Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Physiology - Paris 98 (2004) 407–416
Journal of Physiology Paris
www.elsevier.com/locate/jphysparis

Competitive spiking and indirect entropy minimization of rate code: Efficient search for hidden components

Botond Szatmáry, Barnabás Póczos, András Lőrincz *

Department of Information Systems, Faculty of Informatics, Eötvös Loránd University, Pázmány Péter sétány 1/C., Budapest H-1117, Hungary

Abstract

Our motivation, which originates from the psychological and physiological evidences of component-based representations in the brain, is to find neural methods that can efficiently search for structures. Here, an architecture made of coupled parallel working reconstruction subnetworks is presented. Each subnetwork utilizes non-negativity constraint on the generative weights and on the internal representation. 'Spikes' are generated within subnetworks via winner take all mechanism. Memory components are modified in order to directly minimize the reconstruction error and to indirectly minimize the entropy of the spike rate distribution, via a combination of a stochastic gradient search and a novel tuning method. This tuning dynamically changes the learning rate: the higher the entropy of the spike rate, the higher the learning rate of the gradient search in the subnetworks. This method effectively reduces the search space and increases the escape probability from high entropy local minima. We demonstrate that one subnetwork can develop localized and oriented components. Coupled networks can discover and sort components into the subnetworks; a problem subject to combinatorial explosion. Synergy between spike code and rate code is discussed.

Keywords: Grouping components; Reconstructive network; Spike code; Rate code; Indirect entropy minimization; Dynamic learning rate

1. Introduction

There is a serious gap in our understanding of how information in the brain is encoded. Among others, it has been proposed that information could be carried by (i) temporally distributed 'spike codes', (ii) 'rate codes' with noise reduction capabilities or (iii) 'population codes' [1] that encode features in a spatially distributed manner. From the point of view of computational potential, spike code [2–5] has become one of the favorites in modelling. Spike code allows for fast and efficient information transmission [6], but it is sensitive to temporal perturbations: timing and the number of spikes produced by a single cortical neuron show substantial variability [7,8]. The impact of this variability on information transmission can be decreased by rate coding, a coding strategy that is supported both theoretically [9–12] and experimentally [13–15,8].

In this paper a reconstruction neural network model is introduced, which demonstrates the appealing joined features of spike coding and rate coding. The subject of our work is to find efficient and biologically plausible mechanisms that search for structure and the components of the structure in the input data and are able to develop efficient sparse representation. Our motivation comes from recent theoretical models on neocortical processing (see e.g., [16,17]) which claim that sensory processing may be based on reconstruction networks in which temporal integration takes place at the internal (hidden) representation. Thus, it follows that the internal representation of memory components can develop rate code. Another hint of component formation is provided by a theory of human visual

* Corresponding author. Tel.: +36 1 209 0555x8473; fax: +36 1 381 2140.
E-mail address: andras.lorincz@elte.hu (A. Lőrincz).

0928-4257/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jphysparis.2005.09.007

recognition [18] claiming that recognition works through the recognition of the components. Now, there is a growing number of experimental and theoretical evidences in support of the claim that sensory information is compressed into components as follows: the actual sensory information is represented by a small portion of the neurons, i.e., the code is sparse, and the individual neurons represent different components of the input. Different inputs correspond to different combinations of the components, that is, the code is factorial. For example, in the primary visual cortex, receptive fields of neurons can be reproduced by independent component analysis [19–21] (for a recent review, see e.g., [22]). Other type of factorizations, which are subject to the constraint of non-negativity on neuronal activities as well as on synaptic weights have also been suggested in the literature [23,24]. Such non-negative factorization seems to fit the neuronal properties in the inferotemporal cortex. Here, neurons are mostly selective for moderately complex object features [25] and complex objects are represented by the combinations of columns [26]. Also, neurons show greater sensitivity to non-accidental (i.e., component-like) than to metric (e.g., volumetric) shape differences [27]. In our studies, we applied non-negativity constraint both on neural activities and on synaptic weights.

In our model, information is represented by series of spike sets, spikes are selected by a stochastic winner take all mechanism. Rate code—the averaging of spikes in a moving window of time—forms the internal representation. Non-negativity constraints enables to use entropic prior [28] on the internal representation, which prior biases the searches towards components whose activity (in the internal representation) can survive spike rate averaging and, therefore, are smooth.

The paper is organized as follows. The working and encoding of the reconstruction network architecture is described in Section 2. Results of computer simulation on natural images are presented in Section 3. Numerical experiments on structure extraction using coupled networks are also presented in this section. Connections to a sparse coding algorithm and the interpretation of the results are provided in Section 4. Conclusions are drawn in Section 5.

2. Methods

Reconstruction networks, in general, are designed to reconstruct the input by means of an internal representation. Our proposed architecture can include one subnetwork or more than one parallel, independently working, but jointly reconstructing subnetworks (Fig. 1B). In this architecture,

(1) a subnetwork is a reconstruction network made of three distinct layers: the internal layer, the reconstruction error layer and the bottom-up processed reconstruction error layer (Fig. 1A),

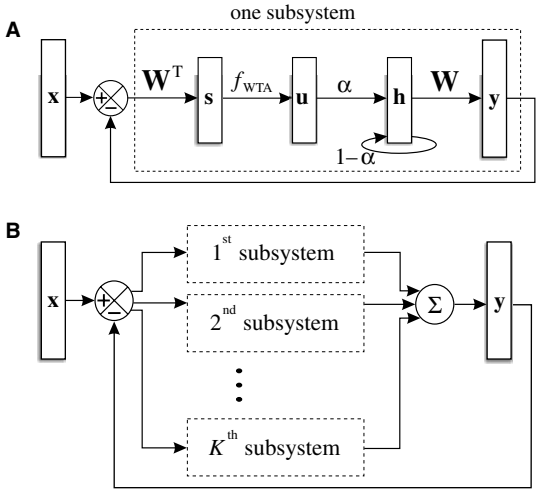

Fig. 1. *Illustration of the architecture*. (A) Basic reconstruction network unit: Input $\mathbf{x}$ enters the network. Internal representation $\mathbf{h}$ generates the reconstructed input ($\mathbf{y}$) through the top-down transformation matrix $\mathbf{W}$ ($\mathbf{y} = \mathbf{W}\mathbf{h}$). The first layer contains the reconstruction error $\mathbf{e} = \mathbf{x} - \mathbf{y}$ denoted by a circle containing the + and − signs. The reconstruction error is processed by the bottom-up transformation matrix $\mathbf{W}^{\mathrm{T}}$ producing bottom-up processed reconstruction error $\mathbf{s} = \mathbf{W}^{\mathrm{T}}\mathbf{e}$. $f_{\mathrm{WTA}}(\cdot)$ selects the most relevant component of $\mathbf{s}$:$\mathbf{u} = f_{\mathrm{WTA}}(\mathbf{s})$. We can consider $\mathbf{s}$ and $\mathbf{u}$ together as the second layer: $\mathbf{u} = f_{\mathrm{WTA}}(\mathbf{W}^{\mathrm{T}}\mathbf{e})$. The third layer contains the internal representation $\mathbf{h}$, which is updated in each cycle with the new value of $\mathbf{u}$ by temporal integration in a moving window (Eq. (2)). (B) The architecture is made of K parallel working basic units (subnetworks), having a single reconstruction error layer, which is the only coupling between the networks. To each input, the full network iterates until convergence or until reaching the maximum iteration number.

(2) computations in subnetworks is subject to non-negativity constraint, and
(3) learning in subnetworks is biased by costs and priors that minimize the reconstruction error and (indirectly) the entropy of the internal representation, respectively.

Let $\mathbf{x}(t) = (x_1(t), \ldots x_n(t)) \in \mathbf{R}_+^n$ denote the input and $\mathbf{y}(t,\tau) \in \mathbf{R}_+^n$ the reconstructed input, where $\mathbf{R}_+$ is the set of non-negative real numbers, t and τ denote the tth input and the τth internal iteration cycle within a subnetwork, respectively. When it is not confusing, indices t and τ will be neglected. The input $\mathbf{x}$ is approximated (reconstructed) by the internal representation $\mathbf{h}(t,\tau) \in \mathbf{R}_+^r$ and the top-down (TD) generative transformation matrix $\mathbf{W}(t,\tau) \in \mathbf{R}_+^{n\times r}$: $\mathbf{x}(t) \approx \mathbf{y}(t,\tau) = \mathbf{W}(t,\tau)\mathbf{h}(t,\tau)$ (Fig. 1A). For simplicity, the transposed form ($\mathbf{W}^{\mathrm{T}}$) of the same matrix is used for bottom-up (BU) transformations. The r columns of TD matrix $\mathbf{W}$ contain the generative weights and are called basis vectors or memory components. Reconstruction error $\mathbf{e}(t,\tau)$ equals $\mathbf{x}(t) - \mathbf{y}(t,\tau)$.

2.1. Forming the internal representation

The BU processed reconstruction error, $\mathbf{s} = \mathbf{W}^{\mathrm{T}}\mathbf{e}(\in \mathbf{R}^r)$, undergoes non-linear transformation: $\mathbf{u} = f_{\mathrm{WTA}}(\mathbf{s})(\in \mathbf{R}^r)$, where $f_{\mathrm{WTA}}(\cdot)$ denotes the stochastic winner take all

(WTA) function and $\mathbf{u}$ is an r dimensional unit vector. Function $f_{\mathrm{WTA}}(\cdot)$ selects a 'firing unit' from the exponentiated and normalized probability distribution created from the components of vector $\mathbf{s}$

$$p_j = \frac{e^{\vartheta(\mathbf{W}^{\mathrm{T}}(\mathbf{x}-\mathbf{y}))_j}}{\sum_k e^{\vartheta(\mathbf{W}^{\mathrm{T}}(\mathbf{x}-\mathbf{y}))_k}} = \frac{e^{\vartheta s_j}}{\sum_k e^{\vartheta s_k}}, \quad j = (1,\ldots,r). \tag{1}$$

The winning component is set to 1, other components are set to zero. Parameter $\vartheta > 0$ can shape the p probability distribution: $\vartheta = 0$ yields uniform distribution, whereas $\vartheta \gg 1$ ($\vartheta \to \infty$) approximates the true WTA algorithm (i.e., the largest activity component is the winner).

The new value of the internal representation, $\mathbf{h}(t, \tau + 1)$, is computed by moving window averaging of $f_{\mathrm{WTA}}(\cdot)$ outputs that corresponds to temporal integration with exponential kernel in the infinitesimal time step limit:

$$\mathbf{h}(t, \tau + 1) = (1 - \alpha)\mathbf{h}(t, \tau) + \alpha\mathbf{u}(t, \tau + 1). \tag{2}$$

Note that $\mathbf{h}$—the rate code of $\mathbf{u}$—is influenced by the length of the moving window proportional to $1/\alpha$, where $0 < \alpha < 1$.

We make the following remarks:

Remark 1. Replacing the stochastic function $f_{\mathrm{WTA}}(\cdot)$ by the identity operation and for small values of α in Eq. (2), the update of the internal representation approximates

$$\Delta\mathbf{h}(t, \tau + 1) = \mathbf{W}^{\mathrm{T}}(\mathbf{x}(t) - \mathbf{W}\mathbf{h}(t, \tau)) = \mathbf{W}^{\mathrm{T}}\mathbf{e}(t, \tau).$$

It can be shown that in this case and without the entropic prior, the algorithm approximately minimizes the cost function $J = \frac{1}{2}\|\mathbf{x} - \mathbf{W}\mathbf{h}\|^2$.

Remark 2. If the value of θ is finite, then no matter what the values of vector $\mathbf{s}$ are, every p_j is greater than zero.

Remark 3. If the reconstruction error is zero ($\mathbf{e} = \mathbf{0}$), then $p_j = p_k$ for all j, k and function $f_{\mathrm{WTA}}(\cdot)$ draws from a uniform distribution.

2.2. Learning of the top-down matrix

For the approximation of the input $\mathbf{x}$ and for deriving learning rules for matrix $\mathbf{W}$, we have to define a cost function. It is assumed that the reconstruction error has a sparse Cauchy probability distribution: $P(\mathrm{error}) = 1/(\pi(1 + \mathrm{error}^2))$. Because Cauchy distribution is heavy tailed, it is suitable for modeling constraints of the kind that are found in images (see also [29]). An entropic prior distribution [28] is used to bias the choice of the internal representation vector $\mathbf{h}$

$$P(\mathbf{x}|\mathbf{W}, \mathbf{h}) \sim \frac{1}{1 + \|\mathbf{x} - \mathbf{W}\mathbf{h}\|^2} \tag{3}$$

$$P(\mathbf{h}) \sim \prod_{j=1}^{r} h_j^{\kappa h_j} = \prod_{j=1}^{r} e^{\kappa h_j \log(h_j)} = e^{\kappa \sum_{j=1}^{r} h_j \log(h_j)} = e^{-\kappa H(\mathbf{h})}, \tag{4}$$

where $\kappa > 0$ is a constant and $H(\cdot)$ denotes the discrete entropy function. Combining Eqs. (3) and (4), one has

$$P(\mathbf{x}, \mathbf{h}|\mathbf{W}) \sim \frac{1}{1 + \|\mathbf{x} - \mathbf{W}\mathbf{h}\|^2} e^{-\kappa H(\mathbf{h})}. \tag{5}$$

Note, however, that Bayes rule is not applicable here, because the randomness of $\mathbf{h}$ is influenced by the stochastic WTA process of Eq. (1). However, assume that the algorithm is applied for fixed $\mathbf{x}$. The shorter the internal iteration cycle indexed by τ, the better this assumption is. Assume further that $\mathbf{W}$ is also fixed and that the internal representation converges to a stationary stochastic process quickly. Then stochastic process $\mathbf{e} = \mathbf{x} - \mathbf{W}\mathbf{h}$ becomes stationary, too. Because distributions of $\mathbf{e}(t)$ and $\mathbf{h}(t)$ clearly depend on $\mathbf{W}$, the task is to find weight matrix $\mathbf{W}$, which is the most probable, provided that (i) distribution of $\mathbf{e}(t)$ is the Cauchy distribution and (ii) distribution of $\mathbf{h}(t)$ follows the entropic prior. Then the optimization will approach a Cauchy distribution for $\mathbf{e}(t)$ and an entropic distribution for $\mathbf{h}(t)$. In turn, Eq. (5) has to be maximized or, alternatively, its inverse—which we shall take as our cost function—has to be minimized

$$J_{\mathbf{W}} = \left(\|\mathbf{x} - \mathbf{W}\mathbf{h}\|^2\right) e^{\kappa H(\mathbf{h})}. \tag{6}$$

Note that the additional constant 1 has been removed, but this change has no effect on the minimum of $\mathbf{W}$. Gradient descent provides the following update rule for matrix $\mathbf{W}$:

$$\mathbf{W}(t, \tau + 1) = \mathbf{W}(t, \tau) + \gamma e^{\kappa H(\mathbf{h}(t, \tau+1))} (\mathbf{x}(t) - \mathbf{W}(t, \tau)\mathbf{h}(t, \tau + 1))\mathbf{h}(t, \tau + 1)^{\mathrm{T}}, \tag{7}$$

where γ is a constant and corresponds to the usual learning rate. After each upgrade, $\mathbf{W}$ is rectified by applying the non-linearity $W_{kl} = \max(0, W_{kl})$ to enforce the non-negativity constraint. The *effective* learning rate depends on γ *and* on the entropy of the internal representation: $\gamma e^{\kappa H(\mathbf{h}(t, \tau+1))}$. The entropy modulation is influenced by parameter κ. Matrix $\mathbf{W}$ may undergo large modifications for large entropies occurring when values of vector $\mathbf{h}$ are of similar magnitude. Fine tuning occurs if the reconstruction error is small and if the entropy is small, i.e., when the h_i values are non-uniform but sparse. In turn, the update will facilitate the network to *escape* from regions of high entropy internal representations and minimization of the entropy of the internal representation is indirectly comprised into the learning rule of $\mathbf{W}$. In our learning procedure, prior knowledge does not effect the direction of tuning but increases or decreases the learning rate dynamically, hereby increases or decreases the exploration time of different domains of the state space.

Another view of the entropic prior says that the prior effectively reduces the search space to regions of low

entropy. The explored search space depends exponentially on the learning rate and the entropic prior may give rise to exponential gains in search time. It might be worth noting that in cases where the gradient of the $J_{\mathbf{W}}$ objective function is either not tractable or not available, one could also use, for example, the following update:

$$\mathbf{W}(t,\tau+1) = \mathbf{W}(t,\tau) + \gamma J_{\mathbf{W}(t,\tau)}\mathbf{R}(t,\tau),$$

where $\mathbf{R}(t,\tau)$ is a random matrix of maximal rank and of unit norm. This type of learning makes a biased random search, where regions of lower costs are better explored.

2.3. Learning of top-down matrices for parallel working subnetworks

The full architecture is made of K subnetworks, all of which generate their reconstruction vectors independently (Fig. 1B). The reconstruction process is the only coupling between the subnetworks. Reconstruction vectors are added up: $\mathbf{y} = \sum_{i=1}^{K}\mathbf{y}^{(i)} = \sum_{i=1}^{K}\mathbf{W}^{(i)}\mathbf{h}^{(i)}$, where superscript i denotes the ith subnetwork. The formation of the internal representation, $\mathbf{h}^{(i)}$, in each subnetwork is as described in Section 2.1. Each subnetwork works on the common reconstruction error vector $\mathbf{e}$, which is calculated as previously: it is the difference between input $\mathbf{x}$ and reconstructed input $\mathbf{y}$. This kind of coupled operation biases the search towards finding additive positive components in the input space. $P(\mathbf{x},\mathbf{h}|\mathbf{W})$ of Eq. (5) is modified as follows:

$$\begin{aligned} &P\left(\mathbf{x},\mathbf{h}^{(1)},\ldots,\mathbf{h}^{(K)} \mid \mathbf{W}^{(1)},\ldots,\mathbf{W}^{(K)}\right) \\ &\quad = P\left(\mathbf{x}|\mathbf{W}^{(1)},\ldots,\mathbf{W}^{(K)},\mathbf{h}^{(1)},\ldots,\mathbf{h}^{(K)}\right)P\left(\mathbf{h}^{(1)},\ldots,\mathbf{h}^{(K)}\right) \\ &\quad \sim \frac{1}{1+\left\|\mathbf{x}-\sum_{i=1}^{K}\mathbf{W}^{(i)}\mathbf{h}^{(i)}\right\|^2}\prod_{i=1}^{K}\prod_{j=1}^{r}h_j^{(i)^{\kappa h_j^{(i)}}} \\ &\quad = \frac{1}{1+\left\|\mathbf{x}-\sum_{i=1}^{K}\mathbf{W}^{(i)}\mathbf{h}^{(i)}\right\|^2}\mathrm{e}^{\kappa\sum_{i=1}^{K}\sum_{j=1}^{r}h_j^{(i)}\log\left(h_j^{(i)}\right)} \\ &\quad = \frac{1}{1+\left\|\mathbf{x}-\sum_{i=1}^{K}\mathbf{W}^{(i)}\mathbf{h}^{(i)}\right\|^2}\mathrm{e}^{\sum_{i=1}^{K}-\kappa H(\mathbf{h}^{(i)})} \end{aligned} \tag{8}$$

and we have the following gradient update rule (for the ith TD matrix $\mathbf{W}^{(i)}$):

$$\Delta\mathbf{W}^{(i)} = \gamma\mathrm{e}^{\kappa H(\mathbf{h}^{(i)})}\left(\mathbf{x}-\sum_{j=1}^{K}\mathbf{W}^{(j)}\mathbf{h}^{(j)}\right)\mathbf{h}^{(i)^{\mathrm{T}}}. \tag{9}$$

Note that in each subnetwork only the corresponding entropy term modulates learning and, in turn, modulation is local: A subnetwork of low entropy may stay stable while others of high entropy may undergo learning. The pseudo code of the algorithm is provided in Fig. 2. The MATLAB source code is available at http://people.inf.elte.hu/botond/structure.

```
Initialize W^(i), h^(i).
(1) τ := 1
(2) for all i = 1...K:
    e^(i)(t,τ+1) = f( e^{ϑ W^(i)(t,τ) (x(t,τ) − Σ_{i=1}^K W^(i)(t,τ) h^(i)(t,τ))} )
    h^(i)(t,τ+1) = (1−α) h^(i)(t,τ) + α σ^(i)(t,τ+1)
    W^(i)(t,τ+1) = W^(i)(t,τ) + γ e^{κH(h^(i)(t,τ+1))}
                   ( x(t) − Σ_{j=1}^K W^(j)(t,τ) h^(j)(t,τ) )
                   h^(i)^T(t,τ+1)
    W_kl^(i)(t,τ+1) = max( 0, W_kl^(i)(t,τ+1) )
If 'convergence' of all W^(i):
    W^(i)(t+1,1) = W^(i)(t,τ)
    h^(i)(t+1,1) = h^(i)(t,τ)
    t = t+1
    Go to (1)
else:
    τ = τ+1.
    Go to (2)
```

Fig. 2. Pseudo code of the algorithm.

3. Results

3.1. The working of the architecture

The first simulation was designed to demonstrate the working of the architecture on a single network ($K=1$). Every input of the network was a 64(=8 × 8) dimensional vector forming one of the 8 vertical bars of an 8 by 8 square. Applying the learning rule Eq. (7), for TD matrix $\mathbf{W}\in\mathbf{R}^{64\times 8}$, the 8 columns of matrix $\mathbf{W}$ learned to represent each of the 8 vertical bars one-by-one. That is, each column of the TD matrix consisted of 56 small valued components and 8 components close to value 1 that formed a vertical bar when rearranged into a square.

In order to reveal the dynamic properties of this network, suppose first, that the test input of the network corresponds to the ith column of the TD memory matrix. In this case, the ith component of $\mathbf{h}$ is 'excited' and will 'fire' vigorously. (Here, 'excited' means that a component of the internal representation is selected by the WTA function with high probability and the output of that component is set to one, i.e., the neuron 'fires'.) Later, spiking rate decreases and not considering the noise produced by spiking, moving window averaging will converge if the input is steady (Fig. 3A). Spike rate is the largest at presenting of the input and decreases afterwards. This effect—which is barely visible in Fig. 3A—becomes more pronounced for lower α values (Eq. (2)). Components of the internal representation belonging to other columns of the TD matrix ($h_j, j\neq i$) are not necessary for reconstruction but their

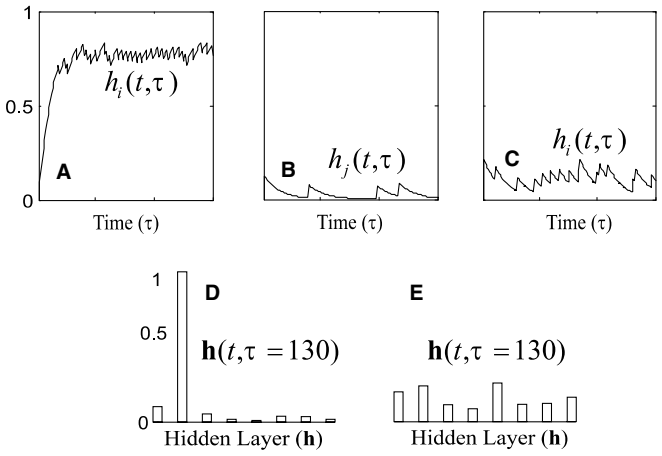

Fig. 3. *Activity types.* Top row: History of activities of different components of the internal representation $\mathbf{h}(t,\tau)$ for the tth input. The internal representation integrates spikes by exponential kernel moving window method (Eq. (2)). Number of iterations for τ: 130. Input for subfigures A and B: the ith column of matrix $\mathbf{W}$. A (B): the spike rate of the ith (jth, $j \neq i$) component is increasing (decreasing) by time. Input for subfigure (C) is not contained by the columns of matrix $\mathbf{W}$. Bottom row: Values of components of the internal representation $\mathbf{h}(t,\tau)$ at $\tau = 130$. D (E): Temporally averaged activities for the tth input, (not) belonging to the subspace of matrix $\mathbf{W}$.

activities do not disappear (Fig. 3B), since firing is maintained by two reasons: (i) the drawing function f_{WTA} is non-deterministic, i.e., it draws from a probability distribution and (ii) in case of a small reconstruction error, probabilities p_j $(j = 1,\ldots,r)$ are approximately uniformly distributed, i.e., all components have similar chances to fire. Different situation emerges, when the input does not correspond to the TD matrix, i.e., it is not within the subspace of $\mathbf{W}$. Now, all components may contribute to the reconstruction process (Fig. 3C). In this case, firing rates of the elements of internal representation $\mathbf{h}$ may become uniformly distributed. Fig. 3D and E summarize the results: (i) Large (small) amplitude firing rate corresponds to 'directly' excited (not excited) internal component (Fig. 3D), whereas (ii) approximately uniformly distributed low firing rates (Fig. 3E) correspond to inputs not represented by TD matrix $\mathbf{W}$.

3.2. Natural input patches

A single subnetwork was tested on a database consisting of natural images available at http://redwood.ucdavis.edu/bruno/sparsenet.html. The images had been preprocessed by a spatial filter that approximately whitened the data [30]. To avoid the artifacts that could arise for orthogonal grids [30], the pixels of a 13×13 window, arranged on a hexagonal grid were used for sampling. The values of the sampling points were interpolated from neighboring image pixels of the square grid of the original image. Positive and negative values of each image patch were separated into distinct channels (similarly but not analogously to ON and OFF channels of visual processing) and the representation was doubled to form a $2 \times 169 = 338$ dimensional vector. If one of the ON (OFF) components of the input vector was positive then the corresponding OFF (ON) component was set to zero.

The subnetwork can produce an overcomplete representation, i.e., the number of basis vectors (also called filters) can be greater than the effective dimensionality of the input (see also [31–33,29] and references therein). Fig. 4 shows 289 (>169) combined ON and OFF basis vectors that were developed by the algorithm. Each component of the $13 \times 13 = 169$ dimensional basis vector corresponds to different pixels of the hexagonal grid. For visualization purposes the values of the hexagonal grid are interpolated between the nodes. The sensitive regions of the basis vectors resemble spatial receptive fields of simple cells: They are oriented and localized filters with different spatial frequencies. The three small subfigures of Fig. 4 represent an 'ON', the corresponding 'OFF' and the combined 'ON and OFF' basis vectors (see also [34]). We note that without the entropic prior, the algorithm develops global and noise-like filters because there are many different representations, which can satisfy the constraint of reconstruction.

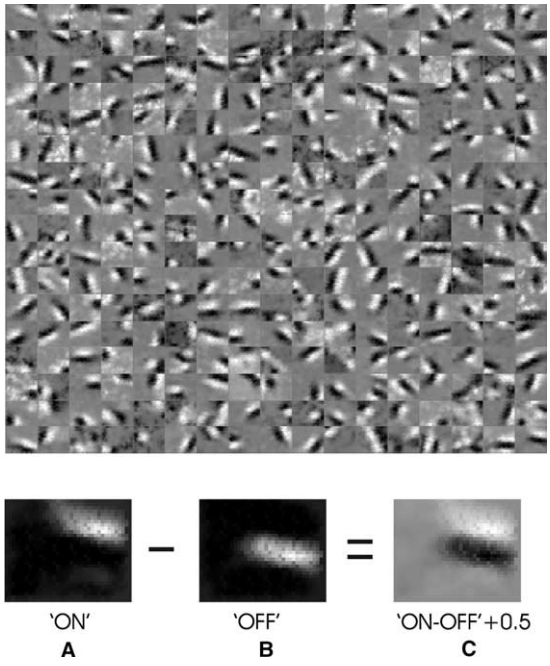

Fig. 4. *Edge filters.* All the 289 filters (basis vectors) developed by a single subsystem on 13×13 pixel sized natural image patches are presented. The basis vectors are shown by scaling all components of the 'OFF' basis vectors and the corresponding 'ON' basis vectors to take minimum value zero and maximum value 0.5 and then computing 'ON' − 'OFF' + 0.5 that makes all components to take values between 0 and 1 to form a proper gray scale. One 'ON' and 'OFF' pair, together with the corresponding combined 'ON' − 'OFF' + 0.5 basis vector is shown at the bottom. Parameter of moving temporal window: $\alpha = 0.5$. Multiplier of the entropy: $\kappa = 2.5$.

3.3. Structure finding and component sorting

Alike in the previous subsection, a 64 dimensional input space was used, but here each input consisted of two randomly chosen vertical and two randomly chosen horizontal bars. At overlapping pixels, pixel values were added. Two subnetworks ($K = 2$) were used to reconstruct and to learn the bar structure. The corresponding TD matrices are denoted by $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$. Clearly, the sparsest representation is gained if one of the TD matrices represents the horizontal bars and the other represents the vertical bars.

In one batch of experiments $\kappa = 0$ and $\vartheta = 20$ were used, i.e., the effect of entropic prior was neglected and the probability distribution of Eq. (1) approximated a perfect maximum selection mechanism. In this case, learning converged to the global minimum in approximately 20% of the experiments, but was trapped in local minima in 80% of the cases. Local minima corresponded to 7:1 or to 6:2 solutions, that is the basis vectors of one TD matrix approximated 7 or 6 horizontal (or vertical) and 1 or 2 vertical (or horizontal) bars. See inset A_1 of Fig. 5. Although noise should enable the system to escape from local minima, but still, the system stayed trapped even if the learning time was increased by an order of magnitude.

The situation has changed considerably when the entropic prior was turned on ($\kappa > 0$, Fig. 5 $t = 5000$). In this case, escape from the local minima was successful and a global minimum was found. That is, TD matrices converged to an 8:0 solutions, i.e., only horizontal or vertical bars were represented within subsystems (mark A_3 and inset A_4 in Fig. 5). The entropic prior increased the magnitude of learning steps for high entropy internal representations and decreased it for low ones. In turn, exploration was coarser (finer) for higher (lower) entropy values. Note that the entropy of the internal representation close to a global minimum (8:0 in this case) is small and it is higher around local minima (e.g., 7:1 or 6:2).

An intriguing finding is that the standard deviation of the reconstruction error drops suddenly when the entropic prior is introduced (Fig. 5A, $t = 5000$). The learned global minimum stayed stable in all of our experiments, because the contribution of the entropic prior became small. In turn, the probability of escaping from the neighborhood of a global minimum is much smaller than the probability of finding one. Thus the entropic prior makes relevant contribution in the discovery of structures. Because of the noisy operation (e.g., Remark 3) columns of TD matrices contain some noise. In order to improve the quality of the basis vectors, one might decrease the learning rate gradually by lowering the value of κ or γ (inset A_4 of Fig. 5) or both. Alternatively, weight sparsification procedure [35,36] could be used, too.

According to the computer runs, finding of a global minimum is sudden (Fig. 6, inset A_2) and it occurs with high probability within 10,000 input presentations (Figs. 5B and 6B). The case of soft competition ($\vartheta = 3$) and no contribution from entropic prior ($\kappa = 0$) is depicted in Fig. 6C. Strong modifications of matrix $\mathbf{W}$ were observed in this case. Noisy basis vectors resembling to the 8:0 solution emerged after about 30,000 learning steps. 20,000 more learning steps seemed to keep the arrangement that can be characterized as follows: (i) Basis vectors had large noise contents and were controversial. (ii) Visual inspection indicated that the basis vectors did not move out from the neighborhood of the global minimum. (iii) Both the mean and the standard deviation of the reconstruction error stayed very high and were approximately constant over the whole 50,000 time steps that would not allow for detecting a sudden improvement as in Fig. 6A. In turn, gradual decrease of the learning rate could be problematic for this case.

We have also investigated the case, when the internal representation was derived without spiking. Only the gradient descent rule (see Remark 1)

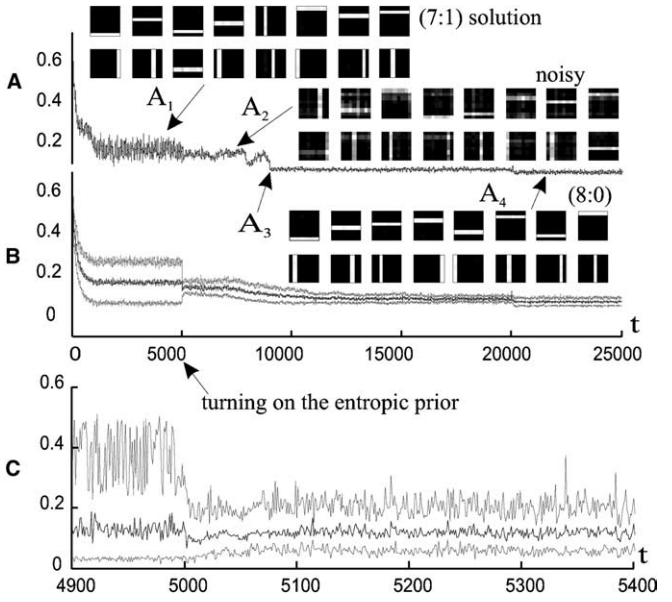

Fig. 5. *Effect of parameters on learning.* Reconstruction error (vertical axis) vs. input number (t) (horizontal axis) is shown. The insets are examples of the developed $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$ TD matrices (belonging to the marked position). The upper (lower) row of an inset: memory components of the TD matrix of the first (second) subsystem. Notation: (n:m) denotes a minimum with n horizontal (vertical) and m vertical (horizontal) bars in the first (second) TD matrix. Global minima are denoted by (8:0). Inputs consisted of vertical and horizontal bars. The iteration number of each subnetwork was 70, i.e., $\tau = 1 \ldots 70$ for each $\mathbf{x}(t)$ input. (A) One of the computer runs. (A_1) Typical results without entropic prior. (A_2) Escaping from the local minimum upon turning on the entropic prior. (A_3) sudden finding of a global minimum. (A_4) improved results for lower value of κ. (B) mean reconstruction error averaged over 50 runs (black curve), upper gray curve: mean plus the standard deviation, lover gray curve: mean minus the standard deviation. The error surface was smoothened for visualization purposes. (A) and (B) Number of inputs: 25,000. In the first 5000 input: $\kappa = 0$ (the effect of entropy was neglected). At the 5001th input: entropic prior was turned on, κ was set to 2. At the 20,001th input: κ was set to 0.8 that lowered the learning rate. (C) Reconstruction error between $t = 4900$ and $t = 5400$. Black line: mean value. Upper (lower) gray line: maximum (minimum) values of the 50 runs.

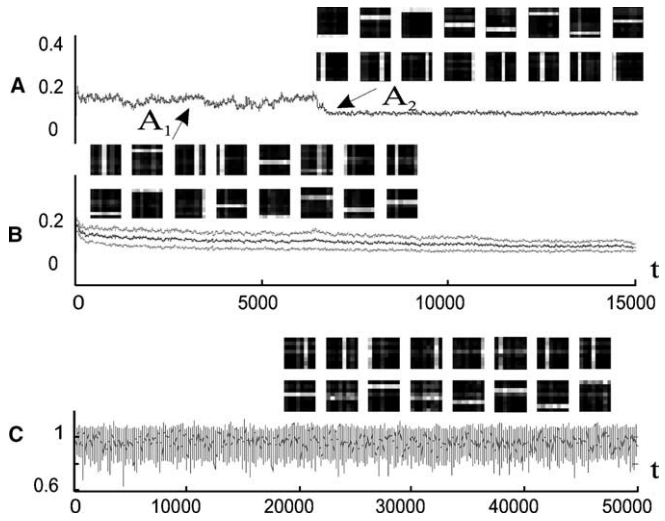

Fig. 6. *Learning with fixed parameters.* Reconstruction error (vertical axis) vs. input number (horizontal axis). The insets are examples of the developed basis sets. The upper (lower) row of an inset: memory components of the first (second) basis set. (A) One of the computer runs. (B) Averaged values for 50 runs. (A) and (B) Number of inputs: 15,000. Entropy parameter: $\kappa = 2$. (C) Soft competition and no entropic prior. Number of inputs: 50,000, $\kappa = 0$, $\vartheta = 3$. Note also the differences in the vertical scales of the subfigures.

$$\Delta \mathbf{h}^{(i)}(t, \tau + 1) = \mathbf{W}^{(i)^{\mathrm{T}}} \left(\mathbf{x}(t) - \sum_{j=1}^{K} \mathbf{W}^{(j)} \mathbf{h}^{(j)}(t, \tau) \right) \tag{10}$$

and the entropic prior were applied in each (i) subsystem. Somewhat surprisingly, 4:4 and 5:3 but not better local minima solutions were found by the algorithm. Although memory components (columns of $\mathbf{W}$) represented bars and had low noise content in many distinct time intervals, still, these memory components were relatively unstable. Columns switched between different 4:4 and 5:3 solutions often. We could force the algorithm to find a global minimum by initializing the internal representation to $\mathbf{h}(t, 0) = \mathbf{W}^{\mathrm{T}}\mathbf{x}(t)$ for each t (i.e., for each input). However, this additional initialization requires the knowledge of the 'arrival time' of a new input and also requires a separate mechanism, which operates at the arrival of each input. In turn, WTA based spiking is important in our algorithm, because learning is less efficient in the non-spiking continuous situation.

4. Discussion

4.1. Competition based spiking and rate code in reconstruction networks

The relaxation of the internal representation in reconstruction networks can be slow, which fact seems to contradict neurobiology: fast transients in sensory processing suggest feedforward processing [37]. However, reconstruction networks can also be fast, if bottom-up and top-down matrices invert each other [17], because in this case, 'perfect' internal representation is formed in one step. Even if this assumption, in general, may not hold, the bottom-up processed information can still be regarded as a good prediction of future internal representation. This is also true if the transient signals are sorted out by a WTA-like mechanism providing temporally separated spikes, because the forefront of a spike train may properly approximate the relaxed representation (see Fig. 3). In the feedforward (bottom-up) part of our proposed reconstruction network, the fast WTA mechanism selects the most relevant part (component) of the input and provides an approximation of it. Therefore, the bottom-up part of the beginning of the reconstruction dynamics approximates fast feedforward signal transmission. Other fast selection methods have also been suggested in the literature. One example is rank order coding [38], which coding utilizes a particular ordering of the components. Our method relaxes the need for such knowledge: prompt decisions can be made by using the exponentiated and normalized bottom-up filtered reconstruction error, i.e., a rough probability estimation.

The model makes use of both spiking and temporal averaging. The network provides internal clues about the quality of internal representation. Such clue could be the entropy of the internal representation or the mean and the standard deviation of the reconstruction error. An intriguing possibility conjectured by the numerical simulations is that spiking operation and rate code are more efficient *together* in finding structured components. In particular, the indirect entropy minimization of the internal representation biased the search towards components whose activity can survive the time window of rate code. Such long-lived components could not be found without spiking in our numerical studies without the external knowledge of the time of input presentation and the application of a special initialization procedure for each input.

4.2. Discussion of the architecture and the computer simulations

Non-negativity constraint has been applied in our architecture for both the transformation matrices and for the internal representation. Similar constraints have been applied in other works dealing with possible algorithms of information processing in the brain [23,24,39,34].

The architecture executes two main operations, namely working and learning. Working concerns the approximation of the internal representation. A stochastic WTA layer samples the components in every iteration step. The following layer, called rate code layer, executes moving window temporal averaging on WTA spiking and approximates the perfect continuous values of the internal representation. The internal representation produces 'realistic' activity patterns: (i) spike generation is not deterministic but stochastic, (ii) an input is described by frequent spiking of a few neurons, and (iii) every neuron produces spikes

over longer time windows (see Fig. 3). Background noise is also present in the system. This noise content is a parameter of the model (ϑ in Eq. (1)). Strong drive to select the largest activity neuron as the winner of the stochastic WTA operation reduces noise. Energy consumption is limited in our network, because the WTA operation limits firing to a single spike for each subnetwork at each time step. Such limitations may have played an important role in the evolution of computational principles adopted by the nervous system (see e.g., [40] and references therein).

The other operation, i.e., learning, concerns memory component formation, which makes use of the internal representation and the reconstruction error. Our proposed architecture, similarly to other models (see e.g., [41,30,34]) develops localized and oriented filters (memory components) on natural images patches.

In another learning test, two subsystems were used. The question was whether the algorithm can group memory components and generate the lowest entropy 'code'. In a sense, we are looking for computational advantages of columnar organization. Our presumption is that columnar organization may help to overcome combinatorial explosion. Further, we assume that columnar organization is responsible for the discovery of components that serve the 'recognition by components' process [18]. Important support to the 'recognition by components' comes from physiology: complex objects are represented by the combinations of columns in the inferotemporal cortex [26]. Our computational observation is that an entropic prior acting locally and allowing considerably different learning rates in different columns is helpful and might be necessary for the development of such combinatorial codes: In our experiments, the system learned to represent different bars by different neurons. It also learned to represent bars of different directions in different columns. Such component formation may fit the experimental observations mentioned above. This point deserves further researches.

The entropy of the internal representation close to a global minimum is small. On the other hand, the entropy around a local minimum, i.e., non-appropriate grouping of components, can be significantly higher. Consequently, the dynamically changing learning rate induced by the entropic prior enables fast escapes from local minima and guides the learning procedure away from high entropy regions but enables fine gradient search in regions having low entropy representation. We found that without the entropic prior, component formation can be trapped in local minima. However, the entropic prior *and* the WTA competition *together* helped the learning process in reaching the neighborhood of one of the global minima.

It seems attractive that the quality of the memory components can be measured by *internal means*, such as the mean reconstruction error and its standard deviation. These internal measures enable unsupervised performance monitoring. In case of soft competition (i.e., small ϑ values) and for no entropic prior, these measures become noisy (Fig. 6C).

Neuronal representation that could enforce the entropic prior by Hebbian means is not seen at this moment. However, one might assume that entropic prior is a local effect, acting through, for example, local chemical substances. According to this assumption, the entropic prior acts slowly, which is in full agreement with our rate code assumption that develops the prior. The conjecture that local firing of different neurons makes learning rates higher than the same firing rate but from a single neuron, could be validated or falsified experimentally.

4.3. Relation to sparse coding

The operation executed by one subsystem can be related to the sparse coding algorithm of Olshausen and Field [31]. We have replaced the assumption of normal distribution of the reconstruction error by Cauchy distribution, which is realistic for natural data, and the non-linear sparsifying function by the entropic prior. Now, the probability distribution to be optimized can be written as

$$P(\mathbf{x},\mathbf{h}|\mathbf{W}) \sim \frac{1}{1+\|\mathbf{x}-\mathbf{W}\mathbf{h}\|^2} e^{-\kappa H(\mathbf{h})}. \tag{11}$$

The optimal TD matrix of the subsystem ($\mathbf{W}^*$) is as follows:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \left\langle \max_{\mathbf{h}} \frac{1}{1+\|\mathbf{x}-\mathbf{W}\mathbf{h}\|^2} e^{-H(\mathbf{h})} \right\rangle_{\mathbf{x}}, \tag{12}$$

where $\langle\cdot\rangle_{\mathbf{x}}$ means averaging over $\mathbf{x}$ samples. Eq. (12) is equivalent to

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \left\langle \min_{\mathbf{h}} \{\|\mathbf{x}-\mathbf{W}\mathbf{h}\|^2\} e^{H(\mathbf{h})} \right\rangle_{\mathbf{x}}. \tag{13}$$

Olshausen and Field [31] optimizes a similar expression (Eq. (13) in [31]) in two separate phases, one nested inside the other. In the first phase, the objective was minimized with respect to the internal representation for each input, whereas in the second phase, it was decreased by changing the basis vectors through a stochastic gradient algorithm averaged over many inputs. In our case, learning of the basis vectors operated *during* the relaxation of the internal representation. Another difference is that, here, instead of using a gradient descent algorithm a momentum method was applied to update the internal representation: $\mathbf{h} \leftarrow (1-\alpha)\mathbf{h} + \alpha\mathbf{u}$ (see Eq. (2)). In turn, our expression for $\mathbf{W}^*$ is

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \left\langle \|\mathbf{x}-\mathbf{W}\mathbf{h}\|^2 e^{H(\mathbf{h})} \right\rangle_{\mathbf{u}}$$

and the stochastic gradient descent update

$$\mathbf{W} \leftarrow \mathbf{W} + \gamma e^{H(\mathbf{h})}(\mathbf{x}-\mathbf{W}\mathbf{h})\mathbf{h}^{\mathrm{T}},$$

which corresponds to our learning rule of Eq. (7) follows. It may be worth noting that averaging concerns all time instants (t, τ).

Lastly, an important difference is that in our model, explicit sparsification is replaced by an implicit one that dynamically changes the learning rate.

5. Conclusions

A reconstruction network architecture has been proposed for structure finding. It was shown that a subnetwork finds localized and oriented filters on natural images. Moreover, a coupled set of subnetworks discovers low entropy grouping of components, i.e., the components were sorted into subnetworks and structure of them were found. Therefore, the architecture has promises for solving a problem subject to combinatorial explosion. A reconstruction network architecture has been proposed for structure finding. A coupled set of subnetworks was shown to discover low entropy coding. Components of the structure were found and were sorted into subnetworks by the algorithm. Therefore, the architecture has promises for solving combinatorial learning tasks. Our reconstruction model is minimal in the sense that it becomes less effective if any of the following algorithmic components are left out: (i) non-negativity constraint on the memory components as well as on the activities, (ii) local competition, i.e., WTA-like non-linear spiking operation—based on the bottom-up processed reconstruction error—within each subnetwork, and (iii) minimization of the entropy of the spike rate, i.e., of the internal representation. There is a synergy between these algorithmic components: the spiking mechanism selects the most relevant part of the input and the network approximates fast feedforward signal transmission; spikes are temporally averaged enabling rate coding; the tuning of the memory components minimizes the reconstruction error and (indirectly) the entropy of the internal representation. Beyond the demonstration of the synergy of spike and rate codes, a novel contribution of our work is the indirect minimization of the cost function. This can be achieved by modulating the learning rate as the function of the prior knowledge, e.g., the cost function. This simple trick biases the search to more promising regions and may give rise to exponential gains in the searches.

Acknowledgements

Helpful discussions with Gábor Szirtes are gratefully acknowledged. We are most grateful to Csaba Szepesvári for his notes on the formal treatment of our approximations. This work was partially supported by Hungarian National Science Foundation (Grant No. OTKA 32487).

References

[1] S. Panzeri, F. Petroni, R. Petersen, M. Diamond, Decoding neuronal population activity in rat somatosensory cortex: role of columnar organization, Cerebral Cortex 13 (1) (2003) 45–52.
[2] A. Huxley, A. Hodgkin, A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes, J. Physiol. (London) 117 (1952) 500–544.
[3] F. Rieke, D. Warland, R. Steveninck, W. Bialek, Spikes—Exploring the Neural Code, MIT Press, Cambridge, MA, 1996.
[4] S. Thorpe, D. Fize, C. Marlot, Speed of processing in the human visual system, Nature 381 (1996) 520–522.
[5] C. Keysers, D. Xiao, P. Földiák, D. Perrett, The seed of sight, J. Cognitive Neurosci. 13 (1) (2001) 90–101.
[6] R.V. Rullen, S. Thorpe, Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex, Neural Comput. 13 (2001) 1255–1283.
[7] M. Shadlen, W. Newsome, The variable discharge of cortical neurons: implications for connectivity, computation and information coding, J. Neurosci. 18 (1998) 3870–3896.
[8] M. Mazurek, M. Shadlen, Limits to the temporal fidelity of cortical spike rate signals, Nature Neurosci. 5 (2002) 463–471.
[9] E. Adrian, The impulses produced by sensory nerve endings, J. Physiol. 61 (1926) 49–72.
[10] E. Kandel, J. Schwartz, Principles of Neural Science, Elsevier, New York, 1991.
[11] M. Stemmler, C. Koch, How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate, Nature 2 (6) (1999) 521–527.
[12] A.V. Egorov, B.N. Hamam, E. Fransén, M.E. Hasselmo, A.A. Alonso, Graded persistent activity in entorhinal cortex neurons, Nature 420 (2002) 173–178.
[13] M. Shadlen, W. Newsome, Noise, neural codes and cortical organization, Curr. Opin. Neurobiol. 4 (1994) 569–579.
[14] K. Zhang, I. Ginzburg, B. McNaughton, T. Sejnowski, Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells, J. Neurophysiol. 79 (1998) 1017–1044.
[15] T. Lu, L. Lang, X. Wang, Temporal and rate representations of time-varying signals in the auditory cortex of awake primates, Nature Neurosci. 4 (11) (2001) 1131–1138.
[16] A. Lőrincz, Gy. Buzsáki, The parahippocampal region: implications for neurological and psychiatric diseases, in: H. Scharfman, M. Witter, R. Schwarz (Eds.), Annals of the New York Academy of Sciences, vol. 911, New York Academy of Sciences, New York, 2000, pp. 83–111, Ch. Two-phase computational model training long-term memories in the entorhinal-hippocampal region.
[17] A. Lőrincz, B. Szatmáry, G. Szirtes, Mystery of structure and function of sensory processing areas of the neocortex: a resolution, J. Comput. Neurosci. 13 (2002) 187–205.
[18] I. Biederman, Recognition-by-components: a theory of human image understanding, Psychol. Rev. 94 (1987) 115–147.
[19] A.J. Bell, T.J. Sejnowski, The 'independent components' of natural scenes are edge filters, Vision Res. 37 (23) (1997) 3327–3338.
[20] J.H. Hateren, D.L. Ruderman, Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex, Proc. Roy. Soc. London B 265 (1998) 2315–2320.
[21] B. Szatmáry, A. Lőrincz, Independent component analysis of temporal sequences subject to constraints by lgn inputs yields all the three major cell types of the primary visual cortex, J. Comput. Neurosci. 11 (2001) 241–248.
[22] B. Olshausen, D. Field, Sparse coding of sensory inputs, Curr. Opin. Neurobiol. 14 (2004) 481–487.
[23] D. Charles, C. Fyfe, Modelling multiple cause structure using rectification constraints, Network: Comput. Neural Syst. 9 (1998) 167–182.
[24] D. Lee, H. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[25] K. Tanaka, Inferotemporal cortex and object vision, Ann. Rev. Neurosci. 19 (1996) 109–139.
[26] K. Tsunoda, Y. Yamane, M. Nishizaki, M. Tanifuji, Complex objects are represented in macaque inferotemporal cortex by the combination of feature columns, Nature Neurosci. 4 (2001) 832–838.
[27] R. Vogels, I. Biederman, M. Bar, A. Lőrincz, Inferior temporal neurons show greater sensitivity to nonaccidental than to metric shape differences, J. Cognitive Neurosci. 13 (2001) 444–453.

[28] M. Brand, Pattern discovery via entropy minimization, 1998. Available from: <citeseer.nj.nec.com/brand98pattern.html>.
[29] M. Welling, G. Hinton, S. Osindero, Learning sparse topographic representations with products of student-*t* distributions, in: Neural Inform. Process. Syst., Vancouver, Canada, 2002.
[30] B. Olshausen, D. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (1996) 607–609.
[31] B. Olshausen, D. Field, Sparse coding with an overcomplete basis set: a strategy employed by v1? Vision Res 37 (23) (1997) 3311–3325.
[32] A. Hyvarinen, M. Inki, Estimating overcomplete independent component bases for image windows, J. Math. Imaging Vision 17 (2002) 139–152.
[33] D. Charles, C. Fyfe, D. McDonald, J. Koetsier, Unsupervised neural networks for the identification of minimum overcomplete basis in visual data, Neurocomputing 47 (2002) 119–143.
[34] P. Hoyer, Modeling receptive fields with nonnegative sparse coding, Neurocomputing 52–54 (2003) 547–552.
[35] B. Szatmáry, B. Póczos, J. Eggert, E. Körner, A. Lőrincz, Nonnegative matrix factorization extended by sparse code shrinkage and by weight sparsification, in: F. van Harmelen (Ed.), Proceedings of the 15th European Conference on Artificial Intelligence, ECAI 2002, IOS Press, Amsterdam, NY, 2002, pp. 503–507.
[36] A. Hyvarinen, K. Raju, Imposing sparsity on the mixing matrix in independent component analysis, Neurocomputing 49 (2002) 151–162.
[37] C. Koch, T. Poggio, Predicting the visual world: silence is golden, Nature Neurosci. 2 (1999) 9–10.
[38] S. Thorpe, J. Gautrais, Computational Neuroscience: Trends in Research, Plenum Press, New York, 1998, Ch. Rank Order Coding, pp. 113–118.
[39] G. Xijin, I. Shuichi, Learning the parts of objects by auto-association, Neural Networks 15 (2002) 285–295.
[40] R. Baddeley, An efficient code in v1? Nature 381 (1996) 560–561.
[41] A. Bell, T. Sejnowski, An information-maximisation approach to blind separation and blind deconvolution, Neural Comput. 7 (1995) 1129–1159.