Building a better probabilistic model of images by factorization

Benjamin J. Culpepper
UC Berkeley
Berkeley, CA
bjc@cs.berkeley.edu

Jascha Sohl-Dickstein
UC Berkeley
Berkeley, CA
jascha@berkeley.edu

Bruno A. Olshausen
UC Berkeley
Berkeley, CA
baolshausen@berkeley.edu

Abstract

We describe a directed bilinear model that learns higher-order groupings among features of natural images. The model represents images in terms of two sets of latent variables: one set of variables represents which feature groups are active, while the other specifies the relative activity within groups. Such a factorized representation is beneficial because it is stable in response to small variations in the placement of features while still preserving information about relative spatial relationships. When trained on MNIST digits, the resulting representation provides state of the art performance in classification using a simple classifier. When trained on natural images, the model learns to group features according to proximity in position, orientation, and scale. The model achieves high log-likelihood (-94 nats), surpassing the current state of the art for natural images achievable with an mcRBM model.

1. Introduction

Many image analysis and classification tasks depend on having a good representation of the data. In recent years, significant effort has been devoted to learning these representations from the statistics of natural images. In particular, sparse coding models have been shown to yield features that match the properties of neurons in the visual cortex and also yield improved performance at image analysis and classification tasks [38, 1, 39, 16, 30, 19, 18, 28, 26]. However, the coefficients resulting from sparse coding typically exhibit strong statistical dependencies and can change abruptly in response to even small changes in the input. Presumably, methods that capture these dependencies and more explicitly represent the group structure among coefficients should provide more stable representations and hence better performance in classification. One approach to modeling these dependencies is based on subspace models, which impose a prescribed group structure on the coefficients by penalizing activations of groups rather than individual members within a group [13, 3, 9]. However, this approach is still unsatisfying because the group structure is not learned. Recent attempts to learn the group structure have been made in undirected models [17, 30, 31, 6]. Here, we do the same in a directed model.

Previous work on modeling image transformations [21, 25, 22], separating content and style [7, 10, 11], and capturing higher-order dependencies in natural images [14, 15, 27], draws upon a simple but powerful bilinear model, in which an observation $\mathbf{x}$ is described by a mixture of basis elements with pairs of multiplicative coefficients $\mathbf{c}$ and $\mathbf{d}$:

$$x_l = \sum_{jk} \Gamma_{jkl}\, c_j\, d_k \,. \tag{1}$$

This model was shown to be capable of separating an object's identity (or 'content') from its pose ('style') with some success [7]. In [7], Equation 1 is fit using an iterative, alternating optimization method based on SVD. The authors do not describe a probabilistic interpretation, but if Gaussian priors are assumed then their technique can be understood as a variational (MAP) EM based algorithm; similar approaches have also been investigated with sparse priors [10, 3, 8, 14, 9].

Using a probabilistic (Monte Carlo) method, we fit bilinear models to the MNIST handwritten digit database and image patches drawn from natural scenes. In MNIST, we show that improvements in the model likelihood under hold-out data correlate with improvements in a classification task. In natural scenes, the parameters learn sensible group dependency structures, and samples from the models capture several salient statistical properties of natural image patches, including their contrast variance and long-range correlations (phase alignments of edges).

2. Model

The model is formulated in terms of two layers of representation, as illustrated in Figure 1. The first layer represents the image data in terms of a set of features using a linear generative model:

$$\mathbf{x} = \mathbf{\Phi}\, \mathbf{a} + \mathbf{n} \,, \tag{2}$$

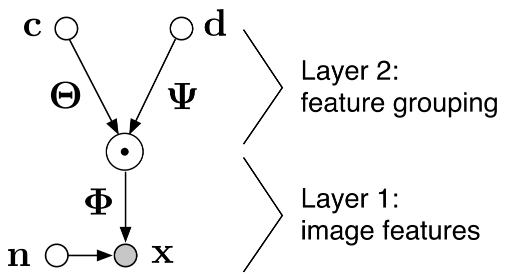

Figure 1. The bilinear model is formulated in terms of two layers of representation. The first layer represents the image data in terms of a set of features $\mathbf{\Phi}$. The second layer factorizes the first layer representation in terms of feature groups ($\mathbf{\Psi}$) represented by latent variables $\mathbf{d}$, and their relative activations ($\mathbf{\Theta}$) represented by latent variables $\mathbf{c}$.

where $\mathbf{x} \in \mathbb{R}^L$ is a vector representing the image data, such as pixels, or PCA coefficients. $\mathbf{\Phi} \in \mathbb{R}^{L \times M}$ is a matrix of features, and $\mathbf{n} \in \mathbb{R}^L$ is Gaussian noise with variance σ^2, $\mathbf{n} \sim \mathcal{N}(\mathbf{\Phi}\,\mathbf{a}, \sigma^2\,\mathbf{I})$. The second layer is a factored representation of $\mathbf{a}$:

$$\mathbf{a} = \mathbf{\Theta}\,\mathbf{c} \odot \mathbf{\Psi}\,\mathbf{d}\,, \tag{3}$$

with $\mathbf{\Theta} \in \mathbb{R}^{M \times J}$, $\mathbf{\Psi} \in \mathbb{R}^{M \times K}$, and $\odot$ denoting an element-wise product. The prior distributions on the latent variables $\mathbf{c}$ and $\mathbf{d}$ impose a functional asymmetry between what is learned by $\mathbf{\Theta}$ and $\mathbf{\Psi}$. The intention is for $\mathbf{\Psi}$ to learn how groups of features co-activate (their presence or absence), while $\mathbf{\Theta}$ learns the relative amplitudes of features after the part modeled by $\mathbf{\Psi}\,\mathbf{d}$ has been factored out. The $\mathbf{d}$ variables should represent patterns of co-activation that are invariant to local perturbations of shape in the image, while the variations in shape are represented by the $\mathbf{c}$ variables.

The $\mathbf{c}$ variables are encouraged to explain variability by the use of an isotropic Gaussian prior, which is rotationally invariant and non-sparse:

$$\mathbf{c} \sim \mathcal{N}(0, \mathbf{I})\,. \tag{4}$$

The $\mathbf{\Psi}$ matrix is encouraged to learn meaningful, frequently occurring patterns in the image feature activations by selecting a prior distribution for the $\mathbf{d}$ variables that prefers directions aligned with the coordinate axes. The following choices are explored:

$$\mathbf{d} \sim \mathcal{E}(\mathbf{I}) \tag{5}$$
$$\mathbf{d} \sim \mathcal{L}(\mathbf{I}) \tag{6}$$
$$\log \mathbf{d} \sim \mathcal{N}(0, \mathbf{I}) \tag{7}$$
$$\log \mathbf{d} \sim \mathcal{N}(0, \mathbf{\Omega}\,\mathbf{\Omega}^{\mathrm{T}}) \tag{8}$$

where $\mathcal{E}(\mathbf{I})$ and $\mathcal{L}(\mathbf{I})$ denote factorial exponential and Laplace distributions with rate parameter one.

This two-layer hierarchical structure can also be viewed as a factorization of the parameter tensor $\mathbf{\Gamma}$ from Equation 1 into the product of three matrices,

$$\Gamma_{jkl} = \sum_m \Phi_{lm}\,\Psi_{mj}\,\Theta_{mk}\,. \tag{9}$$

This factorization reduces the number of model parameters from JKL to $M(J + K + L)$, where L is the number of values in the image data, J is the number of content dimensions, K is the number of style dimensions, and M is the number of factors coupling each l, j, and k. This technique has recently been used successfully by [22, 31] to find reduced dimensionality parameterizations that are reasonably matched to structure in the data being modeled.

Finally, note that if $\mathbf{\Psi} = \mathbf{I}$ and $\mathbf{\Theta} = \mathbf{I}$, this model takes the form of a Gaussian scale mixture [36].

3. Model estimation and evaluation

Since $\mathbf{c}$ and $\mathbf{d}$ are latent, we use Expectation Maximization (EM) to fit model parameters $\Lambda \equiv \{\mathbf{\Phi}, \mathbf{\Psi}, \mathbf{\Theta}, \mathbf{\Omega}\}$. In the E-step we update samples from the posterior distribution computed at the previous time step $Q^{(t)}(\mathbf{c}, \mathbf{d}|\mathbf{x})$ using a Hamiltonian Monte Carlo sampler, described in the next section. In the M-step, maximization is performed using L-BFGS [2].

3.1. Sampling from the posterior

$P(\mathbf{c}, \mathbf{d}|\mathbf{x}, \Lambda)$ is sampled using a variation on a Langevin dynamics sampler with partial momentum refreshment. The underlying technique is introduced in [12], and is clearly presented in [24] (Sections 5.2 and 5.3). The momentum refreshment rate β is set such that half the momentum power is replaced per unit simulation time. Thus, the update to the momentum $\mathbf{p}$, applied after every leapfrog step, is given by

$$\mathbf{p}' = -\sqrt{1-\beta}\,\mathbf{p} + \sqrt{\beta}\,\mathbf{r}, \tag{10}$$
$$\beta = 1 - \exp\left(\epsilon\,\log\left(1/2\right)\right)\,, \tag{11}$$

where $\mathbf{p}'$ is the new momentum, $\mathbf{r} \sim \mathcal{N}(0, \mathbf{I})$, and ϵ is the length of each leapfrog simulation step.

Rather than sampling from the posterior repeatedly for each item, an entire batch of data is loaded into memory, and one particle for each data item is evolved simultaneously. This avoids a significant number of burn-in steps that would be required if the particles were initialized with randomized positions and momenta on each iteration of E-M. Instead, the sampling algorithm is initialized once, at the beginning of learning, by loading a large set of data $\{\mathbf{x}_b\}_{b=1..B}$, allocating space for one particle (represented by a position and momentum) per data item, and initializing the position and momentum by drawing from their respective priors. The position consists of the $\mathbf{c}$ and $\mathbf{d}$ together. The sampling timestep ϵ and the number of sampling iterations τ between M steps are chosen at the beginning of learning and held fixed.

With $\epsilon = 0.01$, only a slight discretization error is incurred, and nearly all update steps are accepted. We therefore skip the rejection step entirely, reducing the computations per update step. We measure the likelihood of the solutions we obtain [33], and have confirmed that this approximation does not reduce the quality of the learned model, and sometimes decreases the number of learning steps required as well as the time per learning step. When samples are far from equilibrium, even rare momentum reversals can slow their approach to the equilibrium distribution.

Since the exponential distribution is defined only over the positive real interval, there remains the practical matter of handling the constraint at zero, which we accomplish by negating the position and momentum for $\mathbf{d}$ variables when their position becomes negative after running a step in the dynamics, as described in [24] (Section 5.1 and Figure 8).

3.2. Maximization with respect to Λ

Maximizing the lower bound $\mathcal{L}(Q, \Lambda)$ w.r.t. Λ amounts to maximizing $\int Q(\mathbf{c}, \mathbf{d}|\mathbf{x}) \log P(\mathbf{x}, \mathbf{c}, \mathbf{d}|\Lambda)\, d\mathbf{c}\, d\mathbf{d}$. This is equivalent to minimizing the energy, E, averaged over the samples $\arg\min_\Lambda \frac{1}{B} \sum_{b=1}^{B} E(\mathbf{q}_b)$, which is accomplished using L-BFGS [2].

3.3. Likelihood measurement

Separately, we have developed a practical, robust method for log likelihood estimation, Hamiltonian Annealed Importance Sampling [33], which is based on Hamiltonian Monte Carlo and annealed importance sampling [23]. We use this technique to evaluate the log likelihood of several image models for comparison purposes.

4. Model recovery

This E-M learning algorithm can recover model parameters used to generate artificial data. We demonstrate recovery for model parameters chosen as follows. $\mathbf{\Phi} = \mathbf{I}$, entries in $\mathbf{\Theta}$ drawn randomly from a Laplace distribution with scale parameter one, and $\mathbf{\Psi}$ set to have a non-trivial overlapping group structure: with half as many $\mathbf{d}$ variables as $\mathbf{c}$ variables, each $\mathbf{d}$ variable was coupled to three layer-one features by activating three entries of each $\mathbf{\Psi}$ column, and setting the rest to zero. The rows of $\mathbf{\Psi}$ were given unit norm. These parameter settings are illustrated in the first column of Figure 2. 2,000 training samples were generated using this model and Equation 2, with $\sigma = 0.1$, $L = M = J = 16$, and $K = 8$.

Next, the model parameters were optimized using the E-M procedure described above, starting from the following random initialization: the columns of $\mathbf{\Phi}$ were set to random unit length vectors, $\mathbf{\Theta}$ to $\mathbf{I}$, and all the entries of $\mathbf{\Psi}$ were set to $1/\sqrt{K}$. Since there is a multiplicative degeneracy between the length of the columns of $\mathbf{\Phi}$ and the rows of $\mathbf{\Psi}$ and

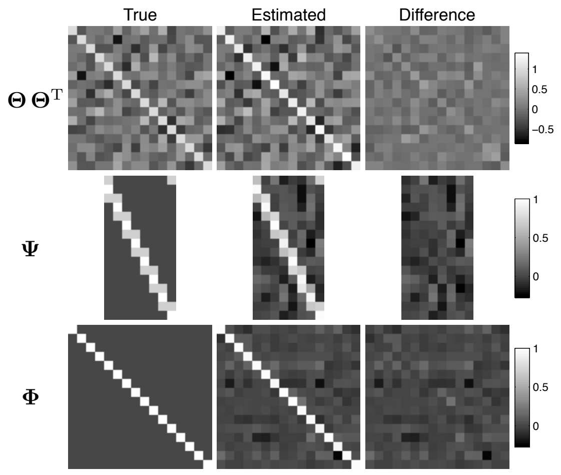

Figure 2. Model recovery. 2,000 data samples were generated from the model by setting the parameters as shown in the 'True' column, drawing from the priors for $\mathbf{c}$, $\mathbf{d}$, and $\mathbf{n}$, then generating $\mathbf{x}$ samples via Equation 2. The model parameters were then estimated by our E-M algorithm, from random initial conditions. The 'Estimated' column shows the parameters after 10,000 learning iterations. The 'Difference' column shows the difference between true and estimated parameters.

$\mathbf{\Theta}$, after every M-step we multiplied by the appropriate factors to maintain unit norm in the columns of $\mathbf{\Phi}$ and the rows of $\mathbf{\Psi}$. Figure 2 shows the parameters recovered after 10,000 iterations of E-M in the 'Estimated' column. (Most of the correct structure is apparent after only 1,000 iterations.)

5. Experiments on handwritten digits

A widely accepted hypothesis in the vision community is that improvements to statistical models of images will lead to improvements in classification tasks, object recognition tasks, and other applications such as denoising and in-painting. Though performance on the fully supervised MNIST problem is near perfect, there is no algorithm that can classify digits with as few labeled examples as a human. For this reason the MNIST dataset is well suited to demonstrate the utility of unsupervised learning for classification. Here we show that increasing the log likelihood, and complexity, of a probabilistic model of digits also increases classification accuracy.

5.1. Training

The MNIST handwritten digit recognition problem consists of 60,000 training and 10,000 testing examples, each a 784 dimensional (28x28 pixels) binary vector. Our approach to classification is to first use the 60,000 training examples – without their labels – to estimate the model parameters. The learned model should then function to factor the pixels into latent variables that better represent the causal structure of the data. Assuming that the model effectively carries out this task, the learned representation may improve accuracy in a classification task, even though the goal of classification did not play a role in the parameter estimation process.

We train one-vs-all classifiers for each of the digits, 0-9, and compute class probabilities based on the distance of a point from the margin; points are assigned to the class with the highest probability. The C-SVM formulation [5] is used, where the parameter C establishes the cost of turning on support vectors to achieve separability; $C = 10$ for all of the experiments described in this paper. Using a modified version of LIBSVM [4] that incorporates the fast approximate intersection kernel [20], we solve the SVM optimization problem, inputting the model's latent representation for each digit.

5.2. Results

First, a model with $\sigma = \sqrt{0.1}$, $L = 784$, $J = 200$, $K = 200$, and $M = 200$ is applied to the MNIST benchmark task. Each digit is preprocessed by subtracting the mean, and scaling the resulting real valued vector to have unit norm. Next, 10,000 iterations of E-M are run to learn the model parameters using $\mathbf{d} \sim \mathcal{E}(\mathbf{I})$, which takes about one day on an eight core machine. When only $\mathbf{\Phi}$ is learned, and $\mathbf{\Psi}$ and $\mathbf{\Theta}$ are set to the identity, basis functions in $\mathbf{\Phi}$ developed large-scale spatial structure, sometimes tracing out entire digits. When $\mathbf{\Psi}$ and $\mathbf{\Theta}$ are subsequently learned along with $\mathbf{\Phi}$, these more global structures disappear completely, accompanied by the emergence of non-zero off-diagonal entries learned in $\mathbf{\Psi}$ and $\mathbf{\Theta}$. After learning, MAP estimates for all of the training and testing examples are computed using L-BFGS. The representation used for classification is a concatenation of the $\mathbf{c}$ and $\mathbf{d}$ variables (i.e., 400 coordinates per digit).

Using the RBF kernel, this procedure achieves state of the art performance for knowledge-free methods: 0.96% error. During evaluation, the order of complexity of a support vector machine classifier is in general a monotonically increasing function of the product of the number of support vectors and the number of coordinates in the representation. However, certain kernels, such as the linear kernel and the fast approximate intersection kernel, have run-time complexities that depend only on the number of coordinates and are thus advantageous. For this reason, it is also worth noting that our representation achieves 1.72% error using the fast intersection kernel, superior to [40] in terms of accuracy, with the added benefit of requiring fewer coordinates.

Finally, three separate models are trained on a more difficult variation of the MNIST task, where the digits are represented by only 128 PCA components, and only 2,500 labeled training examples are utilized. This experiment reveals a correlation between average log-likelihood on the test set and classification performance. We probe the relationship between likelihood and classification in two ways: first, in the context of changing from a factorial prior to a non-factorial prior; second, in the context of dimensionality expansion (from 144 to 256). For these models, we perform different preprocessing than for the benchmark task. Here, we subtract the mean from each digit, then project onto the top 128 PCA components (of the training set), which retains 95.19% of the variance of the original digits, then whiten by rescaling each dimension to unit norm. All three models have $L = 128$, $\sigma = 0.1$, and $J = K = M$; thus, the model size can be specified by a single number. The first model has size 144, but only $\mathbf{\Phi}$ is learned – $\mathbf{\Psi}$ and $\mathbf{\Theta}$ are set to $\mathbf{I}$. The second model is identical to the first, but $\mathbf{\Phi}$, $\mathbf{\Psi}$ and $\mathbf{\Theta}$ are all learned. The third model has size 256 and all parameters are learned. Table 1 shows the log likelihood of each of these three models averaged over 100 random digits from the test set. Model one has the lowest average log likelihood, and the worst classification performance. When a non-factorial prior is learned, the average log-likelihood improves by 54 nats, and classification accuracy improves by 10.6%. When the size of the model is expanded slightly from 144 to 256, the corresponding improvements are 15 nats and 1.49%.

Model	Size	Avg. log likelihood	Accuracy
factorial	144	-161.52 ± 6.76	81.90%
full	144	-107.16 ± 4.34	92.50%
full	256	-92.23 ± 5.74	93.99%

Table 1. MNIST classification accuracy by model using a linear kernel with only 2,500 training labels and the first 128 PCs.

6. Experiments on natural images

Four bilinear models are trained on 100,000 16x16 pixel image patches taken at random from 4,112 linearized images of natural scenes from the van Hateren dataset [35]. The extracted image patches are first logged, and then mean subtracted. They are then projected onto the top 100 PCA components, which retains 97.11% of the variance of the original patches, and whitened by rescaling each dimension to unit norm. For all models, $L = 100$ (the number of PCA components), and $J = K = M$. We set σ to 0.1, initialize $\mathbf{\Psi}$ and $\mathbf{\Theta}$ to $\mathbf{I}$, and $\mathbf{\Phi}$ to random with unit norm columns, then run 40,000 iterations of E-M using $\tau = 10$, $\epsilon = 0.01$, and 8 iterations of L-BFGS in the M-step. This takes about two days on an eight core machine.

6.1. Comparison to the mcRBM

The mean and covariance restricted Boltzmann machine (mcRBM) [29] is an undirected analogue of our directed bilinear model. The high quality of the samples produced

Model	Size	$\langle \log \text{likelihood} \rangle$
linear, $\mathbf{a} \sim \mathcal{L}(\mathbf{I})$	100	-122.80 ± 2.43
bilinear, $\mathbf{d} \sim \mathcal{L}(\mathbf{I})$	100	-117.27 ± 2.66
bilinear, $\mathbf{d} \sim \mathcal{L}(\mathbf{I})$	400	-99.52 ± 2.73
bilinear, $\log \mathbf{d} \sim \mathcal{N}(0, \mathbf{I})$	100	-98.95 ± 2.28
student-t POE	400	-97.60 ± 2.55
mcRBM	400	-97.58 ± 2.67
bilinear, $\log \mathbf{d} \sim \mathcal{N}(0, \mathbf{\Omega}\,\mathbf{\Omega}^{\mathrm{T}})$	100	-94.62 ± 2.44

Table 2. Model quality comparison. Average log-likelihood is measured on a hold-out set of 1000 image patches.

by the mcRBM has motivated the exploration of similar models [6], and for these reasons it is a particularly relevant comparison. We define the mcRBM model using the marginal energy function E_{mcRBM} implemented in the released code, which differs slightly from the version given in [29]:

$$E_{\text{mcRBM}}(\mathbf{x}) = -\sum_{k=1}^{K} \log \left(1 + e^{\frac{1}{2}\sum_{l=1}^{L} P_{lk} \frac{(\mathbf{C}_l \mathbf{x})^2}{||\mathbf{x}||_2^2 + \frac{1}{2}} + b_k^c} \right)$$
$$- \sum_{j=1}^{J} \log \left(1 + e^{\mathbf{W}_j \mathbf{x} + b_j^m} \right) + \frac{1}{2\sigma^2} \mathbf{x}^T \mathbf{x} - \mathbf{x}^T \mathbf{b}^v \,, \quad (12)$$

with parameters: $\mathbf{P} \in \mathbb{R}^{L \times K}$, $\mathbf{C} \in \mathbb{R}^{L \times M}$, $\mathbf{W} \in \mathbb{R}^{J \times M}$, $\mathbf{b}^m \in \mathbb{R}^J$, $\mathbf{b}^c \in \mathbb{R}^K$, $\mathbf{b}^v \in \mathbb{R}^K$, $\sigma \in \mathbb{R}$. We trained two mcRBM models, both with $L = 100$. For the smaller model we use $K = 100$, $M = 100$; for the larger, $K = 400$, $M = 400$. Training was performed using several techniques: CD-1, CD-5, Persistent CD, Fast Persistent CD [34], and Persistent MPF [32]. The model estimated using Persistent MPF had the highest log likelihood, and is the one presented.

6.2. Comparison to the Product of Student's t-test model

The Product of Student's t [37] is a linear, undirected model for natural image patches, which performs significantly better than any current competing linear models. It takes the form:

$$P(\mathbf{x}) \propto \prod_j \frac{1}{\left(1 + (\mathbf{W}_j \mathbf{x})^2\right)^{\alpha_j}} \,, \quad (13)$$

where $\mathbf{W} \in \mathbb{R}^{J \times L}$ and $\alpha \in \mathbb{R}_+^J$. It was trained via Persistent MPF, with $J = 400$.

6.3. Results

The average log-likelihood of several models of natural image patches is compared in Table 2. When the correct prior is selected for the $\mathbf{d}$ variables, the bilinear model exceeds the performance of the mcRBM. The learned parameters for a bilinear model with $\mathbf{d} \sim \mathcal{E}(\mathbf{I})$ are shown in Figures 3 and 4. (The other priors produce qualitatively similar results.) Each column of Figure 3(a) shows 10 columns of $\mathbf{\Phi}$, organized into groups and sorted according to the strength of the corresponding weight in a column of $\mathbf{\Psi}$. The columns of $\mathbf{\Phi}$ learn to be oriented, bandpass functions. Similar to what happens with digits, in natural images we find that the spatial extent of each $\mathbf{\Phi}$ basis element shrinks when the marginal over $\mathbf{a}$ is allowed to be non-factorial. Since a single $\mathbf{d}$ variable can then control a whole group of functions, the model learns to link together edges with similar orientation to form elongated structures. This is evident in Figure 3(b), where the image component controlled by several of the $\mathbf{d}$ variables is shown to produce an elongated structure, and one that is significantly more complex than a Gabor (if $\mathbf{\Theta}\,\mathbf{c}$ is set to be a vector of all ones, the averaged group activations shown in Figure 3(b) are the image components contributed by a single $\mathbf{d}$ variable). Notice there is significant variety among these groups, and that they capture several properties of natural image patches, such as elongated structures across the entire patch, and different types of oriented and non-oriented textures.

The coupling structure encoded by $\mathbf{\Theta}$ can be visualized by displaying the columns of $\mathbf{\Phi}$ that correspond to the largest entries in each column of $\mathbf{\Sigma} = \mathbf{\Theta}\,\mathbf{\Theta}^{\mathrm{T}}$, which gives the covariance of the variable $\mathbf{f} = \mathbf{\Theta}\,\mathbf{c}$ – a useful way to interpret $\mathbf{\Theta}$. In Figure 4, each displayed column corresponds to a column in $\mathbf{\Sigma}$, with the upper squares holding the bases with the strongest corresponding $\mathbf{\Sigma}$ entries. In Figure 5, samples from several bilinear models are provided for visual comparison.

References

[1] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, 2010. 1

[2] R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995. 2, 3

[3] C. F. Cadieu and B. A. Olshausen. Learning transformational invariants from natural movies. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21. MIT Press, 2009. 1

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. 4

[5] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 4

[6] A. Courville, J. Bergstra, and Y. Bengio. A spike and slab restricted Boltzmann machine. In *AISTATS*, 2011. 1, 5

[7] W. Freeman and J. B. Tenenbaum. Learning bilinear models for two-factor problems in vision. In *CVPR*, 1997. 1

[8] P. Garrigues and B. Olshausen. Learning horizontal connections in a sparse coding model of natural images. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, 2007. 1

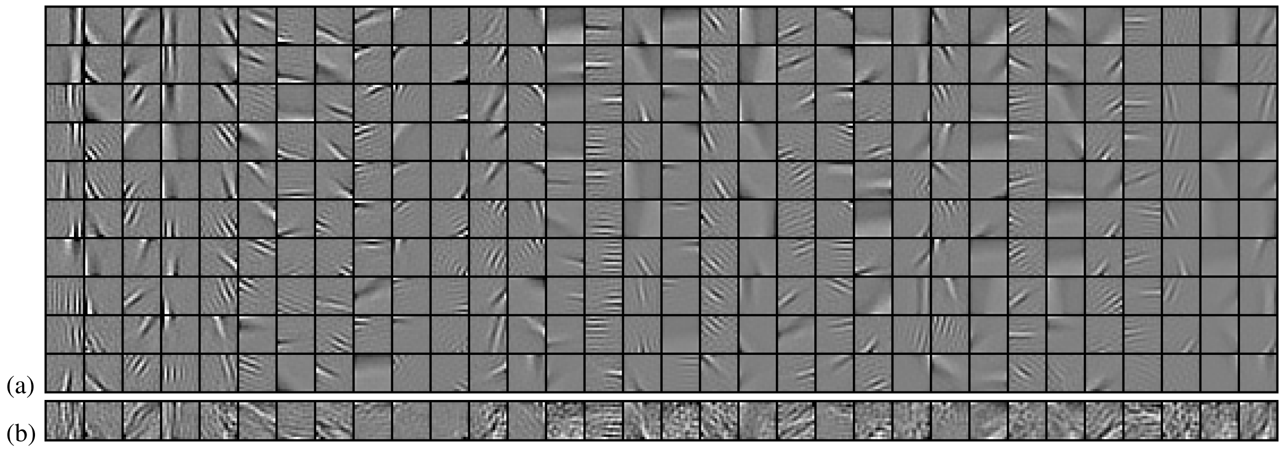
Figure 3. Dependencies between basis functions learned in $\boldsymbol{\Psi}$ from natural images. (a) The 10 most positively coupled columns in $\boldsymbol{\Phi}$ are shown for 32 randomly selected columns in $\boldsymbol{\Psi}$ (out of 256). Bases are weighted by their corresponding coupling strength in $\boldsymbol{\Psi}$. (b) The average value of all weighted bases.

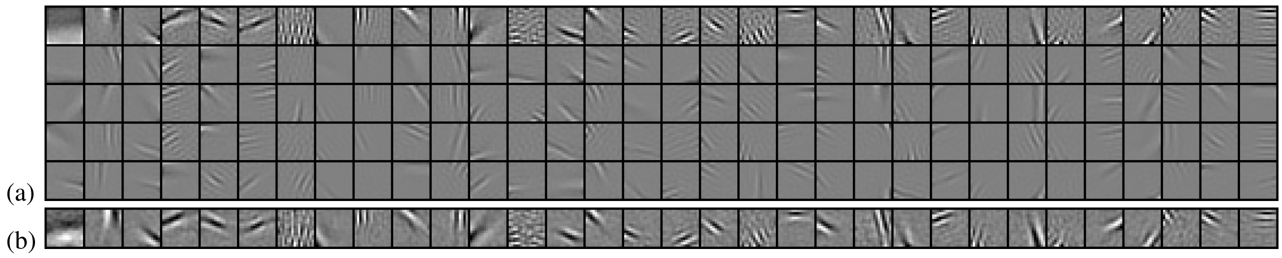
Figure 4. Dependencies between basis functions learned in $\boldsymbol{\Theta}$ from natural images. (a) The 5 most strongly coupled columns in $\boldsymbol{\Phi}$ are shown for 32 randomly selected columns in $\boldsymbol{\Sigma}$ (out of 256). Bases are weighted by their corresponding coupling strength in $\boldsymbol{\Sigma}$. (b) The average value of all weighted bases. Note that, unlike for $\boldsymbol{\Psi}$, the bases combine constructively, suggesting that $\boldsymbol{\Theta}$ is enforcing consistency and sign agreement.

[9] P. Garrigues and B. Olshausen. Group sparse coding with a laplacian scale mixture prior. In *Advances in Neural Information Processing Systems (NIPS)*, volume 23, 2010. 1

[10] D. B. Grimes and R. P. N. Rao. A bilinear model for sparse coding. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15. MIT Press, 2002. 1

[11] D. B. Grimes, A. P. Shon, and R. P. Rao. Probabilistic bilinear models for appearance-based vision. In *ICCV*, 2003. 1

[12] A. Horowitz. A generalized guided monte carlo algorithm. *Physics letters B*, Jan 1991. 2

[13] A. Hyvärinen and P. Hoyer. Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000. 1

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

[15] Y. Karklin and M. S. Lewicki. Is early vision optimized for extracting higher-order dependencies? In *Advances in Neural Information Processing Systems (NIPS)*, volume 18, pages 625–642. MIT Press, 2006. 1

[16] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierarchies for visual recognition. In *NIPS*, 2010. 1

[17] U. Köster and A. Hyvärinen. A two-layer ICA-like model estimated by score matching. In *ICANN*, 2007. 1

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, 2008. 1

[19] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008. 1

[20] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008. 4

[21] R. Memisevic and G. E. Hinton. Unsupervised learning of image transformations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 1

[22] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 22(6):1473–1492, June 2010. 1, 2

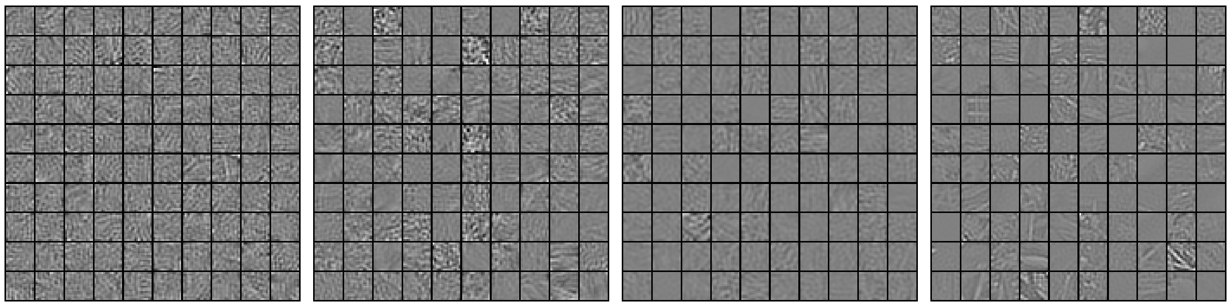
(a) $\mathbf{d} \sim \mathcal{E}(1), \mathbf{\Psi} = \mathbf{I}, \mathbf{\Theta} = \mathbf{I}$ (b) $\mathbf{d} \sim \mathcal{E}(\mathbf{I})$ (c) $\log \mathbf{d} \sim \mathcal{N}(0, \mathbf{\Omega}\,\mathbf{\Omega}^{\mathrm{T}})$ (d) Natural image patches

Figure 5. A comparison of samples drawn from the learned models, shown after projecting through the inverse whitening transform. (a) Bilinear model with $\mathbf{d} \sim \mathcal{E}(\mathbf{I})$, and $\mathbf{\Psi}$ and $\mathbf{\Theta}$ locked to $\mathbf{I}$. Samples were drawn from the priors for $\mathbf{c}$, $\mathbf{d}$ and $\mathbf{n}$, then used to generate from the model according to equation 2. (b) $\mathbf{\Psi}$ and $\mathbf{\Theta}$ were also learned. (c) Bilinear model with $\log \mathbf{d} \sim \mathcal{N}(0, \mathbf{\Omega}\,\mathbf{\Omega}^{\mathrm{T}})$. (d) Natural image patches from the training set.

[23] R. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, Jan 2001. 3

[24] R. Neal. *MCMC using Hamiltonian dynamics*, chapter 5. Chapman & Hall, 2010. 2, 3

[25] B. A. Olshausen, C. F. Cadieu, B. J. Culpepper, and D. K. Warland. Bilinear models of natural images. In *SPIE Proceedings: Human Vision and Electronic Imaging XII*, volume 6492, 2007. 1

[26] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, (381):607–609, 1996. 1

[27] S. Osindero, M. Welling, and G. E. Hinton. Topographic product models applied to natural scene statistics. *Neural Computation*, pages 381–414, 2006. 1

[28] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML 2007*, 2007. 1

[29] M. Ranzato and G. E. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann machines. *IEEE Conference on Computer Vision and Pattern Recognition*, Jan 2010. 4, 5

[30] M. Ranzato, F. J. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR 2007*, 2007. 1

[31] M. Ranzato, A. Krizhevsky, and G. E. Hinton. Factored 3-way restricted Boltzmann machines for modeling natural images. In *AISTATS*, volume 13, 2010. 1, 2

[32] J. Sohl-Dickstein, P. Battaglino, and M. R. DeWeese. Minimum probability flow learning. In *ICML*, 2011. 5

[33] J. Sohl-Dickstein and B. J. Culpepper. Hamiltonian annealed importance sampling for partition function estimation. *Redwood Center Technical Report*, 2011. 3

[34] T. Tieleman and G. Hinton. Using fast weights to improve persistent contrastive divergence. *International Conference on Machine Learning*, 2009. 5

[35] J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 265(1394):359–366, Jan 1998. 4

[36] M. J. Wainwright and E. P. Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In *Advances in Neural Information Processing Systems (NIPS)*, volume 12, 2000. 2

[37] M. Welling, G. Hinton, and S. Osindero. Learning sparse topographic representations with products of student-t distributions. *Advances in Neural Information Processing Systems*, 2003. 5

[38] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 1

[39] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. In *CVPR*, 2010. 1

[40] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems (NIPS)*, volume 22, 2009. 4