain the same mean average firing μ. $w_{\mathrm{tot}}=2.5$ or $w_{\mathrm{tot}}=10$, in the first and second row, respectively. Inset illustrates final receptive field for each case. (B) Corresponding evolution of weights and (C) their final distribution.
doi:10.1371/journal.pcbi.1000757.g009

the input. When several inputs are supergaussian, the learning procedure results in the maximization of the output kurtosis, independent of the shape of the input distributions. Most importantly, we show that, for simple problems when a solution can be obtained by nonlinear PCA, our IP rule significantly speeds up learning of an independent component.

One way to understand these results could be in terms of nonlinear PCA theory. Given that for a random initial weight vector, the total input distribution is close to Gaussian, in order to enforce a sparse output, the IP has to change the transfer function in a way that 'hides' most of the input distribution (for example by shifting u_0 somewhere above the mean of the Gaussian). As a result, the nonlinear part of the transfer function will cover the 'visible' part of the input distribution, facilitating the discovery of sparse inputs by a mechanism similar to nonlinear PCA. In this light, IP provides the means to adapt the transfer function in a way that makes the nonlinear PCA particularly efficient.

Lastly, from an information-theoretic perspective, our approach can be linked to previous work on maximizing information transmission between neuronal input and output by optimizing synaptic learning [23]. This synaptic optimization procedure was shown to yield a generalization of the classic BCM rule [22]. We can show that, for a specific family of STDP implementations, which have a quadratic dependence on postsynaptic firing, IP effectively acts as a sliding threshold for BCM learning (see Text S4).

Discussion

Although ICA and related sparse coding models have been very successful in describing sensory coding in the cortex, it has been unclear how such computations can be realized in networks of spiking neurons in a biologically plausible fashion. We have presented a network of stochastically spiking neurons that performs ICA-like learning by combining different forms of plasticity. Although this is not the only attempt at computing ICA with spiking neurons, in previous models synaptic changes were not local, depending on the activity of neighboring neurons within a population [6,7]. In this light, our model is, to our knowledge, the first to offer a mechanistic explanation of how ICA-like computation could arise by biologically plausible learning mechanisms.

In our model, IP, STDP and synaptic scaling interact to give rise to robust receptive field development. This effect does not depend on a particular implementation of STDP, but it does require an IP mechanism which enforces a sparse output distribution. Although there are very good theoretical arguments why this should be the case [11,13,16], the experimental evidence supporting this assumption is limited [12]. A likely explanation for this situation is the fact that it is difficult to map the experimentally observable output spikes into a probability of firing. Spike count estimates cannot be used directly, as they critically depend on the bin size. Additionally, the inter-spike interval (ISI) of an inhomogeneous Poisson process with exponentially distributed mean $\lambda(t)$ is indistinguishable from the ISI of a homogeneous Poisson distribution with mean $\lambda=E(\lambda(t))$. Hence, more complex statistical analyses are required for disentangling the two (see [35]).

From a computational perspective, our approach is reminiscent of several by-now classic ICA algorithms. As mentioned before, IP enforces the output distribution to be heavy-tailed, like in sparse coding [9]. Our model also shares conceptual similarities to InfoMax [29], which attempts to maximize output entropy (however, at the population level) by regulating the weights and a neuron threshold parameter. Maximizing information transmission between pre- and post-synaptic spike trains under the constraint of a fixed mean postsynaptic firing rate links our method to previous work on synaptic plasticity. A spike-based synaptic rule optimizing the above criterion [23] yields a generalization of the BCM rule [22], a powerful form of learning, which is able to discover heavy-tailed directions in the input [36,37] and to learn Gabor receptive fields [38] in linear neurons. We have shown that, sliding threshold BCM can be viewed as a particular case of IP learning, for a specific family of STDP models.

It is interesting to think of the mechanism presented here in relation to projection pursuit [39], which tries to find good representations of high-dimensional spaces by projecting data on a lower dimensional space. The algorithm searches for interesting projection directions, a typical measure of interest being the non-Gaussianity of the distribution of data in the lower dimensional space. The difference here is that, although we do not explicitly define a contrast function maximizing kurtosis or other similar measure, our IP rule implicitly yields highly kurtotic output distributions. By sparsifying the neuron output, IP guides the synaptic learning towards the interesting (i.e. heavy-tailed) directions in the input.

From a different perspective, we can relate our method to nonlinear PCA. It is known that, for zero mean whitened data, nonlinear Hebbian learning in a rate neuron can successfully capture higher order correlations in the input [40,41]. Moreover, it has been suggested that the precise shape of the Hebbian nonlinearity can be used for optimization purposes, for example for incorporating prior knowledge about the sources' distribution

[40]. IP goes one step further in this direction, by adapting the transfer function online, during learning. From a biological perspective, there are some advantages in adapting the neuron's excitability during computation. Firstly, IP speeds up the nonlinear decorrelation of inputs. Secondly, the system gains great robustness to changes in parameters (as demonstrated in Text S2). Additionally, IP regulation plays a homeostatic role, making constraints on the input mean or second order statistics unnecessary. In the end, all the methods we have mentioned are closely related and, though conceptually similar, our approach is another distinct solution.

Our previous work was restricted to the a rate model neuron [16]. Beyond translating our results to a spiking neuron model, we have shown here that similar principles can be applied when information is encoded as spike-spike correlations, where a model relying just on firing rates would fail. It is a interesting challenge for future work to further investigate the exact mechanisms of receptive field development for different types of input encoding.

Methods

Neuron model

We consider a stochastically spiking neuron with refractoriness [23]. The model defines the neuron's instantaneous probability of firing as a function of the current membrane potential and the refractory state of the neuron,which depends on the time since its last spike. More specifically, the membrane potential is computed as $u(t) = u_{\mathrm{r}} + \sum_{j,f} w_{\mathrm{j}} \varepsilon(t - t_{\mathrm{j}}^{f})$, where $u_{\mathrm{r}} = -70\,\mathrm{mV}$ is the resting potential, while the second term represents the total incoming drive to the neuron, computed as the linear summation of post-synaptic potentials evoked by incoming spikes. Here, w_{j} gives the strength of synapse j, t_{j}^{f} is the time of a presynaptic spike, and $\varepsilon(t - t_{\mathrm{j}}^{f})$ the corresponding evoked post-synaptic potential, modeled as a decaying exponential, with time constant $\tau = 10\,\mathrm{ms}$ (for GABA-ergic synapses, $\tau = 20\,\mathrm{ms}$) and amplitude $1\,\mathrm{mV}$.

The refractory state of the neuron, with values in the interval [0,1], is defined as a function of the time of the last spike $\hat{t}$, namely:

$$R(t) = \frac{(t - \hat{t} - \tau_{\mathrm{abs}})^2}{\tau_{\mathrm{refr}}^2 + (t - \hat{t} - \tau_{\mathrm{abs}})^2} \cdot \Theta(t - \hat{t} - \tau_{\mathrm{abs}}),$$

where $\tau_{\mathrm{abs}} = 3\,\mathrm{ms}$ gives the absolute refractory period, $\tau_{\mathrm{refr}} = 10\,\mathrm{ms}$ is the relative refractory period and $\Theta(\cdot)$ is the Heaviside function.

The probability $\rho(t)$ of the stochastic neuron firing at time t is given as a function of its membrane potential and refractory state [23] $\rho(t) = 1 - e^{-g(u(t))R(t)\Delta t} \approx g(u(t))R(t)\Delta t$, where $\Delta t = 10^{-3}\,\mathrm{s}$ is the time step of integration, and $g(u(t))$ is a gain function, defined as: $g(u) = r_0 \log\left(1 + e^{\frac{u - u_0}{u_\alpha}}\right)$. Here r_0 , u_0 and u_α are model parameters, whose values are adjusted by intrinsic plasticity, as described below.

Intrinsic plasticity

Our intrinsic plasticity model attempts to maximize the mutual information between input and output, for a fixed energy budget [16,42]. More specifically, it induces changes in neuronal excitability that lead to an exponential distribution of the instantaneous firing rate of the neuron [13]. The specific shape of the output distribution is justified from an information theoretic perspective, as the exponential distribution has maximum entropy for a fixed mean. This is true for distributions defined on the interval $[0,\infty)$, but, under certain assumptions, can be a good approximation for the case where the interval is bounded, as it happens in our model due to the neuron's refractory period (see below). Optimizing information transmission under the constraint of a fixed mean is equivalent to minimizing the Kullback-Leibler divergence between the neuron's firing rate distribution and that of an exponential with mean μ:

$$\begin{aligned} D = d(p_{\mathrm{neuron}} \| p_{\exp}) &= \int f_Y(y) \log\left(\frac{f_Y(y)}{\frac{1}{\mu} e^{-y/\mu}}\right) dy \\ &= -H(Y) + \frac{1}{\mu} E(Y) + \log(\mu), \end{aligned}$$

with $y = g(u)$ and $H(\cdot)$ denoting the entropy and $E(\cdot)$ the expected value. Note that the above expression assumes that the instantaneous firing rate of the neuron is proportional to $g(u)$, that is that $R(t) \approx 1$. When taking into account the refractory period of the neuron, which imposes an upper-bound $R = \frac{1}{\tau_{\mathrm{abs}}}$ on the output firing rate, the maximum entropy distribution for a specific mean $\mu \leq R$ is a truncated exponential [43]. The deviation between the optimal exponential for the infinite and the bounded case depends on the values of μ and R, but it is small in cases in which $\mu \ll R$. Hence, our approximation is valid as long as the instantaneous firing rate μ is significantly lower than $1/\tau_{\mathrm{abs}}$, that is when the mean firing rate of the neuron is small. In our case, we restrict $\mu \leq 10\,\mathrm{Hz}$. If not otherwise stated, all simulations have $\mu = 2\,\mathrm{Hz}$. Note also that the values considered here are in the range of firing rates reported for V1 neurons [44].

Computing the gradient of D for r_0, u_0 and u_α, and using stochastic gradient descent, the optimization process translates into the following update rules [42]:

$$\begin{aligned} \Delta r_0 &= \frac{\eta_{\mathrm{IP}}}{r_0}\left(1 - \frac{g}{\mu}\right), \\ \Delta u_0 &= \frac{\eta_{\mathrm{IP}}}{u_\alpha}\left(\left(1 + \frac{r_0}{\mu}\right)\left(1 - e^{-\frac{g}{r_0}}\right) - 1\right), \\ \Delta u_\alpha &= \frac{\eta_{\mathrm{IP}}}{u_\alpha}\left(\frac{u - u_0}{u_\alpha}\left(\left(1 + \frac{r_0}{\mu}\right)\left(1 - e^{-\frac{g}{r_0}}\right) - 1\right) - 1\right). \end{aligned}$$

Here, $\eta_{\mathrm{IP}} = 10^{-5}$ is a small learning rate. Here, the instantaneous firing rate is assumed to be directly accessible for learning. Alternatively, it could be estimated based on the recent spike history.

Additionally, as a control, we have considered a simplified rule, which adjusts a single transfer function parameter in order to maintain the mean firing rate of the neuron to a constant value μ. More specifically, a low-pass-filtered version of the neuron firing rate is used to estimate the current mean firing rate of the neuron $\frac{d\bar{g}}{dt} = -\frac{\bar{g}}{\tau_{\mathrm{m1}}} + \delta(t - t_{\mathrm{f}}^{f})$, where δ is the Dirac function and t_{f}^{f} is the time of firing of the post-synaptic neuron and $\tau_{\mathrm{m1}} = 100\mathrm{ms}$. Based on this estimate, the value of the parameter r_0 is adjusted as $r_0 = r_0 - \eta_{\mathrm{m1}}(\bar{g} - \mu)$. Here, μ is the goal mean firing rate, as before and η_{m1} is a learning rate, set such that, for a fixed Gaussian input distribution, convergence is reached as fast as for our IP rule described before ($\eta_{\mathrm{m1}} = 10^{-4}$).

Synaptic learning

The STDP rule implemented here considers only nearest-neighbor interactions between spikes [24]. The change in weights is determined by:

$$\Delta w_{\mathrm{ij}} = \begin{cases} A_+ e^{-\frac{\Delta t}{\tau_+}}, & \text{if} \quad \Delta t > 0 \\ A_- e^{\frac{\Delta t}{\tau_-}}, & \text{if} \quad \Delta t < 0 \end{cases},$$

where $A_{+/-}$ is the amplitude of the STDP change for potentiation and depression, respectively (default values $A_+ = 1.03 \cdot 10^{-4}$ and $A_- = -0.51 \cdot 10^{-4}$), $\tau_{+/-}$ are the time scales for potentiation and depression ($\tau_+ = 12\,\mathrm{ms}$, $\tau_- = 38\,\mathrm{ms}$; for learning spike-spike correlations $\tau_+ = 10\ \mathrm{ms}$)[19], and $\Delta t = t^{\mathrm{post}} - t^{\mathrm{pre}}$ is the time difference between the firing of the pre- and post-synaptic neuron. For the lateral inhibitory connections, the STDP learning is faster, namely $A^{\mathrm{inh}}_{+/-}/A^{\mathrm{exc}}_{+/-} = 10$. In all cases, weights are always positive and clipped to zero if they become negative.

This STDP implementation is particularly interesting as it can be shown that, under the assumption of uncorrelated or weakly correlated pre- and post-synaptic Poisson spike trains, it induces weight changes similar to a BCM rule, namely [24]:

$$\Delta w \approx xy\left(\frac{A_+}{\tau_+^{-1} + y} + \frac{A_-}{\tau_-^{-1} + y}\right),$$

where x and y are the firing rates of the pre- and post-synaptic neuron, respectively.

For the above expression, the fixed BCM threshold can be computed as:

$$v = -\frac{A_+/\tau_- + A_-/\tau_+}{A_+ + A_-},$$

which is positive when potentiation dominates depression on the short time scale, while, overall, synaptic weakening is larger than potentiation:

$$A_+ > |A_-|,$$

$$|A_-|\tau_- > A_+\tau_+.$$

In some experiments we also consider the classical case of additive all-to-all STDP [26], which acts as simple Hebbian learning, the induced change in weight being proportional to the product of the pre- and post-synaptic firing rates (see [24] for comparison of different STDP implementations). The parameters used in this case are: $A_+ = 8.33 \cdot 10^{-6}$, $A_- = -2.63 \cdot 10^{-6}$ and the same time constants as for the nearest neighbor case. Additionally, the simple triplets STDP model used as an alternative BCM-like STDP implementation is described in Text S4.

Synaptic scaling

As in approaches which directly maximize kurtosis or similar measures [16,30,40], the weight vector is normalized: $\sum_{\mathrm{i}} w_{\mathrm{i}} = w_{\mathrm{tot}}$, with $w_{\mathrm{tot}} = 2.5$, with weights being always positive. This value is arbitrary, as it represents a scaling factor of the total current, which can be compensated for by IP. It was selected in order to keep the final parameters close to those in [23]. Additionally, for the natural image patches the normalization was done independently for the on- and off- populations, using the same value for w_{tot} in each case.

In a neural population, the same normalization is applied for the lateral inhibitory connections. As before, weights do not change sign and are constrained by the L_1 norm: $\sum_{\mathrm{i}} w^{\mathrm{inh}}_{\mathrm{i}} = w^{\mathrm{inh}}_{\mathrm{tot}}$, with $w^{\mathrm{inh}}_{\mathrm{tot}} = -12$.

Currently, the normalization is achieved by dividing each weight by $\frac{\sum_{\mathrm{i}} w_{\mathrm{i}}}{w_{\mathrm{tot}}}$, after the presentation of each sample. Biologically, this operation would be implemented by a synaptic scaling mechanism, which multiplicatively scales the synaptic weights to preserve the average input drive received by the neuron [20].

Setup for experiments

In all experiments, excitatory weights were initialized at random from the uniform distribution and normalized as described before. The transfer function was initialized to the parameters in [23] ($r_0 = 11$ Hz, $u_0 = -65$ mV, $u_\alpha = 2$ mV). Unless otherwise specified, all model parameters had the default values defined in the corresponding sections above. For all spike-based experiments, each sample was presented for a time interval $T = 100\mathrm{ms}$, followed by the weight normalization.

For the experiments involving the rate-based model and a two-dimensional input, each sample was presented for one time step and the learning rates for IP and Hebbian learning were $\eta_{\mathrm{IP}} = 10^{-4}$ and $\eta_{\mathrm{syn}} = 10^{-7}$, respectively. In this case, the weight normalization procedure can influence the final solution. Namely, positive weights with constant L_1 norm always yield a weight vector in the first quadrant, but this limitation can be removed by a different normalization, which keeps the L_2 norm of the vector constant ($|w|_{\mathrm{L}_2} = 1$).

For the demixing problem, the input was generated as described for the rate-based scenario above. After the rectification, the firing rates of the input on the on- and off- channels were scaled by a factor of 20, to speed up learning. After convergence, the total weight of each channel was estimated as the sum of individual weights corresponding to that input. The resulting four-dimensional weight vector was projected back to the original two-dimensional input space using: $|w| = \max(w_{\mathrm{on}}, w_{\mathrm{off}})$, with a sign given by that of the channel with maximum weight (positive for $w_{\mathrm{on}} > w_{\mathrm{off}}$, negative otherwise). This procedure results by a minimum error projection of the weight vector onto the subspace defined by the constraint $w_{\mathrm{on}} \cdot w_{\mathrm{off}} = 0$, see Text S1 for details.

For all variants of the bars problem, the input vector was normalized to $|x| = N$, with $|\cdot|$ defining the L_1 norm, as in [45]. Inputs were encoded using firing rates with mean $f_{\mathrm{bkgnd}} + x_{i,j} \cdot f_{\mathrm{max}}$, where $f_{\mathrm{bkgnd}} = 0.1\mathrm{Hz}$ is the frequency of a background pixel and $f_{\mathrm{max}} = 100\mathrm{Hz}$ gives the maximum input frequency, corresponding to a sample containing a single bar in the original bars problem.

When using the correlation-based encoding, all inputs had the same mean firing rate ($f = 25\mathrm{Hz}$). Inputs corresponding to pixels in the background were uncorrelated, while inputs belonging to bars were all pairwise correlated, with a correlation coefficient $C = 0.75$. Poisson processes with such correlation structure can be generated in a computationally efficient fashion by using dichotomous Gaussian distributions [46].

When learning Gabor receptive fields, images from the van Hateren database [31] were convolved with a difference-of-gaussians filter with center and surround widths of 1.0 and 1.2 pixels, respectively. Random patches of size 10×10 were selected from various positions in the images. Patches having very low contrast were discarded. The individual input patches were normalized to zero mean and unit variance, similar to the processing in [45]. The rectified values of the resulting image were mapped into a firing frequency for an on- and off-input population ($f_{i,j}^{\text{on/off}} = f_{\text{bkgnd}} + x_{i,j}^{\text{on/off}} \cdot f_{\max}$) and, as before, samples were presented for a duration T.

For a neuronal population, input-related parameters were as for the single component, but with $\mu = 5\text{Hz}$, to speed up learning. The initial parameters of the neuron transfer function were uniformly distributed around the default values mentioned above, with variance 0.1, 5, and 0.2 for r_0, u_0, and u_α, respectively. Additionally, the inhibitory weights were initialized at random, with no self-connections, and normalized as described before. The mutual information (MI), estimated within a window of 1000 s, was computed as $\text{MI}^*(X,Y) = \dfrac{\text{MI}(X,Y)}{H(X)+H(Y)}$, with $H(\cdot)$ denoting the entropy (see [45]), applied for the average firing rate of the neurons for each input sample.

Supporting Information

Text S1 Receptive field estimation for the spike-based demixing problem
Found at: doi:10.1371/journal.pcbi.1000757.s001 (0.04 MB PDF)

Text S2 Parameter dependence for learning one IC
Found at: doi:10.1371/journal.pcbi.1000757.s002 (0.07 MB PDF)

Text S3 Learning with a two-dimensional input
Found at: doi:10.1371/journal.pcbi.1000757.s003 (0.53 MB PDF)

Text S4 A link to BCM
Found at: doi:10.1371/journal.pcbi.1000757.s004 (0.15 MB PDF)

Acknowledgments

We would like to thank Jörg Lücke and Claudia Clopath and for fruitful discussions. Additionally, we thank Jörg Lücke for providing the code for the Gabor l.m.s. fit and to Felipe Gerhard for his help with the natural images preprocessing.

Author Contributions

Conceived and designed the experiments: CS JT. Performed the experiments: CS. Analyzed the data: CS. Wrote the paper: CS JT. Developed the mathematical model used for implementing intrinsic plasticity in the paper: PJ.

References

1. Barlow H (2001) Redundancy reduction revisited. Network 12: 241–253.
2. Simoncelli E, Olshausen B (2001) Natural image statistics and neural representations. Annu Rev Neuroscience 24: 1193–1216.
3. Simoncelli E (2003) Vision and the statistics of the visual environment. Curr Op Neurobiol 13: 144–149.
4. Olshausen B, Field D (2004) Sparse coding of sensory inputs. Curr Op Neurobiol 14: 481–487.
5. Hyvärinen A, Karhunen J, Oja E (2001) Independent Component Analysis. Wiley Series on Adaptive and Learning Systems for Signal Processing, Communication and Control.
6. Klampfl S, Legenstein R, Maass W (2009) Spiking neurons can learn to solve information bottleneck problems and extract independent components. Neural Computation 21: 911–959.
7. Parra L, Beck J, Bell A (2009) On the maximization of information flow between spiking neurons. Neural Comp 21: 1–19.
8. Clopath C, Büsing L, Vasilaki E, Gerstner W (2010) Connectivity reflects coding: a model of voltage-based stdp with homeostasis. Nature Neuroscience.
9. Olshausen B, Field D (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research 37: 3311–3325.
10. Olshausen B, Field D (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381: 607–609.
11. Lennie P (2003) The cost of neural computation. Current Biology 13: 493–497.
12. Baddeley R, Abbott L, Booth M, Sengpiel F, Freeman T, et al. (1997) Responses of neurons in primary and inferior temporal visual cortices to natural scenes. Proceedings Biological Sciences 264: 1775–1783.
13. Stemmler M, Koch C (1999) How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. Nature Neuroscience 2: 521–527.
14. Zhang W, Linden D (2003) The other side of the engram: experience-dependent changes in neuronal intrinsic excitability. Nature Reviews Neuroscience 4: 885–900.
15. Cudmore R, Turrigiano G (2004) Long-term potentiation of intrinsic excitability in LV visual cortical neurons. J Neurophysiol 92: 341–348.
16. Triesch J (2007) Synergies between intrinsic and synaptic plasticity mechanisms. Neural Computation 19: 885–909.
17. Gerstner W, Kempter R, van Hemmen J, Wagner H (1996) A neuronal learning rule for sub-millisecond temporal coding. Nature 383: 76–78.
18. Markram H, Lübke J, Frotscher M, Sackmann B (1997) Regulation of synaptic efficacy by coincidence of postsynaptic APS and EPSPS. Science 275: 213–215.
19. Bi G, Poo M (1998) Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. J Neurosci 18: 10464–10472.
20. Turrigiano G, Nelson S (2004) Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience 5: 97–107.
21. Foldiák P (1990) Forming sparse representations by local anti-Hebbian learning. Biological Cybernetics 64: 165–170.
22. Bienenstock E, Cooper L, Munro P (1982) Theory for the development of neuron selectivity: Orientation specificity and binocular interactions in visual cortex. Journal of Neuroscience 2: 32–48.
23. Toyoizumi T, Pfister J, Aihara K, Gerstner W (2005) Generalized Bienenstock-Cooper-Munro rule for spiking neurons that maximizes information transmission. PNAS 102: 5239–5244.
24. Izhikevich E, Desai N (2003) Relating STDP to BCM. Neural Computation 15: 1511–1523.
25. Martinez-Conde S, Macknik S, Hubel D (2004) The role of fixational eye movements in visual perception. Nature Reviews Neuroscience 5: 229–240.
26. Song S, Miller K, Abbott L (2000) Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nature Neurosci 3: 919–926.
27. Pfister J, Gerstner W (2006) Triplets of spikes in a model of spike timing-dependent plasticity. Journal of Neuroscience 26: 9673–9682.
28. Lücke J, Sahani M (2008) Maximal causes for non-linear component extraction. Journal of Machine Learning Research 9: 1227–1267.
29. Bell A, Sejnowski T (1997) The independent components of natural scenes are edge filters. Vision Research 37: 3327–3338.
30. Hyvärinen A (1999) Survey on Independent Component Analysis. Neural Computing Surveys 2: 94–128.
31. van Hateren J (1998) Independent component filters of natural images compared with simple cells in primary visual cortex. Proceedings of the Royal Society B: Biological Sciences 265: 359–366.
32. Falconbridge M, Stamps R, Badcock D (2006) A simple Hebbian/anti-Hebbian network learns the sparse, independent components of natural images. Neural Computation 18: 415–429.
33. Weber C, Triesch J (2008) A sparse generative model of V1 simple cells with intrinsic plasticity. Neural Computation 20: 1261–1284.
34. Lücke J (2009) Receptive field self-organization in a model of the fine-structure in V1 cortical columns. Neural Computation 21: 2805–2845.
35. Cox D (1955) Some statistical methods connected with series of events. Journal of the Royal Statistical Society, Series B 17: 129–164.
36. Intrator N, Cooper L (1992) Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions. Neural Networks 5: 3–17.
37. Intrator N (1998) Neuronal goals: efficient coding and coincidence detection. In: Springer-Verlag, editor (1998) ICONIP Hong Kong: Progress in Neural Information Processing, volume 1. pp 29–34.
38. Blais B, Intrator N, Shouval H, Cooper L (1998) Receptive field formation in natural scene environments: comparison of single cell learning rules. Neural Computation 10: 1797–1813.
39. Huber P (1985) Projection pursuit. The Annals of Statistics 13: 435–475.
40. Hyvärinen A (1997) One-unit contrast functions for independent component analysis: a statistical analysis. In: Neural Networks for Signal Processing. pp 388–397.
41. Oja E (1997) The nonlinear PCA learning rule in independent component analysis. Neurocomputing 17: 25–46.
42. Joshi P, Triesch J (2009) Rules for information-maximization in spiking neurons using intrinsic plasticity. In: Proc. IJCNN. pp 1456–1461.
43. Kapur N (1990) Maximum Entropy Models in Science and Engineering Wiley-Interscience.
44. Olshausen B, Simoncelli E (2001) Natural image statistics and neural representation. Annu Rev Neuroscience 24: 1193–1216.
45. Butko N, Triesch J (2007) Learning sensory representations with intrinsic plasticity. Neurocomputing 70.
46. Macke J, Berens P, Ecker A, Tolias A, Bethge M (2009) Generating spike trains with specified correlation coefficients. Neural Comp 21: 397–423.