Self-Organizing Sparse Codes

Yangqing Jia
Sergey Karayev

2010-12-16

Abstract

Sparse coding as applied to natural image patches learns Gabor-like components that resemble those found in the lower areas of the visual cortex. This biological motivation for sparse coding would also suggest that the learned receptive field elements be organized spatially by their response properties. However, the factorized prior in the original sparse coding model does not enforce this. We investigate ways of enforcing a topography over the learned codes in a locally self-organizing map approach.

1 Introduction

There is strong biological evidence for the presence of neurons whose receptive fields resemble those of Gabor functions in the lower layers of the visual system of mammals [5]. A particularly appealing theoretical explanation for their empirical presence was proposed by Olshausen and Field [12], who showed that solving an optimization problem with a sparsity regularization on the number of active coding units results in learning codes that also greatly resemble Gabor functions.

The proposed computational model, called *sparse coding*, can also be formulated as a probabilistic model with certain choices of distributions for the variables and their priors. In particular, the model describes a linear image patch formation process with a Gaussian noise assumption and a peaky, fully factorized prior on the code activations.

The assumption of a factorized prior becomes problematic when the model is considered for a hierarchical extension. If the first layer of the visual system receives image input and is modeled by sparse coding, then a model of the second layer should receive the sparse code activations as input. But if the code activations have a fully factorized uniform prior, then it is hard to formulate top-down influence of the second layer on the first. While the biological implementation of top-down influence in V1 is not clear, there is theoretical expectation of it as well as some experimental support [10].

Therefore, multiple modifications of the model have been proposed to allow for a non-factorized prior [6, 3, 7, 13]. The common thread in much of this work is the modeling of correlation between code activations. The correlations can be thought of as being due to some topography of the code space, and a seminal work is appropriately called Topographical ICA [6].

The cortical map of the visual system displays a strong topographical organization, with orientation selective columns distributed in a way such that nearby units respond to similar orientations [1]. In past work on neural networks, this kind of organization has successfully been modeled by the Self-Organizing Map [8], which spreads unit activations to nearby units and thus learns a code that tries to match the topology of the input.

Motivated by the desire for a non-factorized prior and biology-inspired topographical organization, we investigate two ways of modifying the sparse coding model. As described in section 3, we first attempt to impose a topography through the prior on the code activations. A well performing solution turns out to be almost exactly the Self-Organizing Map approach. In section 4 we show the resulting codes learned for two- and three-dimensional topographies. But first, we review related work on the subject.

2 Related Work

Self-organizing maps were formulated in the field of neural networks, and is an example of an unsupervised network. The idea has been developed in several papers, but a thorough review is available in [8]. In a self-organizing map as described there, units are connected to units of input as well as each other. The unit that has the highest linear response to the input spreads its activation to units that are connected to it, with strength of connection defined by proximity in some topology. This process tends to converge to a spatially coherent map that is well suited to the topology of the input. The structuring behavior of the self-organizing map was motivated by Kohonen following the observation that a complex information processing task seems to require organization of information into parts.

The sparse coding model is not as simple as a single-layer cortical sheet assumed by the self-organizing map, but it can be seen in this light as a two-layer sheet. The first layer is composed of code units, densely connected to the input with weights defined by the codebook. The second layer is composed of units that define a prior on the code units. To represent the original model's factorized prior, these would be connected one-to-one to the code units with no other connections.

Hyvarinen and Hoyer [6] formulate a sparse coding model that is much like this. The bottom-layer units are interpreted as simple cells that act essentially as linear filters defined by their weight vectors (in this work, there is no modeling of a noise term in the image formation process), while the top-layer units are interpreted as complex cells that perform pooling over the bottom-layer unit activations. The pooling function is linear and is evaluated over a pre-defined neighborhood set of simple cells. The objective function of the model then becomes to maximize sparsity of the top layer, which implies a peaky prior on those units (the authors consider several and find no differences). The neighborhood and the sparsity prior together impose the topography, as to maximize sparsity of the complex cells, simple cells become pooled together so as to maximize their statistical dependencies. Codebooks learned in this manner exhibit topographic organization according to location, orientation, and frequency of the codes. Notably, phase is shown not to be correlated cross codes.

One important motivation of the described model, and one that drives other proposed extensions to sparse coding, is that there are indeed statistical dependencies in the learned codes, which violates the model's independence assumptions [14]. Due to this fact, the assumption of a factorized prior cannot be correct. Garrigues and Olshausen [3] offer a possible solution in the form of including horizontal connections between the codes in the model. These are modeled through an additional layer of Gaussian scale mixture multipliers on top of the code activations such that the *de-facto* prior on the codes is a mixture of a point mass at zero and a Gaussian. This allows both modeling the desired sparse activation behavior as well as correlations between activations. The found correlations qualitatively look like neighbors in the topographic model, result in more sparse activation vectors, and do not seem to facilitate contour integration—one popular proposed

explanation for the horizontal connections seen in V1.

Another approach to hierarchical sparse coding is presented by Karklin and Lewicki [7], who replace the joint prior coefficient distribution (the prior is a generalized Laplacian distribution) with another prior in the form of an additional codebook modeling the distribution of coefficient variances in the image. The key observation motivating the authors is that while the prior distribution in sparse coding does appear to be factorized for highly variant natural images, it cannot be assumed to be independent for particular smaller image regions, which exhibit different variances for different groups of filters (for an example, think of what codes would be activated for a patch of wood grain).

All of these approaches observe that learned codebooks are not always statistically independent, and attempt to modify the structure of the prior on the code coefficients to reflect this fact. A topographic map provides an intuitive and biologically observed method for doing so.

3 Our model

3.1 A Brief Review of Sparse Coding

Proposed in [11], sparse coding aims to represent the observed data (e.g. image features) as a linear combination of a set of codes, while encouraging each observation to only employ a sparse subset of all the available codes. More formally, let $\mathbf{X} \in \Re^{D \times N}$ be the matrix of observations, where each column is a feature vector, sparse coding aims to find a set of codes $\mathbf{A} \in \Re^{D \times M}$, where each column is the feature vector for a code, and a set of linear code activations $\mathbf{S} \in \Re^{M \times N}$, where each column is the code activations for the corresponding observation, by solving the following optimization problem

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{N} ||\mathbf{X} - \mathbf{AS}||^2_{\text{Fro}} + \lambda \phi(\mathbf{S}) \tag{1}$$

where ϕ is a regularizer that encourages sparsity of its input, and λ is the weight that sets the relative influence of both terms. Usually, one may have additional constraints to balance the scale between the codes and the code activations[1]. In practice, different regularizers can be employed to achieve sparsity, and a popular choice is the L_1 regularizer taking the following form:

$$\phi(\mathbf{S}) = \|\mathbf{S}\|_{1,1} = \sum_{i=1}^{N} \|\mathbf{S}_i\|_1 \tag{2}$$

From a probabilistic view, the sparse coding can be considered as imposing a sparse prior (e.g. the Laplacian distribution in the L_1 norm case) on the code activations and a Gaussian noise assumption. Specifically, the probability of an image patch $\mathbf{x}$ is computed by integrating over all possible code activations $\mathbf{s}$:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{A}, \mathbf{s}) p(\mathbf{s}) \, d\mathbf{s} \tag{3}$$

In practice, we usually take the point estimation of $\mathbf{s}$ that maximizes $p(\mathbf{x}|\mathbf{A}, \mathbf{s})p(\mathbf{s})$ as the code activations of the image patch. We refer to [12] for a detailed description of sparse coding and its biological justifications.

[1] In [11], the authors set the variance of each code activation to be a constant value. Another popular choice is to restrict the length of each code as

$$||\mathbf{A}_i|| \leq 1 \, , \; 1 \leq i \leq N_d$$

where bA_i is the i-th column of $\mathbf{A}$.

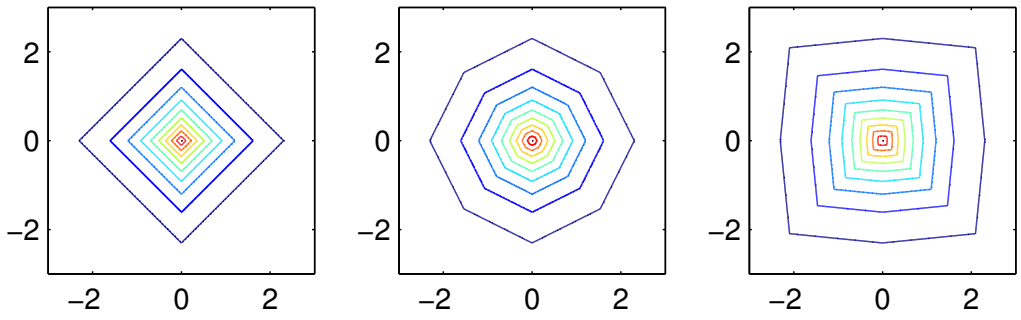


Figure 1: The contour of the prior in Equation 4. From left to right: $\lambda_2/\lambda_1 = 0, 1$ and 9.

3.2 Discounting Prior

Our first try to introduce topographical information into sparse coding is to use a non-factorial prior that reflects topographical constraints, instead of a factorial Laplacian prior employed in standard sparse coding. To this end, we introduce a novel prior term over code activations as follows:

$$p(\mathbf{s}) = \exp\{-\lambda_1 \sum_{i=1}^{M} |\mathbf{s}_i| - \lambda_2 \sum_{i=1}^{M} \max(\mathbf{s}_{\mathcal{N}_i})\} \tag{4}$$

where $s_{\mathcal{N}_i}$ denotes the activations in the neighborhood of code i (including i). Inspired by the work of Topographical ICA, we organized the codes to a grid and used 4 or 8 neighbors on the grid to define the neighborhood $\mathcal{N}_i$ for the codes.

The intuition behind this is that neighboring codes should behave similarly: if one neuron is activated for an image patch, then its nearby neurons may also get activated. The second term in the prior encourages this by only penalizing the maximum activation in the neighborhood. In another word, if one neuron activates, its neighbors get a "discount" to activate as well. Figure 1 illustrates the case of two codes connected as a neighborhood. Specifically, when the ratio λ_2/λ_1 varies from 0 to ∞, the shape of the prior varies from a L_1 norm-like prior to a L_∞ norm-like prior. The latter would encourage one code to be activated when the other code is activated, as the conditional probability of one code activation is more heavy-tailed when the activation of the other code is high. Notably, changing the prior term reduces the sparsity assumption in the neighborhood, as we may end up with a completely activated neighborhood.

3.3 Self-Organizing Sparse Codes

In addition to the topographically constrained prior, which imposes topographical constraints on the code activations we propose another method that imposes topographical constraints on the codebook. Specifically, the activation of a neuron is propagated to its neighbors, with a weight exponentially decreasing with respect to the squared distance between the two neurons. This is motivated by the work of self-organizing map (SOM): in SOM, the activation of one neuron will encourage the weights of the neighbors to move towards the datum, and this learns a way to vector

quantize a manifold of arbitrary size. The difference between our method and SOM is similar to that between sparse coding and K-means: we are learning a additive codebook that generates the data, instead of one maximum activation per datum.

To achieve this, we organize the codes as a 2D grid or a 3D cube similar to the organization in the previous subsection. The activation of neuron i with a value of $\mathbf{s}_i$ is propagated to all the neighbors of i with an exponentially decreasing weight with respect to the distance. For neighbor j, the propagated value is computed as

$$e^{-\gamma d_{ij}^2}\mathbf{s}_i \tag{5}$$

where d_{ij} is the distance between neuron i and j, computed using the Euclidean distance on the grid. After the propagation, each neuron gets an activation value that is finally used to additively construct the image patch. To disambiguate this from the original activation, we will call them "propagated activations". For a set of N data points, denote the activations of the neurons by the matrix $\mathbf{S}$ where $\mathbf{S}_{ij}$ is the activation value of the i-th neuron for the j-th datum, the propagated activations $\mathbf{S}'$ can be computed as:

$$\mathbf{S}' = \mathbf{\Psi}\mathbf{S} \tag{6}$$

where $\mathbf{\Psi}$ is an $M \times M$ matrix whose ij-th element is the exponential of the negative squared distance between neuron i and j if they are neighbors, and 0 otherwise. In addition, the diagonal of $\mathbf{\Psi}$ is always 1.

We still adopt the sparse coding prior by assuming that the code activations follow a factorial multivariate Laplacian distribution. In this way, denote the data matrix by $\mathbf{X}$, we aim to learn the codebook $\mathbf{A}$ and the activations $\mathbf{S}$ by minimizing the following loss function:

$$f(\mathbf{A}, \mathbf{S}) = \|\mathbf{X} - \mathbf{A}\mathbf{\Psi}\mathbf{S}\|_{Fro}^2 + \lambda\|\mathbf{S}\|_{1,1} \tag{7}$$

Similar to sparse coding, this problem is not convex but can be solved by alternate optimization. Specifically, when we fix $\mathbf{A}$ and solve for $\mathbf{S}$, it is a standard Lasso optimization problem by viewing $\mathbf{A}\mathbf{\Psi}$ as the codebook as a whole. When we fix $\mathbf{S}$ and solve for $\mathbf{A}$, standard gradient descent algorithms can be adopted, with the following learning rule:

$$\frac{df}{d\mathbf{A}} \propto [\mathbf{X} - \mathbf{A}\mathbf{\Psi}\hat{\mathbf{S}}]\hat{\mathbf{S}}^T\mathbf{\Psi}^T \tag{8}$$

where $\hat{\mathbf{S}}$ is the learned code activations in the previous step. The learning rule and experimental results we will show in the next section show that we will actually learn codes that look similar to its neighbors. One of the biological backgrounds is the principle of wiring economy [2]: as wiring takes up a significant fraction of the total volume of the brain and space is at a premium, it is reasonable to postulate that evolution has chosen a layout of the neurons such that neurons responsible for similar image appearances are grouped spatially.

Alternative View In the previous paragraphs, our method is described as introducing activation propagation between neighbors of the codes. In fact, an alternative view of the self-organizing sparse codes method is to consider it a sparse coding algorithm with a non-factorial prior over the activations: if we collapse $\mathbf{\Psi}$ and $\mathbf{S}$, and consider $\mathbf{S}'$ only, we can see that the prior imposed on each column of $\mathbf{S}'$, denoted by vector $\mathbf{s}'$, takes the following form:

$$p(\mathbf{s}') \propto \exp\{-\lambda\|\mathbf{\Psi}^{-1}\mathbf{s}'\|_1\} \tag{9}$$

which is different from the standard factorial prior in sparse coding where all code activations are considered independent. The dependency between the neurons are modeled by the weight matrix $\mathbf{\Psi}$, whose role is similar to the covariance matrix in the Gaussian distribution. In our paper, $\mathbf{\Psi}$ is currently hand-coded. Ideally, we would want to automatically learn the matrix Ψ from the data, which is one of the future research directions.

It is worth pointing out that our method is closely related to the newly published paper [4]. In the paper, the authors modeled the relationships between codes via group sparsity, where the groups are defined in a similar way to the nearest neighbor method in our work. We will show that our method learns similar spatially related codes, while the optimization is much simpler (although less powerful).

4 Evaluation

We used two sources of source data for learning our codebooks: the standard set of natural images [12] and the MNIST digits [9]. For learning, we modified the SPARSENET code to fit our model. Parameters were searched over qualitatively.

4.1 Natural Images - Prior Discounting Model

We first implemented the prior discounting model (see section 3), but did not obtain satisfactory results. We varied convergence rate, sparsity term weight, off-diagonal connection weights (weight for non-self connections among the codes), whether there were 4 or 8 connections from each cell, and whether the grid was extended to the opposite side at the edges. We did all this for 128 and 400 size codebooks.

In Figure 2 we present the best result of our parameter search. While some similarities between bases can be observed, it seems that neighboring codes are either almost completely the same or vastly different. We expected to see more smooth transitions between neighbors.

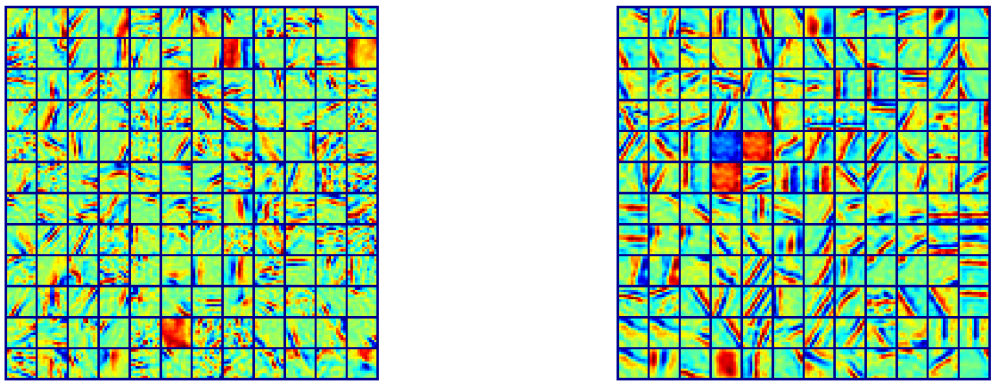
We also considered treating neighbor connections as inhibitory rather than excitatory, which would make a neighbor *less* likely to activate if a nearby code was activated. Codebooks learned in this way did not converge.

4.2 Natural Images - Self-Organizing Map

We then implemented the SOM approach, which produced better results, as can be seen in Figure 3. The neighboring codes do appear similar and they are organized in a topographically smooth way, similar to those learned from Topographical ICA. We varied the convergence rate, sparsity term weight, neighbor window size, time decay of the spread, and whether the grid was two- or three-dimensional.

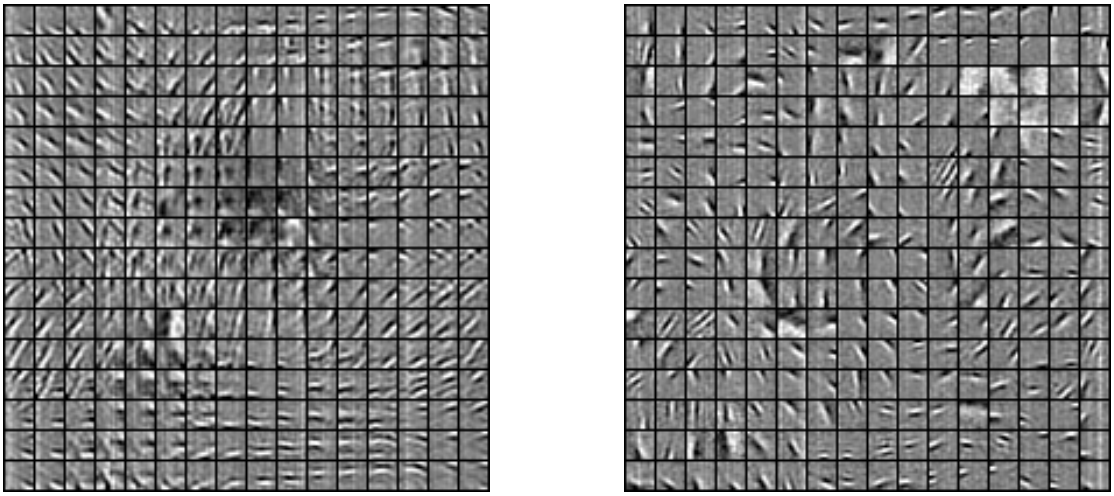
We found no difference between windows of size three and five. Time decay of the spread refers to decreasing the influence of the self-organization with iterations, and seemed to result in higher-frequency components present in the codebook.

Our hypothesis was that a three-dimensional grid would separate frequency out from location and orientation, but that did not occur, and all three properties varied in all three dimensions. Since visualizing codebooks learned in this way is problematic, we analyzed only the two-dimensional codebooks.



(a) (b)

Figure 2: Prior discounting model. Complete representation (128 bases). (a) self-connections (no discounting). (b) 8-neighbor connections (mirrored across edges) with off-diagonal excitations weight of 0.5.



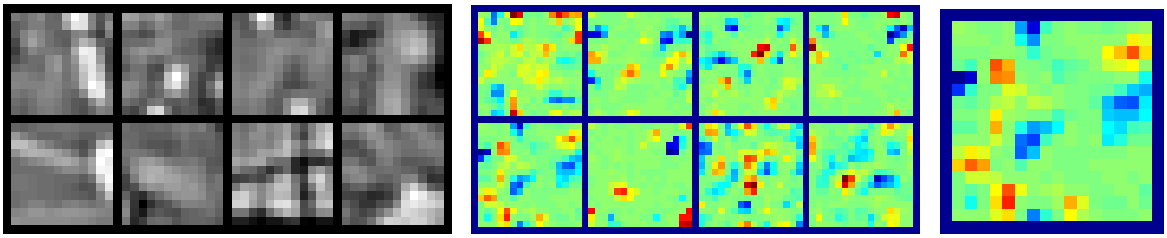
(a) (b)

Figure 3: SOM based model, with 256 bases and grid mirrored across edges. (a) 5-window connections with no weight decay. (b) 3-window connections with weight decay.

To further investigate the code activations, we learned a set of 256 codes from 12×12 image patches and investigated the activations on test image patches. Figure 4 presents the results of eight image patches, showing their appearance and the activations of the neurons (the activation vectors are reshaped to 16×16 grids). It can be observed that the activations of neurons change smoothly over the grid, which is consistent with our assumption.

It is worth pointing out that the SOM approach learns code activations that are less sparse than the standard sparse coding, as it is encouraging neighboring codes to have similar appearances and

consequently similar activations. In some sense, this can be considered as imposing a smoothness regularization over the code activations. This has some biological justifications, but whether it improves performance is still unknown. Further experiments regarding specific tasks such as image or object classification may reveal more evidence.



(a) (b) (c)

Figure 4: Code activations for the SOM approach. (a) The appearance of 8 testing image patches. (b) The code activations, each organized to a 16×16 grid. (c) Close-up of the activation for one patch.

4.3 MNIST

Similarly to natural images, we tested the SOM approach on the MNIST dataset. Each digit is represented by a 28×28 patch which is reshaped to a 784-dimensional vector. For visualization purpose, we trained our model to learn a codebook of size 256, which is undercomplete. Thus, we imposed a stronger sparsity prior to find sparse codes (otherwise the results would be much similar to that of PCA). The parameters are again chosen qualitatively.

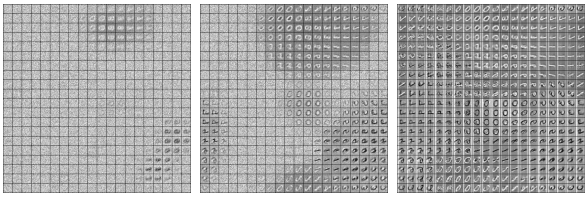
Figure 5 shows the emergence of the learned codebook from random initialization and the final codebook. It can be observed that the learned codes are also organized topographically - the topographical property can also be observed during the learning time, see Figure 5(a). One thing worth mentioning is that we did not get a codebook containing local strokes of the digits as one may expect. We infer that this may be because we used an undercomplete basis and need to impose a strong sparsity prior to obtain sparse results, making the learned code more similar to cluster centers instead of additive codes.

5 Future Direction

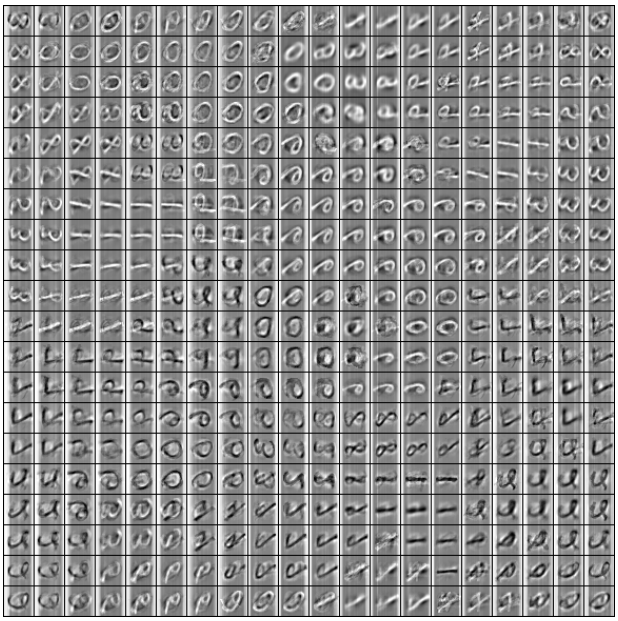
Our work contains the seed of an idea that should be developed further: representing a dictionary matrix as the product of content and connections. The next step could be to run EM optimization over the connection matrix Ψ every few iterations of the basic sparse coding optimization. While not a principled approach, this could result in an efficient way of learning the topography and consequently the covariance of sparse codes.

References

[1] Gary G Blasdel. Orientation selectivity, preference, and continuity in monkey striate cortex. *The Journal of Neuroscience*, 12(8):3139–3161, Sep 1992. 1



(a)



(b)

Figure 5: The result for MNIST. (a) The codebook at the 10th, 30th and the 50th iteration. (b) The final codebook learned from MNIST.

[2] D.B. Chklovskii. Optimal sizes of dendritic and axonal arbors in a topographic projection. *Journal of Neurophysiology*, 83(4):2113, 2000. 5

[3] Pierre J Garrigues and Bruno A Olshausen. Learning horizontal connections in a sparse coding model of natural images. *NIPS 2007*, pages 1–8, Jun 2007. 1, 2

[4] Pierre J Garrigues and Bruno A Olshausen. Group sparse coding with a laplacian scale mixture prior. *Advances in Neural Information Processing Systems 23*, pages 1–9, Oct 2010. 6

[5] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160:106–154, January 1962. 1

[6] Aapo Hyvarinen and Patrik O Hoyer. A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41(18):2413–2423, Apr 2001. 1, 2

[7] Yan Karklin and Michael S Lewicki. Learning higher-order structures in natural images. *Network: Computation in Neural Systems*, 14:483–499, Jun 2003. 1, 3

[8] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1–17, Feb 1990. 1, 2

[9] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010. 6

[10] Scott O Murray, Paul Schrater, and Daniel Kersten. Perceptual grouping and the interactions between visual cortical areas. *Neural Networks*, 17(5-6):695–705, Jul 2004. 1

[11] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, pages 1–3, Dec 1996. 3

[12] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325, May 1997. 1, 3, 6

[13] Honghao Shan, Lingyun Zhang, and Garrison W Cottrell. Recursive ica. *NIPS*, pages 1–8, Jan 2007. 1

[14] Christoph Zetzsche, Gerhard Krieger, and Bernhard Wegmann. The atoms of vision: Cartesian or polar? *J. Opt. Soc. Am. A*, 7:1554–1565, Jun 1999. 2