LETTER Communicated by Aapo Hyvarinen

Role of Homeostasis in Learning Sparse Representations

Laurent U. Perrinet
Laurent.Perrinet@incm.cnrs-mrs.fr
Institut de Neurosciences Cognitives de la Méditerranée, CNRS/University of Provence, 13402 Marseille Cedex 20, France

Neurons in the input layer of primary visual cortex in primates develop edge-like receptive fields. One approach to understanding the emergence of this response is to state that neural activity has to efficiently represent sensory data with respect to the statistics of natural scenes. Furthermore, it is believed that such an efficient coding is achieved using a competition across neurons so as to generate a sparse representation, that is, where a relatively small number of neurons are simultaneously active. Indeed, different models of sparse coding, coupled with Hebbian learning and homeostasis, have been proposed that successfully match the observed emergent response. However, the specific role of homeostasis in learning such sparse representations is still largely unknown. By quantitatively assessing the efficiency of the neural representation during learning, we derive a cooperative homeostasis mechanism that optimally tunes the competition between neurons within the sparse coding algorithm. We apply this homeostasis while learning small patches taken from natural images and compare its efficiency with state-of-the-art algorithms. Results show that while different sparse coding algorithms give similar coding results, the homeostasis provides an optimal balance for the representation of natural images within the population of neurons. Competition in sparse coding is optimized when it is fair. By contributing to optimizing statistical competition across neurons, homeostasis is crucial in providing a more efficient solution to the emergence of independent components.

1 Introduction

The central nervous system is a dynamic, adaptive organ that constantly evolves to provide optimal decisions for interacting with the environment. The early visual pathways provide a powerful system for probing and modeling these mechanisms. For instance, it is observed that edge-like receptive fields emerge in simple cell neurons from the input layer of the primary visual cortex of primates (Chapman & Stryker, 1992). The development of cortical cell orientation tuning is an activity-dependent process, but it is still largely unknown how neural computations implement this

type of unsupervised learning mechanisms. A popular view is that such a population of neurons operates so that relevant sensory information from the retino-thalamic pathway is transformed (or "coded") efficiently. Such efficient representation will allow decisions to be taken optimally in higher-level layers or areas (Atick, 1992; Barlow, 2001). It is believed that this is achieved through lateral interactions that remove redundancies in the neural representation, that is, when the representation is sparse (Olshausen & Field, 1996). A representation is sparse when each input signal is associated with a relatively small number of simultaneously activated neurons within the population. For instance, orientation selectivity of simple cells is sharper than the selectivity that would be predicted by linear filtering. As a consequence, representation in the orientation domain is sparse and allows higher processing stages to better segregate edges in the image (Field, 1994). Sparse representations are observed prominently with cortical response to natural stimuli, that is, to behaviorally relevant sensory inputs (Vinje & Gallant, 2000; DeWeese, Wehr, & Zador, 2003; Baudot et al., 2004). This reflects the fact that at the learning timescale, coding is optimized relative to the statistics of natural scenes. The emergence of edge-like simple cell receptive fields in the input layer of the primary visual cortex of primates may thus be considered as a coupled coding and learning optimization problem. At the coding timescale, the sparseness of the representation is optimized for any given input, while at the learning timescale, synaptic weights are tuned to achieve on average optimal representation efficiency over natural scenes.

Most existing models of unsupervised learning aim at optimizing a cost defined on prior assumptions on representation's sparseness. These sparse learning algorithms have been applied for both images (Fyfe & Baddeley, 1995; Olshausen & Field, 1996; Zibulevsky & Pearlmutter, 2001; Perrinet, 2004; Rehn & Sommer, 2007; Doi, Balcan, & Lewicki, 2007) and sounds (Lewicki & Sejnowski, 2000; Smith & Lewicki, 2006). For instance, learning is accomplished in SparseNet (Olshausen & Field, 1996) on patches taken from natural images as a sequence of coding and learning steps. First, sparse coding is achieved using a gradient descent over a convex cost derived from a sparse prior probability distribution function of the representation. At this step of the learning, it is performed using the current state of the dictionary of receptive fields. Then, knowing this sparse solution, learning is defined as slowly changing the dictionary using Hebbian learning. In general, the parameterization of the prior has major impacts on results of the sparse coding, and thus on the emergence of edge-like receptive fields, and requires proper tuning. In fact, the definition of the prior corresponds to an objective sparseness and does not always fit the observed probability distribution function of the coefficients. In particular, this could be a problem during learning if we use the cost to measure representation efficiency for this learning step. An alternative is to use a more generic L_0 norm sparseness by simply counting the number of nonzero coefficients. It was found that by using an algorithm like Matching Pursuit, the learning algorithm could

provide results similar to SparseNet, but without the need of parametric assumptions on the prior (Perrinet, Samuelides, & Thorpe, 2003; Perrinet, 2004; Smith & Lewicki, 2006; Rehn & Sommer, 2007). However, we observed that this class of algorithms could lead to solutions corresponding to a local minimum of the objective function. Some solutions seem as efficient as others for representing the signal but do not represent edge-like features homogeneously. In particular, during the early learning phase, some cells may learn "faster" than others. There is a need for a homeostasis mechanism that will ensure convergence of learning. The goal of this work is to study the specific role of homeostasis in learning sparse representations and to propose a homeostasis mechanism that optimizes the learning of an efficient neural representation.

To achieve this, we first formulate analytically the problem of representation efficiency in a population of sensory neurons (see section 2) and define the class of Sparse Hebbian Learning (SHL) algorithms. For the particular nonparametric L_0 norm sparseness, we show that sparseness is optimal when average activity within the neural population is uniformly balanced. Based on a previous implementation, Adaptive Matching Pursuit (AMP) (Perrinet et al., 2003; Perrinet, 2004), we define in section 3 a homeostatic gain control mechanism that we will integrate in a novel SHL algorithm. Finally, we compare in section 4 this novel algorithm with AMP and the state-of-the-art SparseNet method (Olshausen & Field, 1996). Using quantitative measures of efficiency based on constraints on the neural representation, we show the importance of the homeostasis mechanism in terms of representation efficiency. We conclude in section 5 by linking this original method with other sparse Hebbian learning schemes and how these may be united to improve our understanding of the emergence of edge-like simple cell receptive fields, drawing the bridge between structure (representation in a distributed network) and function (efficient coding).

2 Problem Statement

2.1 Definition of Representation Efficiency. In low-level sensory areas, the goal of neural computations is to generate efficient intermediate representations to allow efficient decision making. Classically, a representation is defined as the inversion of an internal generative model of the sensory world, that is, by inferring the sources that generated the input signal. Formally, as in Olshausen and Field (1997), we define a linear generative model (LGM) for describing natural, static, gray-scale images $\mathbf{I}$ (represented by column vectors of dimension L pixels), by setting a "dictionary" of M images (or "filters") as the $L \times M$ matrix $\Phi = \{\Phi_i\}_{1 \le i \le M}$. Knowing the associated sources as a vector of coefficients $\mathbf{a} = \{a_i\}_{1 \le i \le M}$, the image is defined using matrix notation as

$$\mathbf{I} = \Phi \mathbf{a} + \mathbf{n}, \tag{2.1}$$

where $\mathbf{n}$ is a decorrelated gaussian additive noise image of variance σ_n^2. The decorrelation of the noise is achieved by applying principal component analysis to the raw input images without loss of generality, since this preprocessing is invertible. Generally the dictionary Φ may be much larger than the dimension of the input space (that is, if $M \gg L$), and it is then said to be overcomplete. However, given an overcomplete dictionary, the inversion of the LGM leads to a combinatorial search, and typically there may exist many coding solutions $\mathbf{a}$ to equation 2.1 for one given input $\mathbf{I}$. The goal of efficient coding is to find, given the dictionary Φ and for any observed signal $\mathbf{I}$, the best representation vector—that is, as close as possible to the sources that generated the signal. It is therefore necessary to define an efficiency criterion in order to choose between these different solutions.

Using the LGM, we will infer the best coding vector as the most probable. In particular, from the physical synthesis of natural images, we know a priori that image representations are sparse: they are most likely generated by a small number of features relative to the dimension M of representation space. Similarly to Lewicki and Sejnowski (2000), this can be formalized in the probabilistic framework defined by the LGM (see equation 2.1) by assuming that we know the prior distribution of the coefficients a_i for natural images. The representation cost of $\mathbf{a}$ for one given natural image is then

$$\begin{aligned}\mathcal{C}(\mathbf{a} \mid \mathbf{I}, \Phi) &= -\log P(\mathbf{a} \mid \mathbf{I}, \Phi) \\ &= \log Z + \frac{1}{2\sigma_n^2}\|\mathbf{I} - \Phi\mathbf{a}\|^2 - \sum_i \log P(a_i \mid \Phi),\end{aligned} \tag{2.2}$$

where Z is the partition function, which is independent of the coding, and $\|\cdot\|$ is the L_2 norm in image space. This efficiency cost is measured in bits if the logarithm is of base 2, as we will assume without loss of generality thereafter. For any representation $\mathbf{a}$, the cost value corresponds to the description length (Rissanen, 1978). On the right-hand side of equation 2.2, the second term corresponds to the information from the image that is not coded by the representation (reconstruction cost) and thus to the information that can be at best encoded using entropic coding pixel by pixel (it is the log likelihood in Bayesian terminology). The third term, $S(\mathbf{a} \mid \Phi) = -\sum_i \log P(a_i \mid \Phi)$, is the representation or sparseness cost: it quantifies representation efficiency as the coding length of each coefficient of $\mathbf{a}$ independently that would be achieved by entropic coding knowing the prior. In practice, the sparseness of coefficients for natural images is often defined by an ad hoc parameterization of the prior's shape. For instance, the parameterization in Olshausen and Field (1997) yields the coding cost:

$$\mathcal{C}_1(\mathbf{a} \mid \mathbf{I}, \Phi) = \frac{1}{2\sigma_n^2}\|\mathbf{I} - \Phi\mathbf{a}\|^2 + \beta \sum_i \log\left(1 + \frac{a_i^2}{\sigma^2}\right), \tag{2.3}$$

where β corresponds to the prior's steepness and σ to its scaling (see Figure 13.2 from Olshausen, 2002). This choice is often favored because it results in a convex cost for which known numerical optimization methods such as conjugate gradient may be used.

A nonparametric form of sparseness cost may be defined by considering that neurons representing the vector **a** are either active or inactive. In fact, the spiking nature of neural information demonstrates that the transition from an inactive to an active state is far more significant at the coding timescale than smooth changes of the firing rate. This is, for instance, perfectly illustrated by the binary nature of the neural code in the auditory cortex of rats (DeWeese et al., 2003). Binary codes also emerge as optimal neural codes for rapid signal transmission (Bethge, Rotermund, & Pawelzik, 2003; Nikitin, Stocks, Morse, & McDonnell, 2009). With a binary event-based code, the cost is incremented only when a new neuron becomes active, regardless of the analog value. When it is stated that an active neuron carries a bounded amount of information of λ bits, an upper bound for the representation cost of neural activity on the receiver end is proportional to the count of active neurons, that is, to the L_0 norm:

$$C_0(\mathbf{a} \mid \mathbf{I}, \Phi) = \frac{1}{2\sigma_n^2} \|\mathbf{I} - \Phi\mathbf{a}\|^2 + \lambda \|\mathbf{a}\|_0. \tag{2.4}$$

This cost is similar with information criteria such as the AIC (Akaike, 1974) or distortion rate (Mallat, 1998). This simple nonparametric cost has the advantage of being dynamic. The number of active cells for one given signal grows in time with the number of spikes reaching the receiver (see the architecture of the model in Figure 1, left). But equation 2.4 defines a harder cost to optimize since the hard L_0 norm sparseness leads to a nonconvex optimization problem that is NP-complete with respect to the dimension M of the dictionary (Mallat, 1998).

2.2 Sparse Hebbian Learning. Given a sparse coding strategy that optimizes any representation efficiency cost as defined above, we may derive an unsupervised learning model by optimizing the dictionary Φ over natural scenes. On the one hand, the flexibility in the definition of the sparseness cost leads to a wide variety of proposed sparse coding solutions (for a review, see Pece, 2002) such as numerical optimization (Olshausen & Field, 1997; Lee, Battle, Raina, & Ng, 2007), nonnegative matrix factorization (Lee & Seung, 1999; Ranzato, Poultney, Chopra, & LeCun, 2007), or Matching Pursuit (Perrinet et al., 2003; Perrinet, 2004; Smith & Lewicki, 2006; Rehn & Sommer, 2007). On the other hand, these methods share the same LGM model (see equation 2.1), and once the sparse coding algorithm is chosen, the learning scheme is similar.

Indeed, after every coding sweep, the efficiency of the dictionary Φ may be increased with respect to equation 2.2. By using the online gradient

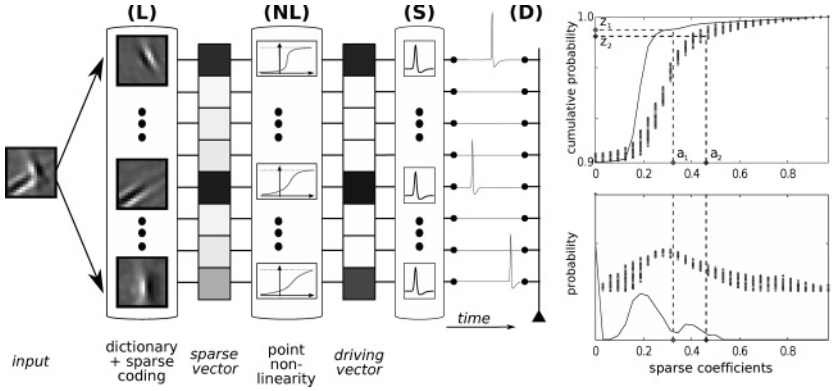

Figure 1: Simple neural model of sparse coding and role of homeostasis. (Left) We define the coding model as an information channel constituted by a bundle of linear/nonlinear spiking neurons. **(L)** A given input image patch is coded linearly by using the dictionary of filters Φ_i and transformed by sparse coding (such as Matching Pursuit) into a sparse vector $\mathbf{a}$. Each coefficient is transformed into a driving coefficient in the **(NL)** layer by using a point nonlinearity that **(S)** drives a generic spiking mechanism. **(D)** On the receiver end (e.g., in an efferent neuron), one may then estimate the input from the neural representation pattern. This decoding is progressive, and if we assume that each spike carries a bounded amount of information, the representation cost in this model increases proportionally with the number of activated neurons. (Right) However, for a given dictionary, the distribution of sparse coefficients a_i and hence the probability of a neuron's activation is in general not uniform. We show (lower panel) the log-probability distribution function and (upper panel) the cumulative distribution of sparse coefficients for a dictionary of edge-like filters with similar selectivity (dotted scatter) except for one filter, which was randomized (continuous line). This illustrates a typical situation that may occur during learning when some components did learn less than others. Since their activity will be lower, they are less likely to be activated in the spiking mechanism, and from the Hebbian rule, they are less likely to learn. Instead of comparing sparse coefficients with respect to a threshold (vertical dashed lines) when selecting an optimal sparse set for a given input, it should instead be done on the significance value z_i (horizontal dashed lines). In this particular case, the less selective neuron ($a_1 < a_2$) is selected by the homeostatic cooperation ($z_1 > z_2$). The role of homeostasis during learning is that even if the dictionary of filters is not homogeneous, the point nonlinearity in (NL) modifies sparse coding in (L) such that the probability of a neuron's activation is uniform across the population.

descent approach given the current sparse solution, learning may be achieved using $\forall i$,

$$\Phi_i \leftarrow \Phi_i + \eta a_i(\mathbf{I} - \Phi\mathbf{a}), \tag{2.5}$$

where η is the learning rate. Similarly to equation 17 in Olshausen and Field (1997) or to equation 2 in Smith and Lewicki (2006), the relation is a linear Hebbian rule (Hebb, 1949) since it enhances the weight of neurons proportionally to the correlation between pre- and postsynaptic neurons. Note that there is no learning for nonactivated coefficients. The novelty of this formulation compared to other linear Hebbian learning rule, such as Oja (1982) is to take advantage of the sparse representation—hence, the name *Sparse Hebbian Learning* (SHL).

SHL algorithms are unstable without homeostasis. In fact, starting with a random dictionary, the first filters to learn are more likely to correspond to salient features (Perrinet, Samuelides, & Thorpe, 2004) and are therefore more likely to be selected again in subsequent learning steps. In SparseNet, the homeostatic gain control is implemented by adaptively tuning the norm of the filters. This method equalizes the variance of coefficients across neurons using a geometric stochastic learning rule. The underlying heuristic is that this introduces a bias in the choice of the active coefficients. In fact, if a neuron is not selected often, the geometric homeostasis will decrease the norm of the corresponding filter, and therefore—from equation 2.1 and the conjugate gradient optimization—this will increase the value of the associated scalar. Finally, since the prior functions defined in equation 2.3 are identical for all neurons, this will increase the relative probability that the neuron is selected with a higher relative value. The parameters of this homeostatic rule have a great importance for the convergence of the global algorithm. We will now try to define a more general homeostasis mechanism derived from the optimization of representation efficiency.

2.3 Efficient Cooperative Homeostasis in SHL. The role of homeostasis during learning is to make sure that the distribution of neural activity is homogeneous. In fact, neurons belonging to a same neural ensemble (Hebb, 1949) form a competitive network and should a priori carry similar information. This optimizes the coding efficiency of neural activity in terms of compression (van Hateren, 1993) and thus minimizes intrinsic noise (Srinivasan, Laughlin, & Dubs, 1982). Such a strategy is similar to introducing an intrinsic adaptation rule such that the prior firing probability of all neurons has a similar Laplacian probability distribution (Weber & Triesch, 2008). Dually, since neural activity in the ensemble actually represents the sparse coefficients, we may understand the role of homeostasis as maximizing the average representation cost $\mathcal{C}(\mathbf{a} \mid \Phi)$ at the timescale of learning. This is equivalent to saying that homeostasis should act such that at any time, invariantly to the selectivity of features in the dictionary, the probability of selecting one feature is uniform across the dictionary.

This optimal uniformity may be achieved in all generality for any given dictionary by using point nonlinearities z_i applied to the sparse coefficients: In fact, a standard method to achieve uniformity is to use an equalization of the histogram (Atick, 1992). This method may be easily derived if we

know the probability distribution function dP_i of variable a_i by choosing the nonlinearity as the cumulative distribution function transforming any observed variable $\bar{a}_i$ into

$$z_i(\bar{a}_i) = P_i(a_i \leq \bar{a}_i) = \int_{-\infty}^{\bar{a}_i} dP_i(a_i). \tag{2.6}$$

This is equivalent to the change of variables that transforms the sparse vector $\mathbf{a}$ to a variable with uniform probability distribution function in $[0, 1]^M$. The transformed coefficients may thus be used as a normalized drive to the spiking mechanism of the individual neurons (see Figure 1, left). This equalization process has been observed in the neural activity of a variety of species and is, for instance, perfectly illustrated in the salamander's retina (Laughlin, 1981). It may evolve dynamically to slowly adapt to varying changes in luminance or contrast values, such as when the light diminishes at twilight (Hosoya, Baccus, & Meister, 2005).

This novel and simple nonparametric homeostatic method is applicable to SHL algorithms by using this transform on the sparse coefficients. Let us imagine, for instance, that one filter corresponds to a feature of low selectivity, while others correspond to similarly selective features. As a consequence, this filter will correspond on average to lower sparse coefficients (see Figure 1, right). However, the respective gain control function z_i will be such that all transformed coefficients have the same probability density function. Using the transformed coefficients to evaluate which neuron should be active, the homeostasis will therefore optimize the information in the representation cost defined in equation 2.4. We will now illustrate how it may be applied to Adaptive Matching Pursuit (Perrinet et al., 2003; Perrinet, 2004) and measure its role on the emergence of edge-like simple cell receptive fields.

3 Methods

3.1 Matching Pursuit and Adaptive Matching Pursuit. We first define Adaptive Matching Pursuit. We saw that optimizing the efficiency by minimizing the L_0 norm cost leads to a combinatorial search with regard to the dimension of the dictionary. In practice, it means that for a given dictionary, finding the best sparse vector according to minimizing $\mathcal{C}_0(\mathbf{a} \mid \mathbf{I}, \Phi)$ (see equation 2.4) is hard, and thus that learning an adapted dictionary is difficult. As Perrinet, Samuelides, and Thorpe (2002), proposed, we may solve this problem using a greedy approach. In general, a greedy approach is applied when finding the best combination of elements is difficult to solve globally. A simpler solution is to solve the problem progressively, one element at a time.

Applied to equation 2.4, it corresponds to first choosing the single element $a_i \Phi_i$ that best fits the image. From the definition of the LGM, we

know that for a given signal $\mathbf{I}$, the probability $P(\{a_i\} \mid \mathbf{I}, \Phi)$ corresponding to a single source $a_i \Phi_i$ for any i is maximal for the dictionary element i^* with maximal correlation coefficient:

$$i^* = \mathrm{ArgMax}_i(\rho_i), \quad \text{with} \quad \rho_i = \left\langle \frac{\mathbf{I}}{\|\mathbf{I}\|}, \frac{\Phi_i}{\|\Phi_i\|} \right\rangle. \tag{3.1}$$

This formulation is slightly different from equation 21 in Olshausen and Field (1997). It should be noted that ρ_i is the L-dimensional cosine (L is the dimension of the input space) and that its absolute value is therefore bounded by 1. The value of $\mathrm{ArcCos}(\rho_i)$ would therefore give the angle of $\mathbf{I}$ with the pattern Φ_i, and, in particular, the angle (modulo 2π) would be equal to zero if and only if $\rho_i = 1$ (full correlation), π if and only if $\rho_i = -1$ (full anticorrelation), and $\pm\pi/2$ if $\rho_i = 0$ (both vectors are orthogonal; there is no correlation). The associated coefficient is the scalar projection

$$a_{i^*} = \left\langle \mathbf{I}, \frac{\Phi_{i^*}}{\|\Phi_{i^*}\|^2} \right\rangle. \tag{3.2}$$

Second, knowing this choice, the image can be decomposed in

$$\mathbf{I} = a_{i^*} \Phi_{i^*} + \mathbf{R}, \tag{3.3}$$

where $\mathbf{R}$ is the residual image. We then repeat this two-step process on the residual (i.e., with $\mathbf{I} \leftarrow \mathbf{R}$) until some stopping criterion is met.

Hence, we have a sequential algorithm that permits reconstructing the signal using the list of choices; we call it sparse spike coding (Perrinet et al., 2002). The coding part of the algorithm produces a sparse representation vector $\mathbf{a}$ for any input image. Its L_0 norm is the number of active neurons. Note that the norm of the filters has no influence in this algorithm on the choice function or on the cost. For simplicity and without loss of generality, we will thereafter set the norm of the filters to 1: $\forall i, \|A_i\| = 1$. It is equivalent to the MP algorithm (Mallat & Zhang, 1993), and we have proven previously that this yields an efficient algorithm for representing natural images. Using MP in the SHL scheme defined in section 2.2 defines Adaptive Matching Pursuit (AMP) (Perrinet et al., 2003; Perrinet, 2004) and is similar to other strategies such as those of Smith and Lewicki (2006) and Rehn and Sommer (2007). This class of SHL algorithms offers a nonparametric solution to the emergence of simple cell receptive fields, but compared to SparseNet, the results often appear to be qualitatively nonhomogeneous. Moreover, the heuristic used in SparseNet for the homeostasis may not be used directly, since in MP, the choice is independent of the norm of the filter. The coding algorithm's efficiency may be improved using Optimized Orthogonal MP (Rebollo-Neira & Lowe, 2002) and be integrated in an SHL scheme (Rehn & Sommer, 2007). However, this

optimization is separate from the problem that we try to tackle here by optimizing the representation at the learning timescale. We will now study how we may use cooperative homeostasis in order to optimize the overall coding efficiency of the dictionary learned by AMP.

3.2 Competition-Optimized Matching Pursuit. In fact, we may now include cooperative homeostasis into AMP. At the coding level, it is important to note that if we simply equalize the sparse output of the MP algorithm, transformed coefficients will indeed be uniformly distributed, but the sequence of chosen filters will not be changed. However, the MP algorithm is nonlinear, and the choice of an element at one step may influence the rest of the choices. This sequence is therefore crucial for the representation efficiency. In order to optimize the competition of the choice step, we may instead choose at every matching step the item in the dictionary corresponding to the most significant value computed, thanks to the cooperative homeostasis (see Figure 1, right). In practice, it means that we select the best match in the vector corresponding to the transformed coefficients $\mathbf{z}$, that is, in the vector of the residual coefficients weighted by the nonlinearities defined by equation 2.6. This scheme thus extends the MP algorithm that we used previously by linking it to a statistical model that optimally tunes the ArgMax operator in the matching step. Over natural images, for any given dictionary—and thus independent of the selectivity of the different items from the dictionary—the choice of a neuron is statistically equally probable. Thanks to cooperative homeostasis, the efficiency of every match in MP is thus maximized—hence the name Competition-Optimized Matching Pursuit (COMP).

We now explicitly describe the COMP coding algorithm step by step. Initially, given the signal $\mathbf{I}$, we set up for all i an internal activity vector $\bar{\mathbf{a}}$ as the linear correlation using equation 3.2. The output sparse vector is set initially to a zero vector: $\mathbf{a} = \mathbf{0}$. Using the internal activity $\bar{\mathbf{a}}$, the neural population will evolve dynamically in an event-based manner by repeating the two following steps. First, the "matching" step is defined by choosing the address with the most significant activity:

$$i^* = \mathrm{ArgMax}_i[z_i(\bar{a}_i)]. \tag{3.4}$$

Then we set the winning sparse coefficient at address i^* with $a_{i^*} \leftarrow \bar{a}_{i^*}$. In the second "pursuit" step, as in MP, the information is fed back to correlated dictionary elements by

$$\bar{a}_i \leftarrow \bar{a}_i - a_{i^*}\langle \Phi_{i^*}, \Phi_i \rangle. \tag{3.5}$$

Note that after the update, the winning internal activity is zero—$\bar{a}_{i^*} = 0$—and that, as in MP, a neuron is selected at most once. Physiologically, as previously described, the pursuit step could be implemented by a lateral, correlation-based inhibition. The algorithm is iterated with equation 3.4

until some stopping criterion is reached, such as when the residual error energy is below the noise level σ_n^2. As in MP, since the residual is orthogonal to Φ_{i^*}, the residual error energy $E = \|\mathbf{I}\|^2$ may be easily updated at every step as

$$E \leftarrow E - a_{i^*}^2. \tag{3.6}$$

COMP transforms the image $\mathbf{I}$ into the sparse vector $\mathbf{a}$ at any precision $\sqrt{E}$. As in MP, the image may be reconstructed using: $\bar{\mathbf{I}} = \sum_i a_i \Phi_i$, which thus gives a solution for equation 2.1. COMP differs from MP only by the "matching" step and shares many properties with MP, such as the monotonous decrease of the error (see equation 3.6) or the exponential convergence of the coding. However, the decrease of E will always be faster in MP than in COMP from the constraint in the matching step.

Yet for a given dictionary, we do not know a priori the functions z_i since they depend on the computation of the sparse coefficients. In practice, the z_i functions are initialized for all neurons to similar arbitrary cumulative distribution functions (COMP is then equivalent to the MP algorithm since choices are not affected). Since we have at most one sparse value a_i per neuron, the cumulative histogram function for each neuron for one coding sweep is $P(a_i \leq \bar{a}_i) = \delta(a_i \leq \bar{a}_i)$, where variable $\bar{a}_i$ is the observed coefficient to be transformed and δ is the Dirac measure: $\delta(B) = 1$ if the Boolean variable B is true and 0 otherwise. We evaluate equation 2.6 after the end of every coding using an online stochastic algorithm, $\forall i, \forall \bar{a}_i$:

$$z_i(\bar{a}_i) \leftarrow (1 - \eta_h) z_i(\bar{a}_i) + \eta_h \delta(a_i \leq \bar{a}_i), \tag{3.7}$$

where η_h is the homeostatic learning rate. Note that this corresponds to the empirical estimation and assumes that coefficients are stationary on a timescale of $\frac{1}{\eta_h}$ learning steps. The timescale of homeostasis should therefore in general be less than the timescale of learning. Moreover, due to the exponential convergence of MP, for any set of components, the z_i functions converge to the correct nonlinear functions as defined by equation 2.6.

3.3 Adaptive Sparse Spike Coding. We may finally apply COMP to SHL (see section 2.2). Since the efficiency is inspired by the spiking nature of neural representations, we call this algorithm adaptive Sparse Spike Coding (aSSC). From the definition of COMP, we know that whatever the dictionary, the competition between filters will be fair because of the cooperative homeostasis. We add no other homeostatic regulation. We normalize the energy of the filters since it is a free parameter in equation 3.1.

In summary, the learning algorithm is given by the following nested loops in pseudo-code:

1. Initialize the point nonlinear gain functions z_i to similar cumulative distribution functions and the components Φ_i to random points on the unit L-dimensional sphere,
2. Repeat until learning converged:
   (a) draw a signal $\mathbf{I}$ from the database, its energy is $E = \|\mathbf{I}\|^2$,
   (b) set sparse vector $\mathbf{a}$ to zero, initialize $\bar{a}_i = \langle \mathbf{I}, \Phi_i \rangle$ for all i,
   (c) while the residual energy E is above a given threshold do:
      i. select the best match: $i^* = \text{ArgMax}_i[z_i(\bar{a}_i)]$,
      ii. set the sparse coefficient: $a_{i^*} = \bar{a}_{i^*}$,
      iii. update residual coefficients: $\forall i, \bar{a}_i \leftarrow \bar{a}_i - a_{i^*}\langle \Phi_{i^*}, \Phi_i \rangle$,
      iv. update energy: $E \leftarrow E - a_{i^*}^2$.
   (d) when we have the sparse representation vector $\mathbf{a}$, apply $\forall i$:
      i. modify dictionary: $\Phi_i \leftarrow \Phi_i + \eta a_i(\mathbf{I} - \Phi\mathbf{a})$,
      ii. normalize dictionary: $\Phi_i \leftarrow \Phi_i / \|\Phi_i\|$,
      iii. update homeostasis functions: $z_i(\cdot) \leftarrow (1 - \eta_h)z_i(\cdot) + \eta_h \delta(a_i \leq \cdot)$.

4 Results on Natural Images

The aSSC algorithm differs from the SparseNet algorithm by the MP sparse coding algorithm and by the cooperative homeostasis. Using natural images, we evaluate the relative contribution of these different mechanisms to the representation efficiency.

4.1 Receptive Field Formation. We first compare the dictionaries of filters obtained by both methods. We use a similar context and architecture as the experiments described in Olshausen and Field (1997) and specifically the same database of image patches as the SparseNet algorithm. These images are static, gray scale, and whitened according to the same parameters to allow a one-to-one comparison of both algorithms. Here we show the results for 16×16 image patches (so that $L = 256$) and the learning of $M = 324$ filters, which are replicated as ON and OFF filters. Assuming this symmetry in the aSSC algorithm, we use the absolute value of the coefficient in equations 3.4 and 3.7, the rest of the algorithm being identical.[1] Results replicate the original results of Olshausen and Field (1997) and are comparable for both methods: dictionaries consist of edge-like filters similarly to the receptive fields of simple cells in the primary visual cortex (see Figure 2). Study of the evolution of receptive fields during learning shows that they first represent any salient feature (such as sharp corners or edges) because these correspond to larger Lipschitz coefficients (Perrinet et al., 2004). If a receptive field contains multiple singularities, only the most salient remains later during learning: due to the competition between filters, the algorithm

[1] That is, following section 3.3, step 2-c-i becomes $i^* = \text{ArgMax}_i[z_i(|\bar{a}_i|)]$, and step 2-d-iii is changed to $z_i(\cdot) \leftarrow (1 - \eta_h)z_i(\cdot) + \eta_h \delta(|a_i| \leq \cdot)$.

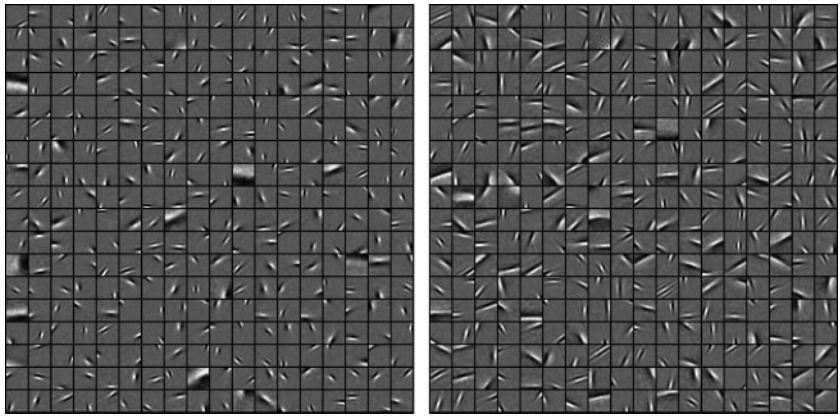

Figure 2: Comparison of the dictionaries obtained with SparseNet and aSSC. We show the results of SHL using two different sparse coding algorithms at convergence (20,000 learning steps). (Left) Conjugate gradient function (CGF) method as used in SparseNet (Olshausen & Field, 1997) with (right) COMP as used in aSSC. Filters of the same size as the image patches are presented in a matrix (separated by a black border). Note that their position in the matrix is arbitrary, as in ICA.

eliminates features that are duplicated in the dictionary. Filters that already converged to independent components will be selected sparsely and with high associated coefficients, but inducing slower learning since the corresponding error is small (see equation 2.5). We observe for both algorithms that when considering very long learning times, the solution is not fixed, and edges may slowly drift from one orientation to another while global efficiency remains stable. This is due to the fact that there are many solutions to the same problem (note, for instance, that solutions are invariant up to a permutation of neurons' addresses). It is possible to decrease these degrees of freedom by including, for instance, topological links between filters (Bednar, Kelkar, & Miikkulainen, 2004). Qualitatively, the main difference between both results is that filters produced by aSSC look more diverse and broad (so that they often overlap), while the filters produced by SparseNet are more localized and thin.

We also perform robustness experiments to determine the range of learning parameters for which these algorithms converged. One advantage of aSSC is that it is based on a nonparametric sparse coding and a nonparametric homeostasis rule and is entirely described by two structural parameters (L and M) and two learning parameters (η and η_h), while parameterization of the prior and of the homeostasis for SparseNet requires five more parameters to adjust (three for the prior, two for the homeostasis). By observing at convergence the probability distribution function of selected

filters, homeostasis in aSSC converges for a wide range of η_h values (see equation 3.7). Furthermore, we observe that at convergence, the z_i functions become very similar (see the dotted lines in Figure 1, right) and that homeostasis does not favor the selection of any particular neuron as strongly as at the beginning of the learning. Therefore, thanks to the homeostasis, equilibrium is reached when the dictionary homogeneously represents different features in natural images, that is, when filters have similar selectivities. Finally, we observe the counterintuitive result that nonlinearities implementing cooperative homeostasis are important for the coding only during the learning period but that it may be ignored for the coding after convergence since at this point, nonlinearities are the same for all neurons.

Both dictionaries appear to be qualitatively different and, for instance, parameters of the emerging edges (frequency, length, width) are distributed differently. In fact, it seems that rather than the shape of each dictionary element taken individually, it is their distribution in image space that yields different efficiencies. Such an analysis of the filters' shape distribution was performed quantitatively for SparseNet in Lewicki and Sejnowski (2000). The filters were fitted by Gabor functions (Jones & Palmer, 1987). A recent study compares the distribution of fitted Gabor functions' parameters between the model and receptive fields obtained from neurophysiological experiments conducted in primary visual cortex of macaques (Rehn & Sommer, 2007). It has shown that their SHL model based on Optimized Orthogonal MP better matches to physiological observations than SparseNet does. However, there is no theoretical basis for the fact that receptive fields' shape should be well fitted by Gabor functions (Saito, 2001), and the variety of shapes observed in biological systems may, for instance, reflect adaptive regulation mechanisms when reaching different optimal sparseness levels (Assisi, Stopfer, Laurent, & Bazhenov, 2007). Moreover, although this type of quantitative method is certainly necessary, it is not sufficient to understand the role of each individual mechanism in the emergence of edge-like receptive fields. To assess the relative role of coding and homeostasis in SHL, we compare these different dictionaries quantitatively in terms of representation efficiency.

4.2 Coding Efficiency in SHL. To address this issue, we first compare the quality of both dictionaries (from SparseNet and aSSC) by computing the mean efficiency of their respective coding algorithms (respectively, CGF and COMP). Using 10^5 image patches drawn from the natural image database, we perform the progressive coding of each image using both sparse coding methods. When the probability distribution function of the sparse coefficients is plotted, one observes that distributions fit well the bivariate model introduced in Olshausen and Millman (2000), where a subset of the coefficients is null (see Figure 3, left). Log-probability distributions of nonzero coefficients are quadratic with the initial random dictionaries. At convergence, nonzero coefficients fit well to a Laplacian probability

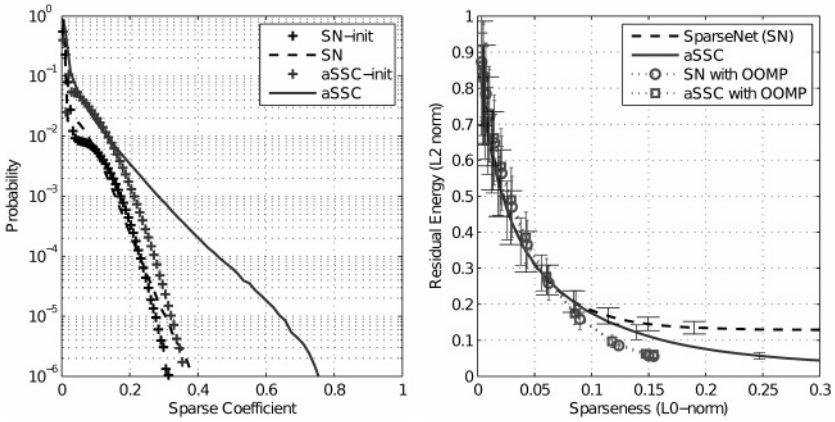

Figure 3: Coding efficiency of SparseNet versus aSSC. We evaluate the quality of both learning schemes by comparing the coding efficiency of their respective coding algorithms, that is, CGF and COMP, with the respective dictionary that was learned (see Figure 2). (Left) We show the probability distribution function of sparse coefficients obtained by both methods with random dictionaries (respectively, SN-init and aSSC-init) and with the dictionaries obtained after convergence of respective learning schemes (respectively, SN and aSSC). At convergence, sparse coefficients are more sparsely distributed than initially, with more kurtotic probability distribution functions for aSSC in both cases. (Right) We plot the average residual error (L_2 norm) as a function of the relative number of active (nonzero) coefficients. This provides a measure of the coding efficiency for each dictionary over the set of image patches (error bars are scaled to one standard deviation). The L_0 norm is equal to the coding step in COMP. Best results are those providing a lower error for a given sparsity (better compression) or a lower sparseness for the same error (Occam's razor). We observe similar coding results in aSSC despite its nonparametric definition. This result is also true when using the two different dictionaries with the same OOMP sparse coding algorithm: the dictionaries still have similar coding efficiencies.

distribution function. Measuring mean kurtosis of resulting sparse vectors proves to be very sensitive and a poor indicator of global efficiency, in particular at the beginning of the coding, when many coefficients are still strictly zero. In general, COMP provides a sparser final distribution. Dually, plotting the decrease of the sorted coefficients as a function of their rank shows that coefficients for COMP are first higher and then decrease more quickly due to the link between the z_i functions and the function of sorted coefficients (see equation 2.6). As a consequence, a Laplacian bivariate model for the distribution of sparse coefficient emerges from the statistics of natural images. The advantage of aSSC is that this emergence is not dependent on a parametric model of the prior.

In a second analysis, we compare the efficiency of both methods while varying the number of active coefficients (the L_0 norm). We perform this in COMP by simply measuring the residual error (L_2 norm) with respect to the coding step. To compare this method with the conjugate gradient method, we use a two-pass sparse coding: a first pass identifies best neurons for a fixed number of active coefficients, while a second pass optimizes the coefficients for this set of active vectors. This method was also used in Rehn and Sommer (2007) and proved to be fair when comparing both algorithms. We observe in a robust manner that the greedy solution to the hard problem (i.e., COMP) is as efficient as conjugate gradient as used in SparseNet (see Figure 3, right). We also observe that aSSC is also slightly more efficient for the cost defined in equation 2.3, a result that may reflect the fact that the L_0 norm defines a stronger sparseness constraint than the convex cost. Moreover, we compare the coding efficiency of both dictionaries using OOMP. Results show that OOMP provides a slight coding improvement but also confirms that both dictionaries are of similar coding efficiency, independent of their respective coding algorithm.

These results prove that without the need of a parameterization of the prior, coding in aSSC is as efficient as SparseNet. In addition, a number of other advantages stern from this approach. First, COMP simply uses a feedforward pass with lateral interactions, while conjugate gradient is implemented as the fixed point of a recurrent network (see Figure 13.2 from Olshausen, 2002). Moreover, we have already seen that aSSC is a nonparametric method controlled by fewer parameters. Therefore, applying a "higher-level" Occam razor confirms that for a similar overall coding efficiency, aSSC is better since it is of lower structural complexity.[2] Finally, in SparseNet and in algorithms defined in Lewicki and Sejnowski (2000), Smith and Lewicki (2006), and Rehn and Sommer (2007), representation is analog without explicitly defining a quantization. This is not the case in the aSSC algorithm, where cooperative homeostasis introduces a regularity in the distribution of sparse coefficients.

4.3 Role of Homeostasis in Representation Efficiency. In the context of an information channel such as implemented by a neural ensemble, one should rather use the coefficients that could be decoded from the neural signal in order to define the reconstruction cost (see Figure 1, left). As was described in section 2.1, knowing a dictionary Φ, it is indeed better to consider the overall average coding and decoding cost over image patches

[2] A quantitative measure of the structural complexity for the different methods is given by the minimal length of a code that would implement them, this length being defined as the number of characters of the code implementing the algorithm. It would therefore depend on the machine on which it is implemented, and there is, of course, a clear advantage of aSSC on parallel architectures.

$\mathcal{C}(\hat{\mathbf{a}} \mid \mathbf{I}, \Phi)$ (see equation 2.2), where $\hat{\mathbf{a}}$ corresponds to the analog vector of coefficients inferred from the neural representation. The overall transmission error may be described as the sum of the reconstruction and the quantization error. This last error will increase with both intertrial variability but also with the nonhomogeneity of the represented features. It is, however, difficult to evaluate a decoding scheme in most sparse coding algorithms since this problem is generally not addressed. Our objective when defining $\mathcal{C}_0(\hat{\mathbf{a}} \mid \mathbf{I}, \Phi)$ (see equation 2.4) was to define sparseness as it may be represented by spiking neural representations. Using a decoding algorithm on such a representation will help us to quantify overall coding efficiency.

An effective decoding algorithm is to estimate the analog values of the sparse vector (and thus reconstruct the signal) from the order of neurons' activation in the sparse vector (Perrinet, 2007). In fact, knowing the address of the fiber i^0 corresponding to the maximal value, we may infer that it has been produced by an analog value on the emitter side in the highest quantile of the probability distribution function of a_{i^0}. We may therefore decode the corresponding value with the best estimate, which is given as the average maximum sparse coefficient for this neuron by inverting z_{i^0} (see equation 2.6): $a_{i^0} = z_{i^0}^{-1}(1)$.[3] This is also true for the following coefficients. We write as $\frac{r}{M}$ the relative rank of the rth and o the order function that gives the address of the winning neuron at rank r. Since $z_{o(r)} = 1 - \frac{r}{M} = z_{o(r)}(a_{o(r)})$, we can reconstruct the corresponding value as

$$\hat{a}_{o(r)} = z_{o(r)}^{-1}\left(1 - \frac{r}{M}\right). \tag{4.1}$$

Physiologically, equation 4.1 could be implemented using interneurons, which would "count" the number of received spikes, and by modulating efficiency of synaptic events on receiver efferent neurons—for instance, with shunting inhibition (Delorme & Thorpe, 2003). Recent findings show that this type of code may be used in cortical in vitro recurrent networks (Shahaf et al., 2008). This corresponds to a generalized rank coding scheme. However this quantization does not require that neural information explicitly carries rank information. In fact, this scheme is rather general and is analogous to scalar quantization using the modulation function z_i^{-1} as a look-up table. It is very likely that fine temporal information such as interspike intervals also plays a role in neural information transmission. As in other decoding schemes, the quantization error directly depends on the variability of the modulation functions across trials (Perrinet et al., 2004). This scheme thus shows a representative behavior for the retrieval of information from spiking neural activity.

[3] Mathematically, the z_i are not always *strictly* increasing, and we state here that $z_i^{-1}(z)$ is defined in a unique way as the average value of the coefficients a_i such that $z_i(a_i) = z$.

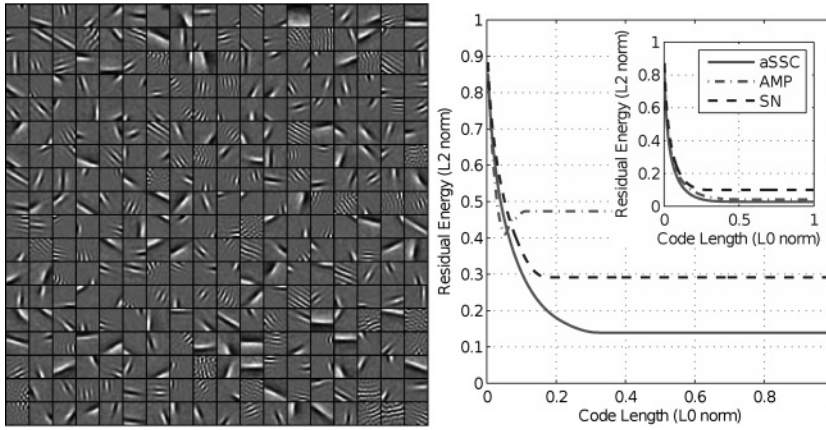

Figure 4: Cooperative homeostasis implements efficient quantization. (Left) When switching off the cooperative homeostasis during learning, the corresponding SHL algorithm, AMP, converges to a set of filters that contain some less localized filters and some high-frequency Gabor functions that correspond to more textural features (Perrinet et al., 2003). One may wonder if these filters are inefficient and capturing noise or if they correspond to independent features of natural images in the LGM model. (Right, inset) In fact, when residual energy is being plotted as a function of L_0 norm sparseness with the MP algorithm (as plotted in Figure 3, right), the AMP dictionary gives a slightly worse result than aSSC. (Right) Moreover, one should consider representation efficiency as the overall coding and decoding algorithm. We compare the efficiency for these dictionaries thanks to same coding method (SSC) and the same decoding method (using rank-quantized coefficients). Representation length for this decoding method is proportional to the L_0 norm, with $\lambda = \frac{\log(M)}{L} \approx 0.032$ bits per coefficient and per pixel, as defined in equation 2.4. We observe that the dictionary obtained by aSSC is more efficient than the one obtained by AMP, while the dictionary obtained with SparseNet (SN) gives an intermediate result thanks to the geometric homeostasis. Introducing cooperative homeostasis globally improves neural representation.

To evaluate the specific role of cooperative homeostasis, we compare previous dictionaries (see Figure 2) with the one obtained by AMP. In fact, SparseNet and aSSC differ at the level of the homeostasis but also for the sparse coding. The only difference between aSSC and AMP is the introduction of cooperative homeostasis. To obtain the solution to AMP, we use the same sparse coding algorithm but switch off the cooperative homeostasis during learning ($\eta_h = 0$ in equation 3.7). We observe at convergence that the dictionary corresponds qualitatively to features that are different from aSSC and SparseNet (see Figure 4, left). In particular, we observe

the emergence of Gabor functions with broader width, which better match textures. These filters correspond to lower Lipschitz coefficients (Perrinet et al., 2004), and because of their lower saliency, these textural filters are more likely to be selected with lower correlation coefficients. They fit more to the Fourier filters that are obtained using principal component analysis (Fyfe & Baddeley, 1995) and are still optimal to code arbitrary image patches such as noise (Zhaoping, 2006). When we plot the L_2 norm with respect to the L_0 norm for the different dictionaries with the same MP coding algorithm averaged over a set of 10^5 image patches from natural scenes (see Figure 4, right inset), the resulting dictionary from AMP is less efficient than those obtained with aSSC and SparseNet. This is not an expected behavior since COMP is more constrained than MP (MP is the "greediest" solution), and using both methods with a similar dictionary would necessarily give an advantage to MP: the AMP thus reached a local minima of the coding cost. To understand why, recall that in the aSSC algorithm, the cooperative homeostasis constraint, by its definition in equation 2.6, plays the role of a gain control and that the point nonlinearity from equation 3.4 ensures that all filters are selected equally. Compared to AMP, textured elements are boosted during learning relative to a more generic salient edge component and are thus more likely to evolve (see Figure 1, right). This explains why they would end up being less probable and that at convergence, there are no textured filters in the dictionary obtained with aSSC.

Finally, we test quantitatively the representation efficiency of these different dictionaries with the same quantization scheme. At the decoding level, we compute in all cases the modulation functions as defined in equation 4.1 on a set of 10^5 image patches from natural scenes. Since addresses' choices may be generated by any of the M neurons, the representation cost is defined as $\lambda = \log(M)$ bits per chosen address (see equation 2.4). Then, when the quantization is used (see equation 4.1), the AMP approach displays a larger variability, reflecting the lack of homogeneity of the features represented by the dictionary. There is a much larger reconstruction error and a slower decrease of error's energy (see Figure 4, right). The aSSC, on the contrary, is adapted to quantization thanks to cooperative homeostasis, and consequently it yields a more regular decrease of coefficients as a function of rank, that is, a lower quantization error. The dictionary obtained with the SparseNet algorithm yields an intermediate result. This shows that the heuristic implementing the homeostasis in this algorithm regulates relatively well the choices of the elements during the learning. It also explains why the three parameters of the homeostasis algorithm had to be properly tuned to fit the dynamics of the heuristics. Results therefore show that homeostasis optimizes the efficiency of the neural representation during learning and that the cooperative homeostasis provides a simple and effective optimization scheme.

5 Discussion

We have shown in this letter that homeostasis plays an essential role in sparse Hebbian learning (SHL) schemes and thus on our understanding of the emergence of simple cell receptive fields. First, using statistical inference and information theory, we have proposed a quantitative cost for the coding efficiency based on a nonparametric model using the number of active neurons, that is, the L_0 norm of the representation vector. This allowed the design of a cooperative homeostasis rule based on neurophysiological observations (Laughlin, 1981). This rule optimizes the competition between neurons by simply constraining the choice of every selection of an active neuron to be equiprobable. This homeostasis defined a new sparse coding algorithm, COMP, and a new SHL scheme, aSSC. Then we confirmed that the aSSC scheme provides an efficient model for the formation of simple cell receptive fields, similar to other approaches. The sparse coding algorithms in these schemes are variants of conjugate gradient or of Matching Pursuit (MP). They are based on correlation-based inhibition since this is necessary to remove redundancies from the linear representation. This is consistent with the observation that lateral interactions are necessary for the formation of elongated receptive fields (Bolz & Gilbert, 1989). With a correct tuning of parameters, all schemes show the emergence of edge-like filters. The specific coding algorithm used to obtain this sparseness appears to be of secondary importance as long as it is adapted to the data and yields sufficiently efficient sparse representation vectors. However, the resulting dictionaries vary qualitatively among these schemes, and it was unclear which algorithm is the most efficient and what was the individual role of the different mechanisms that constitute SHL schemes. At the learning level, we have shown that the homeostasis mechanism had a great influence on the qualitative distribution of learned filters. In particular, using the comparison of the coding and decoding efficiency of aSSC with and without this specific homeostasis, we have proven that cooperative homeostasis optimized overall representation efficiency. This efficiency is comparable to that of SparseNet, but with the advantage that our unsupervised learning model is nonparametric and does not need to be properly tuned.

This work might be advantageously applied to signal processing problems. First, we saw that optimizing the representation cost maximizes the independence between features and is related to the goal of ICA. Since we have built a solution to the LGM inverse problem that is more efficient than standard methods such as the SparseNet algorithm, it is thus a good candidate solution to blind source separation problems. Second, at the coding level, we optimized in the COMP algorithm the efficiency of MP by including an adaptive cooperative homeostasis mechanism. We proved that for a given compression level, image patches are more efficiently coded than in the MP algorithm. Since we have shown previously that MP compares

favorably with compression methods such as JPEG with a fixed log-Gabor filter dictionary (Fischer, Redondo, Perrinet, & Cristóbal, 2007), we can predict that COMP should provide promising results for image representation. An advantage over other sparse coding schemes is that it provides a progressive dynamical result, while the conjugate gradient method has to be recomputed for every different number of coefficients. The most relevant information is propagated first, and progressive reconstruction may be interrupted at any time. Finally, a main advantage of this type of neuromorphic algorithm is that it uses a simple set of operations: computing the correlation, applying the point nonlinearity from a look-up table, choosing the ArgMax, doing a subtraction, and retrieving a value from a look-up table. In particular, the complexity of these operations, such as the ArgMax operator, in theory would not depend on the dimension of the system in parallel machines and the transfer of this technology to neuromorphic hardware such as aVLSIs (Schemmel, Gruebl, Meier, & Mueller, 2006; Brüderle et al., 2009) will provide a supralinear gain of performance.

In this letter, we focused on transient input signals and of relatively abstract neurons. This choice was made to highlight the powerful function of the parallel and temporal competition between neurons in contrast to traditional analog and sequential strategies using analog spike frequency representations. This strategy allowed a comparison of the proposed learning scheme with state-of-the-art algorithms. One obvious extension to the algorithm is to implement learning with more realistic inputs. In fact, sparseness in image patches is only local while it is also spatial and temporal in whole-field natural scenes. For instance, it is highly probable in whole natural images that large parts of the space, such as the sky, are flat and contain no information. Our results should thus be taken as a lower bound for the efficiency of aSSC in natural scenes. This also suggests the extension to representations with some built-in invariances, such as translation and scaling. A gaussian pyramid, for instance, provides a multiscale representation where the set of learned filters would become a dictionary of mother wavelets (Perrinet, 2007). Such an extension leads to a fundamental question: How does representation efficiency evolve with the number M of elements in the dictionary, that is, with the complexity of the representation? In fact, when increasing the overcompleteness in aSSC, one observes the emergence of different classes of edge filters: at first different positions, then different orientations of edges, followed by different frequencies, and so on. This specific order indicates the existence of an underlying hierarchy for the synthesis of natural scenes. This hierarchy seems to correspond to the level of importance of the different transformations that are learned by the system—respectively, translation, rotation, and scaling. Exploring the efficiency results for different dimensions of the dictionary in aSSC will thus give a quantitative evaluation of the optimal complexity of the model needed to describe images in terms of a trade-off between accuracy and generality. But it may also provide a model for the clustering of

the low-level visual system into different areas, such as the emergence of position-independent representations in the ventral visual pathway versus motion-selective neurons in the dorsal visual pathway.

Acknowledgments

This work was supported by a grant from the French Research Council (ANR NatStats) and by EC IP project FP6-015879, FACETS. I thank the team at the Redwood Neuroscience Institute for stimulating discussions, particularly Jeff Hawkins, Bruno Olshausen, Fritz Sommer, Tony Bell, Dileep George, Kilian Koepsell, and Matthias Bethge. Special thanks to Jo Hausmann, Guillaume Masson, Nicole Voges, Willie Smit, and Artemis Kosta for essential comments on this work. I thank the anonymous referees for their helpful comments on the manuscript. Special thanks to Bruno Olshausen, Laura Rebollo-Neira, Gabriel Peyré, Martin Rehn, and Fritz Sommer for providing the source code for their experiments.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716–723.

Assisi, C., Stopfer, M., Laurent, G., & Bazhenov, M. (2007). Adaptive regulation of sparseness by feedforward inhibition. *Nature Neuroscience*, *10*(9), 1176–1184.

Atick, J. J. (1992). Could information theory provide an ecological theory of sensory processing? *Network: Computation in Neural Systems*, *3*(2), 213–252.

Barlow, H. B. (2001). Redundancy reduction revisited. *Network: Computation in Neural Systems*, *12*, 241–245.

Baudot, P., Levy, M., Monier, C., Chavane, F., René, A., Huguet, N., et al. (2004). Time-coding, low noise Vm attractors, and trial-by-trial spiking reproducibility during natural scene viewing in V1 cortex. In *Society for Neuroscience Abstracts: 34th Annual Meeting of the Society for Neuroscience, San Diego, USA* (pp. 948–512). Washington, DC: Society for Neuroscience.

Bednar, J. A., Kelkar, A., & Miikkulainen, R. (2004). Scaling self-organizing maps to model large cortical networks. *Neuroinformatics*, *2*(3), 275–302.

Bethge, M., Rotermund, D., & Pawelzik, K. (2003). Second order phase transition in neural rate coding: Binary encoding is optimal for rapid signal transmission. *Physical Review Letters*, *90*(8), 088104.

Bolz, J., & Gilbert, C. D. (1989). The role of horizontal connections in generating long receptive fields in the cat visual cortex. *European Journal of Neuroscience*, *1*(3), 263–268.

Brüderle, D., Müller, E., Davison, A., Muller, E., Schemmel, J., & Meier, K. (2009). Establishing a novel modeling tool: A python-based interface for a neuromorphic hardware system. *Frontiers in Neuroinformatics*, *3*, 17.

Chapman, B., & Stryker, M. P. (1992). Origin of orientation tuning in the visual cortex. *Current Opinion in Neurobiology*, *2*(4), 498–501.

Delorme, A., & Thorpe, S. J. (2003). Early cortical orientation selectivity: How fast shunting inhibition decodes the order of spike latencies. *Journal of Computational Neuroscience*, *15*, 357–365.

DeWeese, M. R., Wehr, M., & Zador, A. M. (2003). Binary coding in auditory cortex. *Journal of Neuroscience*, *23*(21), 7940–7949.

Doi, E., Balcan, D. C., & Lewicki, M. S. (2007). Robust coding over noisy overcomplete channels. *IEEE Transactions in Image Processing*, *16*(2), 442–452.

Field, D. J. (1994). What is the goal of sensory coding? *Neural Computation*, *6*(4), 559–601.

Fischer, S., Redondo, R., Perrinet, L., & Cristóbal, G. (2007). Sparse approximation of images inspired from the functional architecture of the primary visual areas. *EURASIP Journal on Advances in Signal Processing*, *vol. 2007(1)*:122.

Fyfe, C., & Baddeley, R. J. (1995). Finding compact and sparse-distributed representations of visual images. *Network: Computation in Neural Systems*, *6*, 333–344.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. New York: Wiley.

Hosoya, T., Baccus, S. A., & Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, *436*(7047), 71–77.

Jones, J. P., & Palmer, L. A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, *58*(6), 1233–1258.

Laughlin, S. B. (1981). A simple coding procedure enhances a neuron's information capacity. *Zeitung für Naturforschung*, *9–10*(36), 910–912.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by nonnegative matrix factorization. *Nature*, *401*, 788–791.

Lee, H., Battle, A., Raina, R., & Ng, A. (2007). Efficient sparse coding algorithms. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems, 19* (pp. 801–808). Cambridge, MA: MIT Press.

Lewicki, M. S., & Sejnowski, T. J. (2000). Learning overcomplete representations. *Neural Computation*, *12*(2), 337–365.

Mallat, S. (1998). *A wavelet tour of signal processing* (2nd ed.). Orlando, FL: Academic Press.

Mallat, S., & Zhang, Z. (1993). Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, *41*(12), 3397–3414.

Nikitin, A. P., Stocks, N. G., Morse, R. P., & McDonnell, M. D. (2009). Neural population coding is optimized by discrete tuning curves. *Physical Review Letters*, *103*(13), 138101.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*, 267–273.

Olshausen, B. A. (2002). Sparse codes and spikes. In R. P. N. Rao, B. A. Olshausen, & M. S. Lewicki (Eds.), *Probabilistic models of the brain: perception and neural Function* (pp. 257–272). Cambridge, MA: MIT Press.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, *37*, 3311–3325.

Olshausen, B. A., & Millman, K. J. (2000). Learning sparse codes with a mixture-of-gaussians prior. In M. I. Jordan, M. J. Kearns, & S. A. Solla (Eds.), *Advances in neural information processing systems*, *12* (pp. 887–893). Cambridge, MA: MIT Press.

Pece, A. E. C. (2002). The problem of sparse image coding. *Journal of Mathematical Imaging and Vision*, *17*, 89–108.

Perrinet, L. (2004). Finding independent components using spikes: A natural result of Hebbian learning in a sparse spike coding scheme. *Natural Computing*, *3*(2), 159–175.

Perrinet, L. (2007). Dynamical neural networks: Modeling low-level vision at short latencies. *European Physical Journal*, *142*, 163–225.

Perrinet, L., Samuelides, M., & Thorpe, S. J. (2002). Sparse spike coding in an asynchronous feed-forward multi-layer neural network using Matching Pursuit. *Neurocomputing*, *57C*, 125–134.

Perrinet, L., Samuelides, M., & Thorpe, S. J. (2003). Emergence of filters from natural scenes in a sparse spike coding scheme. *Neurocomputing*, *58–60*(C), 821–826.

Perrinet, L., Samuelides, M., & Thorpe, S. J. (2004). Coding static natural images using spiking event times: Do neurons cooperate? *IEEE Transactions on Neural Networks*, *15*(5), 1164–1175.

Ranzato, M. A., Poultney, C. S., Chopra, S., & LeCun, Y. (2007). Efficient learning of sparse overcomplete representations with an energy-based model. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems*, *19* (pp. 1137–1144). Cambridge, MA: MIT Press.

Rebollo-Neira, L., & Lowe, D. (2002). Optimized orthogonal Matching Pursuit approach. *IEEE Signal Processing Letters*, *9*(4), 137–140.

Rehn, M., & Sommer, F. T. (2007). A model that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of Computational Neuroscience*, *22*(2), 135–146.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Saito, N. (2001). The generalized spike process, sparsity, and statistical independence. In D. N. Rockmore & D. M. Healy (Eds.), *Modern signal processing* (pp. 317–340). Cambridge: Cambridge University Press.

Schemmel, J., Gruebl, A., Meier, K., & Mueller, E. (2006). Implementing synaptic plasticity in a VLSI spiking neural network model. In *Proceedings of the 2006 International Joint Conference on Neural Networks*. Piscataway, NJ: IEEE Press.

Shahaf, G., Eytan, D., Gal, A., Kermany, E., Lyakhov, V., Zrenner, C., et al. (2008). Order-based representation in random networks of cortical neurons. *PLoS Computational Biology*, *4*(11), e1000228+.

Smith, E. C., & Lewicki, M. S. (2006). Efficient auditory coding. *Nature*, *439*(7079), 978–982.

Srinivasan, M. V., Laughlin, S. B., & Dubs, A. (1982). Predictive coding: A fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, *216*(1205), 427–459.

van Hateren, J. H. (1993). Spatiotemporal contrast sensitivity of early vision. *Vision Research*, *33*, 257–267.

Vinje, W. E., & Gallant, J. L. (2000). Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, *287*, 1273–1276.

Weber, C., & Triesch, J. (2008). A sparse generative model of V1 simple cells with intrinsic plasticity. *Neural Computation*, *20*(5), 1261–1284.

Zhaoping, L. (2006). Theoretical understanding of the early visual processes by data compression and data selection. *Network: Computation in Neural Systems*, *17*(4), 301–334.

Zibulevsky, M., & Pearlmutter, B. A. (2001). Blind source separation by sparse decomposition. *Neural Computation*, *13*(4), 863–882.

Received May 28, 2008; accepted November 25, 2009.