A Two-Layer Model of Natural Stimuli Estimated with Score Matching

Urs Köster and Aapo Hyvärinen
Department of Computer Science and HIIT
University of Helsinki, Finland

February 24, 2009

Abstract

We consider a hierarchical two-layer model of natural signals in which both layers are estimated simultaneously. Estimation is accomplished by Score Matching, a recently proposed estimation principle for energy-based models. By squaring the first layer outputs, and constraining the second layer weights to be non-negative, the model learns responses similar to complex cells in primary visual cortex from natural images. The second layer pools a small number of features with similar orientation and frequency, but differing in spatial phase. For speech data, we obtain analogous results. The model unifies previous extensions to ICA and is promising as a general method which can be applied in many different domains.

1 Introduction

A variety of methods like Independent Component Analysis (ICA) (Comon, 1994) and Sparse Coding (Olshausen & Field, 1997) have been applied to model the statistical structure of natural signals such as images and sounds. In computational neuroscience, the goal of modelling these signals with unsupervised learning methods is to understand more about sensory processing, which has been linked to the statistics of ecologically valid stimuli (Barlow, 1961; Hyvärinen et al., 2009).

ICA is limited in scope since it is linear and cannot remove arbitrary dependencies, so novel models use a nonlinear representation to better model the structure of the data. These models can roughly be divided into two classes: Direct extensions to ICA such as Independent Subspace Analysis (ICA) and topographic ICA (Hyvärinen et al., 2001; Hyvärinen & Hoyer, 2000) make use of a manually selected, fixed second layer that models dependencies between first layer features. Fixing the second layer, the probability density function (pdf) can still be normalized in closed form, making a simple gradient optimization on the log likelihood computationally feasible.

More recently a new class of models has begun to receive attention. At the expense of a a more complicated estimation, this second class of models tries to learn the optimal connectivity in both layers. A successful model of this kind is (Karklin & Lewicki, 2005), where linear features in the first layer are estimated first, using a variant of linear ICA. Subsequently the second layer is learned conditional on the first layer features. Simultaneous estimation of both layers is computationally expensive, so the estimation was simplified by sequential learning of the two layers. In a related model Osindero et al. (2006) reported that this does not affect the resulting first and second layer features, whereas a later study reports a large change in first layer features as they adapt to the second layer (Karklin & Lewicki, 2006). Thus an important question is, what effect does a simultaneous estimation of both layers have on these models of natural signals? In particular, as the authors in the previous study (Karklin & Lewicki, 2006) do not report on the effect that a simultaneous estimation has on the pooling patterns, this question deserves further investigation.

In this paper, we propose an alternative approach based on Score Matching. We argue that this method is particularly suited for simultaneous learning in a multilayer model, and compare the results to those obtained by (Karklin & Lewicki, 2005) and (Osindero et al., 2006). Recent work (Hyvärinen, 2005, 2007a) has established

Score Matching as a novel way of estimating parameters in energy-based models, which cannot be normalized in closed form. Compared to previous methods like Contrastive Divergence (Hinton, 2002) and Markov Chain Monte Carlo, its computational efficiency makes it particularly attractive for learning in a multi-layer framework. The alternative methods tend to be computationally expensive and careful tuning of parameters is often required.

We present a two-layer model of natural stimuli where both layers are estimated simultaneously, and analyze the resulting pooling patterns in the second layer. We follow the classical energy model of complex cells (Adelson & Bergen, 1985; Spitzer & Hochstein, 1985), where linear filter outputs are squared and then pooled. We therefore constrain the model by limiting the second layer to non-negative connectivity. Estimating the model with natural stimulus data, this leads to the emergence of sparse pooling in the second layer. For natural images, this corresponds to complex cell like properties in the outputs, where a few units with similar orientation and frequency, but different phase are pooled together. For speech data, similar results are obtained, with a pooling of basis functions that are similar in timing and frequency, but differ in phase. We compare the results from a simultaneous estimation of both layers with sequential estimation and show that with simultaneous estimation the outputs show more phase invariance and are therefore more like the complex cells of primary visual cortex than the outputs from the simpler model with sequential estimation of both layers. Thus we argue that simultaneous estimation of both layers should be preferred as the most principled model estimation.

In Section 2, we give a more detailed account on modelling natural image with a focus on ICA and its extensions in particular. In Section 3, the two-layer model itself is presented including mathematical details of our implementation and an introduction to the estimation method we used, Score Matching. In particular we quickly review how the Score Matching objective function is derived, highlighting the simplicity of the approach. We show experiments performed on synthetic data in Section 4 to demonstrate the performance of the model and estimation under controlled conditions. Then, in Section 5, we apply the method to natural image data. We provide a detailed account of the complex cell properties we obtain by analyzing the tuning statistics of the modelled cells. Similar experiments with speech data are shown in Section 6, where we obtain a sparse pooling in the second layer that is very similar to the results for natural images. In Section 7 we discuss how our work compares to similar two-layer models that have been developed recently. We highlight the principled estimation of our model and the effect of estimating the layers simultaneously rather than sequentially like other authors have done. Finally we conclude with Section 8, and sketch some possible future research directions. Preliminary results have been published in (Köster & Hyvärinen, 2007).

2 Modelling of Natural Images

The statistics of natural image patches have recently received a great deal of attention in computational neuroscience. Since mammalian visual receptive fields (RF's) were first described by Hubel and Wiesel in the 1960's (see e.g. (Hubel & Wiesel, 1959, 1962)), efforts have been made to understand why the RF's have the properties that were observed. One very fruitful route to this end was based on the idea that neural processing should be matched to statistics of ecologically valid stimuli, e.g. natural images. The amount of information collected by the retina is so large that is was conjectured that techniques from signal processing such as compression and denoising - which are based on statistical properties - would be employed by the brain. This lead to the development of statistical models like sparse coding (Olshausen & Field, 1996) and ICA (Jutten & Herault, 1991; Comon, 1994), which produced receptive fields with a strong resemblance to those of simple cells.

The model we use in this paper here takes its inspiration from the classical ICA model, so we will quickly review the principles of ICA and how it can be applied to natural images. ICA is built around the assumption that a vector of independent components $\mathbf{s}$ is mixed to generate the observed data vector $\mathbf{x}$. This is usually written as

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

In the simplest case that is usually considered the dimensionality of the component vector and the observation vector is the same, so $\mathbf{A}$ is a square mixing matrix which can be inverted. Thus the components can be recovered from the data with the inverse transform

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{2}$$

There is a range of methods for the estimation of this model, one of the most common being a likelihood-based approach. There, the distribution of the components is modelled by probability density functions (pdf's) p_i,

$$p(\mathbf{s}) = \prod_i p(s_i) \tag{3}$$

where the independence of the s_i is used to factorize the pdf. This allows us to write down the pdf of the data as

$$p(\mathbf{x}) = |\det \mathbf{W}| \prod_i p_i(\mathbf{w}_i^T \mathbf{x}) \tag{4}$$

where $\mathbf{w}_i^T$ is one of the rows of $\mathbf{W}$, and the determinant serves to normalize the distribution. Thus we obtain the sample version of the log-likelihood of the parameters as

$$\log L(\mathbf{W}) = \sum_t \sum_i \log p_i(\mathbf{w}_i^T \mathbf{x}(t)) + T \log(|\det \mathbf{W}|) \tag{5}$$

where $\mathbf{x}(t)$ runs over T samples from the data. This can easily be maximized by gradient ascent.

However, most of the cells in primary visual cortex are not well described with a linear response, in particular complex cells which are insensitive to spatial phase cannot be modeled with a linear transform. Therefore efforts were made to develop ICA-type models with a fixed second layer: Methods such as ISA (Independent Subspace Analysis) (Hyvärinen & Hoyer, 2000) and Topographic ICA (Hyvärinen et al., 2001) show that phase-invariant responses can be obtained if a pooling of individual linear filters is imposed on the model. ISA is an extension of ICA where the individual components s_i are projected onto a number of subspaces. Squared norms of projections onto the subspaces are computed as

$$u_j = \sum_{i \in S_j} s_i^2 \tag{6}$$

where the index i runs over all the components that belong to the j^{th} subspace. The model then takes the form

$$p(\mathbf{x}) = |\det \mathbf{W}| \exp\left(- \sum_j f\left(\sum_{i \in S_j} (\mathbf{w}_i^T \mathbf{x})^2 \right) \right) \tag{7}$$

where the squaring operation implies a projection on subspaces, and the scalar nonlinearity $f(.)$ defines the overall shape of the pdf. The pooling can also be viewed as a second linear transformation, or layer of neurons, where a number of first-layer units converge into one higher order unit. Since independence is required only for the higher order units u_j, the linear features that project to one subspace may have dependencies. Applied to natural images, this has the effect of grouping features with similar frequency and location, but different spatial phase. Thus it can be argued that complex cells are tuned to minimize energy dependencies, or correlations in the variance of higher-order units (Schwartz & Simoncelli, 2001), which the above model makes explicit by the squaring operation.

3 The Model and its Estimation

3.1 A two-layer model

While the models like ISA and TICA described above give important insight into the interpretation of simple and complex cells as feature detectors tuned to the statistics of natural stimuli, they are somewhat limited as explanations *why* the specific pooling is taking place, since only a linear transformation is learned from the data and the additional connectivity is pre-specified. This rules out certain types of connectivity that might provide a better model of the data in favor of architectures that have been observed physiologically or have been hypothesized from theoretical principles. A less strongly constrained two-layer model allows to evaluate

whether the kind of connectivity used in earlier models is actually valid from the point of view of statistical optimality. A conceptually very simple extension to the models described in the previous section would be to retain the basic structure, but learning the second layer from the data rather than pre-supposing it. To this end, let us start by formally defining the two-layer model. We define a pdf that can be viewed as describing a rather generic two-layer network

$$\log p(\mathbf{x}) = \sum_h f\left[\mathbf{v}_h g\left(\mathbf{W}\mathbf{x}\right)\right] - \log Z \tag{8}$$

where the total log-probability is given by the sum of individual probabilities, which correspond to the outputs of individual second layer units. The constant Z is the normalization constant, i.e. a function of the model parameters $\mathbf{W}$ and $\mathbf{V}$ which makes the integral of the pdf equal to one. Here, the $\mathbf{v}_h$ are row vectors of the second layer weight matrix $\mathbf{V}$, while the first layer $\mathbf{W}$ has been retained from the ICA model. Because of the generality of the estimation method the two matrices do not need to be square, so in general $\mathbf{W}$ will be of size $n \times m$ and $\mathbf{V}$ of size $m \times o$. We have two scalar nonlinearities $g(.)$ and $f(.)$, the first of which allows us to form a nonlinear representation of the data, whereas the second shapes the overall pdf. Such a model cannot be normalized easily since the normalization constant Z cannot be computed analytically. Therefore Score Matching (SM) is the estimation principle of choice, since it provides a straight-forward path to learning in this kind of energy-based model.

For the results presented in this work, we have specified g to be a squaring operation. This is the obvious choice for the energy model of complex-cells, which obtains outputs as a pooling of squared, or rectified simple cell responses (Pollen & Ronner, 1983; Spitzer & Hochstein, 1985). The second nonlinear function is chosen to be of the form

$$f(u) = -\sqrt{|u| + 1} \tag{9}$$

which ensures that the overall distribution of the model is supergaussian. Using these nonlinearities, the model distribution becomes:

$$\log p(\mathbf{x}) = -\sum_h \sqrt{\mathbf{v}_h \left(\mathbf{W}\mathbf{x}\right)^2 + 1} - \log Z \tag{10}$$

This fixed combination of nonlinearities was used in all the simulations, unless otherwise mentioned. Also in accordance with the classic complex cell model, the second layer was constrained non-negative.

A very important point is normalization of the weight layers. We constrain the first layer to be orthogonal and whiten the data. This is typical in ICA models since an orthogonal matrix is sufficient to perform ICA on whitened data. It is less clear what kind of normalization should be applied to the second layer, but the following argument should make clear that some form of normalization is required: When optimizing for sparseness, we have to distinguish between *sparseness of representation* and *sparseness of a feature* (also referred to as population sparseness and lifetime sparseness, see (Willmore & Tolhurst, 2001)). For the representation to be sparse, it is sufficient to have some features that are always active, and others that are completely silent. This is of course a rather meaningless concept, and for a useful representation we require each feature to be sparse, and in particular, become active at some point. We enforce the correct type of sparseness by normalizing the mean activity of each feature to be equal to some constant value. We normalize the $\mathbf{v}_h$ to unit L_1-norm, which corresponds to contraining the second layer units to have unit output energy. After each gradient step, the rows of the second layer were projected on the contour of unit L_1-norm. To make sure that the sparseness in the results is not an artifact of the L_1-norm contraint, we compare the results to experiments where the weights are normalized with respect to the L_2-norm.

3.2 Score Matching

Score matching (Hyvärinen, 2005, 2007b,a) (SM) is a statistical method that allows the estimation of statistical models which can only be determined up to a multiplicative normalization constant. These so-called "energy-based" models come up frequently in machine learning and computational neuroscience. Previously problems of this kind had to be solved with Markov Chain Monte Carlo methods, which are typically very time-consuming.

Consider a random vector $\mathbf{x} \in \mathbb{R}^n$ that follows a pdf $p_{\mathbf{x}}(\boldsymbol{\xi})$. We define a parametrized model density $p(\boldsymbol{\xi}|\boldsymbol{\Theta})$ where $\boldsymbol{\Theta}$ is a parameter vector that we would like to estimate.

For the kind of problem we consider here the normalization constant Z of the pdf cannot be computed, and we use q do denote the unnormalized distribution. In the form of a log-probability we have the model:

$$\log p(\boldsymbol{\xi}|\boldsymbol{\Theta}) = \log q(\boldsymbol{\xi}|\boldsymbol{\Theta}) - \log Z(\boldsymbol{\Theta}) \tag{11}$$

The model score function, which we define as the gradient of the log-probability with respect to the data, is obviously identical for q and p, and given by:

$$\Psi(\boldsymbol{\xi};\boldsymbol{\Theta}) = \nabla_{\boldsymbol{\xi}} \log p(\boldsymbol{\xi};\boldsymbol{\Theta}) \tag{12}$$

Likewise the score function of the observed data is denoted by

$$\Psi_{\mathbf{x}}(.) = \nabla_{\boldsymbol{\xi}} \log p_{\mathbf{x}}(.) \tag{13}$$

Working with the score function thus has the advantage that it is independent of the normalization constant Z. The model can now be estimated by minimizing the squared distance between the *model score function* $\Psi(\xi;\boldsymbol{\Theta})$ and the *data score function* $\Psi_{\mathbf{x}}(.)$ This objective function is defined by

$$J(\boldsymbol{\Theta}) = \frac{1}{2}\int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\|\Psi(\boldsymbol{\xi};\boldsymbol{\Theta}) - \Psi_{\mathbf{x}}(\boldsymbol{\xi})\|^2 d\boldsymbol{\xi} \tag{14}$$

This may not appear to be very useful at first sight, because estimating the data score function is a nonparametric problem, and would require as much effort as an estimate of the normalization constant.

We will now sketch a proof how a much simpler form of the objective function can be obtained. The full proof can be found in (Hyvärinen, 2005). We start by expanding the squared term to

$$J(\boldsymbol{\Theta}) = \frac{1}{2}\int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\|\Psi(\boldsymbol{\xi};\boldsymbol{\Theta})\|^2 d\boldsymbol{\xi} + \frac{1}{2}\int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\|\Psi_{\mathbf{x}}(\boldsymbol{\xi})\|^2 d\boldsymbol{\xi} - \int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\Psi(\boldsymbol{\xi};\boldsymbol{\Theta})^T\Psi_{\mathbf{x}}(\boldsymbol{\xi})d\boldsymbol{\xi} \tag{15}$$

Here we note that the first term does not depend on the data score function, so rewriting it with the inner product expanded as a sum we get

$$\frac{1}{2}\int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\|\Psi(\boldsymbol{\xi};\boldsymbol{\Theta})\|^2 d\boldsymbol{\xi} = \int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}} \sum_{i=1}^{n} \frac{1}{2}\psi_i^2(\boldsymbol{\xi};\boldsymbol{\Theta})d\boldsymbol{\xi} \tag{16}$$

The second term is constant wrt. $\boldsymbol{\Theta}$, so we simply set

$$\frac{1}{2}\int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\|\Psi_{\mathbf{x}}(\boldsymbol{\xi})\|^2 d\boldsymbol{\xi} = C \tag{17}$$

Thus we focus on the third term, where we start by writing out the inner product

$$\sum_i \int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\psi_i(\boldsymbol{\xi};\boldsymbol{\Theta})\psi_{\mathbf{x},i}(\boldsymbol{\xi})d\boldsymbol{\xi} \tag{18}$$

and consider only a single term. We now rewrite the score function $\psi_{\mathbf{x},i}(\boldsymbol{\xi}) = \frac{\partial \log p_x(\boldsymbol{\xi})}{\partial \xi_i}$, so making use of the chain rule, the term becomes

$$\sum_i \int_{\boldsymbol{\xi}\in\mathbb{R}^n} p_{\mathbf{x}}(\boldsymbol{\xi})\psi_i(\boldsymbol{\xi};\boldsymbol{\Theta})\left[\frac{\partial \log p_x(\boldsymbol{\xi})}{\partial \xi_i}\right] d\boldsymbol{\xi} = \sum_i \int_{\boldsymbol{\xi}\in\mathbb{R}^n} \frac{p_{\mathbf{x}}(\boldsymbol{\xi})}{p_{\mathbf{x}}(\boldsymbol{\xi})}\frac{\partial p_x(\boldsymbol{\xi})}{\partial \xi_i}\psi_i(\boldsymbol{\xi};\boldsymbol{\Theta})d\boldsymbol{\xi} \tag{19}$$

We then use multivariate partial integration (Hyvärinen, 2005) to obtain the i-th term as

$$-\int_{\boldsymbol{\xi}\in\mathbb{R}^n} \frac{\partial p_x(\boldsymbol{\xi})}{\partial \xi_i}\psi_i(\boldsymbol{\xi};\boldsymbol{\Theta})d\boldsymbol{\xi} = \int_{\boldsymbol{\xi}\in\mathbb{R}^n} \frac{\partial \psi_i(x;\boldsymbol{\Theta})}{\partial \xi_i} \tag{20}$$

Working with a sample of data, we replace the integrals w.r.t. p_x with sample expectations. Putting this together we obtain the expression

$$\tilde{J}(\mathbf{\Theta}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\left[\frac{\partial\psi_i(x(t);\mathbf{\Theta})}{\partial\xi_i} + \frac{1}{2}\psi_i^2(\mathbf{x}(t);\mathbf{\Theta})\right] + C \tag{21}$$

Score matching has been shown to be consistent in (Hyvärinen, 2005), so if the data follows the model, the method is guaranteed to converge.

3.3 Estimating the model

We can now apply the score matching framework to the model pdf that was given in Equation (10). The score function of the two-layer network is given by

$$\Psi_{\mathbf{x}}(\boldsymbol{\xi}) = \nabla_{\boldsymbol{\xi}}\sum_{h} f\left[\mathbf{v}_h g\left(\mathbf{W}\boldsymbol{\xi}\right)\right] \tag{22}$$

so we can write down the *score match*, i.e. the squared distance between model and data score function as

$$\begin{aligned} J(\mathbf{V},\mathbf{W}) = & \sum_{t=1}^{T}\sum_{k=1}^{n}\sum_{h=1}^{o}\sum_{\ell=1}^{m}\left[(w_\ell^k)^2 v_h^\ell g_\ell''(\mathbf{w}_i^T\mathbf{x}(t))f'(\sum_i v_h^i g_i(\mathbf{w}_i^T\mathbf{x}(t)))\right] \\ + & \sum_{t=1}^{T}\sum_{k=1}^{n}\sum_{h=1}^{o} f_h''(\mathbf{v}_h^T g(\mathbf{W}\mathbf{x}(t)))\left[\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g_\ell'(\mathbf{w}_i^T\mathbf{x}(t))\right]^2 \\ + & \sum_{t=1}^{T}\sum_{k=1}^{n}\frac{1}{2}\left[\sum_{h=1}^{o}\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g_\ell'(\mathbf{w}_i^T\mathbf{x}(t))f_h'(\mathbf{v}_h^T g(\mathbf{W}\mathbf{x}(t)))\right]^2 \end{aligned} \tag{23}$$

Estimating this model is straightforward by gradient descent, which requires the gradients of the above expression with respect to the elements of the weight matrices $\mathbf{W}$ and $\mathbf{V}$. These gradients are algebraically rather tedious, they are given in the Appendix. For the estimation the algorithm was initialized with Gaussian white noise of unit variance for $\mathbf{W}$ and with an identity matrix for $\mathbf{V}$. The convergence of the algorithm was checked by visual inspection of the score matching objective function and the weight matrices.

4 Experiments on Simulated Data

As a test of the identifiability of the model we applied it to simulated data with a known higher-order structure. We generated data following the ISA model, which is a special case of the proposed two-layer model and easy to generate data with. Analyzing this special case also has the advantage of being simple enough for easy analysis. The generated data contains higher order dependencies in the form of common variances for groups of source variables, which are beyond the reach of a linear model such as ICA. Data following the ISA model was generated as follows: To obtain T observations of an n-dimensional vector which contains k independent subspaces, we first create a matrix $\mathbf{M}$ of $n \times T$ observations from a Gaussian distribution of unit variance, and a matrix $\mathbf{B}$ of $k \times T$ variance parameters from a uniform distribution. We introduce dependencies between groups of the Gaussians by multiplying them with a common variance from the uniform distribution: $\mathbf{U}(i,t) = \mathbf{M}(i,t)\mathbf{B}(j,t), \forall i \in S_j$. This supergaussian data is then mixed with a mixing matrix $\mathbf{A}$ that is also generated randomly, so the data matrix is $\mathbf{X} = \mathbf{A}\mathbf{U}$. Before the estimation the data is whitened. For the experiments shown below we choose $T = 5000$, $n = 21$ and $k = 7$, so each subspace had three elements. The performance of the algorithm was judged by visual inspection of the product of the filter matrix $\mathbf{W}$ with the mixing matrix to obtain $\mathbf{Z} = \mathbf{W} \times \mathbf{A}$, which should be identity up to permutations and signs in the case of perfect recovery of the components.

The experiments with artificial data mainly served the purpose to confirm the consistency of the estimation method, but also to determine the correct initialization and normalization procedures for the experiments on

natural stimuli. We compare L_1- and L_2-normalization of the second layer matrix $\mathbf{V}$, and we compare randomly initializing $\mathbf{V}$ and initializing it with an identity matrix. Finally we compare initializing the first layer $\mathbf{W}$ with Gaussian white noise and initializing it by learning ICA filters with a fixed second layer.

Figure 1 shows the results for each of these various trials. In ICA, one can simply multiply the mixing matrix $\mathbf{A}$ with the estimated filter matrix $\mathbf{W}$ which should be a permuted diagonal matrix if the components are identified correctly. Thus visual inspection of $\mathbf{Z} = \mathbf{W} \times \mathbf{A}$ can be used to to determine convergence. The ISA or two layer model is identifiable only up to subspaces, where the second layer determines the subspace ownership of each element of $\mathbf{Z}$. By multiplying the second layer matrix $\mathbf{V}$ with $\mathbf{Z}$, a block-diagonal matrix with permuted rows should be obtained if the algorithm converges correctly.

In each of the four experiments *(a)-(d)* we performed, the top row shows the second layer $\mathbf{V}$ on the left and the product $\mathbf{V} \times \mathbf{Z}$, on the right. In the bottom row, on the left we show $\tilde{\mathbf{V}}$ where the rows have been sorted in such a way that identical rows are next to another. This is purely for visualization purposes and does not affect the objective function. Again, on the right we plot the product $\tilde{\mathbf{V}} \times \mathbf{Z}$. If this results in a permuted block-diagonal matrix, the second layer has correctly identified the dependencies in the first layer.

Firstly, the comparison between *(a)* and *(b)* shows that convergence is possible both from $\mathbf{V}$ initialized with the identity matrix and from a random $\mathbf{V}$. However the number of iterations is about an order of magnitude greater starting from random. For this reason all experiments with image patches were performed with $\mathbf{V}$ initialized as identity. Secondly, in between *(a)* and *(c)* we compare the effect of L_1 and L_2-normalization. In this case, the L_2-normalized model has converged to a local minimum and has not identified all the components correctly. Finally, in *(a)* and *(d)* we analyze the effect of learning the layers separately, rather than simultaneously. The results indicate that the correct solution is still found if the first layer $\mathbf{W}$ is initialized with $\mathbf{V}$ clamped to identity. After $\mathbf{W}$ has converged, learning continued with both layers simultaneously. This approach, which amounts to initializing the first layer with an ICA basis, was retained for the experiments on natural images, because of an increase in the speed of convergence.

5 Experiments on Natural Images

After confirming the consistency of the estimation method with simulated data, we tested the model on natural images which have a particularly rich statistical structure with complex higher order dependencies.

5.1 Methods

All experiments were performed on images taken from P. O. Hoyer's ImageICA package[1], using $20,000$ image patches of size 16×16. The whole images were preprocessed by approximate whitening assuming a $\frac{1}{f^2}$ power spectrum and contrast gain control with a Gaussian neighborhood of 16 pixels diameter. Details of this preprocessing can be found in (Hyvärinen & Köster, 2007). The preprocessing can be given a physiological justification in terms of the processing in the retina (contrast gain control) and *lateral geniculate nucleus* (whitening), or it can be viewed more pragmatically as simplifying the statistical structure of the images slightly. We then randomly samples patches from the images, remove the DC component from the patches, and discard any blank image patches. Finally we used an eigenvalue-decomposition to whiten again and reduced the dimensionality using Principal Component Analysis. The dimensionality was reduced from 256 to 120. The dimensionality reduction corresponds to low-pass filtering and eliminates sampling artifacts from the image patches. In all experiments both weight matrices $\mathbf{W}$ and $\mathbf{V}$ were chosen to be square, of size 120×120.

The matrix $\mathbf{W}$ was initialized as white noise, and $\mathbf{V}$ as an identity matrix. Both were optimized using gradient descent with a constant stepsize. To increase the speed of convergence, we first estimated the first layer only, keeping the second layer fixed. After the conditional convergence of the first layer, we performed two different experiments. In the first type of experiment, the estimation was continued by updating both layers simultaneously after $\mathbf{W}$ had converged. In the second type, which served as a control, $\mathbf{W}$ was held fixed after initial convergence, and only $\mathbf{V}$ was continued to be estimated. During the estimation of the second layer, the

[1] www.cs.helsinki.fi/patrik.hoyer

outputs units (rows of $\mathbf{V}$) were normalized to unit L_1 or L_2 norm after every step. Convergence was determined by visual inspection and took about 300 hours on a Pentium IV workstation.

To analyze the tuning properties of the filters in the first layer, we fit Gabor functions to the rows of the filter matrix $\mathbf{W}$. For each of the first layer responses, we used a least squares fit (adapted from (Hyvärinen et al., 2001)) to determine location, orientation, size, phase and frequency. To compute tuning curves of the second layer complex cell outputs, we followed our model by taking squares of the first layer filter responses and summing them, weighted by a row of $\mathbf{V}$. The second layer nonlinearity is required in the estimation to define a supergaussian pdf, but it is not considered part of our complex cell model, so we analyzed the second layer outputs without any further nonlinearity. To obtain tuning curves the "optimal stimulus" for a particular complex cell was chosen to be a Gabor function constructed from a weighted average of the constituent first layer filter parameters. One of the parameters (location, orientation, phase, frequency) was then varied while the others were held at the optimum value. The response of the complex cell was computed for 100 different instances of the parameter.

5.2 Results

In Figure 2 we show the first and second layer weights learned from 16×16 image patches. We analyze the effect of the two types of normalization of the second layer, *(a-c)* show L_1-norm results and *(d-f)* L_2-norm results. In each case, the first layer *(a, d)* shows Gabor-like filters tuned in orientation, location, frequency and phase, very similar to those obtained by ICA models. In *(b, e)* we plot the most active features for a random selection of second order units. Like complex cells, the units can be seen to share similar frequency, orientation and location, but vary in spatial phase. In *(c, f)* the pooling connections that were learned for the individual features are shown. The connectivity can be seen to be sparse, with only a few first layer features contributing to each row of $\mathbf{V}$. Since $\mathbf{V}$ was initialized with an identity matrix, there is strong activity at the main diagonal.

To show the relation to the work by Karklin and Lewicki, we plot the higher order components in in Figure 3 in a similar way to Fig. 6 in (Karklin & Lewicki, 2005). Rather than plotting a point for each component, we show an ellipse, with the orientation of the major axis corresponds to the orientation of the first-layer basis function. This is an intuitive way of plotting since most of the bases are reasonably well described by location and orientation. The brightness of the ellipse represents the activation strength, light gray being close to zero and black corresponding to maximal connection strength. The most striking feature compared to the components estimated by Karklin and Lewicki is that there are no higher order features in which a large fraction of basis functions are active (this was the case in about half of the components estimated by Karklin & Lewicki).

In Figure 4 we show a quantitative comparison on the effect of simultaneous vs. sequential estimation of the weight layers for L_1-norm experiments. We analyze the tuning curves of the output units by presenting various Gabor stimuli to the estimated units. In *(a)* only the first layer was learned and the second fixed to the identity matrix. This results in simple cell behavior with strong phase selectivity. In *(b)* we present statistics from a sequential estimation of the two weight layers. There is an obvious decrease in selectivity to spatial phase, indicating complex cell properties. In *(c)* both layers have been estimated simultaneously. This leads to a striking decrease in phase selectivity, i.e. the second layer outputs become more complex cell-like. In particular the upper 10% quantile of the outputs becomes essentially completely phase-invariant, whereas in the sequential estimation, there is still a 40% modulation in this quantile. At the same time the selectivity for position, orientation and frequency are not affected considerably, with a slight broadening that seems miniscule in comparison to the change in phase selectivity. This highlights the importance of simultaneously estimating both layers of the model.

Finally in Figure 5 we further analyze the connectivity in the second layer $\mathbf{V}$ for the two different cases of L_1- and L_2-normalization. We show histograms of the values of the elements in $\mathbf{V}$ which give the connectivity between first- and second-layer units. In both cases we clearly see the emergence of sparse connectivity, with the majority of the connections having zero strength. Of the 14,400 connections more than 10,000 fall into the lowest bin in the two experiments.

5.3 Estimation without non-negativity constraints

In addition to the experiments with a non-negative second layer, we also estimated the model with two different nonlinearities that did not have a non-negativity constraint. This has the effect that negative energy correlations can be modelled in addition to positive ones. First, we analyzed the case of a symmetric nonlinearity $f(u) = \log\cosh u$, which leads to an output distribution that is sparse with both negative and positive outputs. Additionally we report on the results with an asymmetric nonlinearity, where the negative half of the nonlinearity corresponds to a Gaussian distribution, and the positive half follows a logistic distribution. In both cases an L_2-norm constraint as imposed on rows of $\mathbf{V}$, which was initialized randomly. Moreover, the first-layer nonlinearity was set as $g(u) = \log\cosh u$ instead of the simple squaring, in order to be able to better model a heavy-tailed distribution.

In the symmetric model (results not shown), we found sparse connectivity of the second layer, but with higher order features very different from the complex cell-like responses of the non-negative model: For some of the outputs, the first layer forms pairs of features which both contain two Gabors in their receptive field, one that is identical in both features, and one that is identical but of opposite sign, as in (Lindgren & Hyvärinen, 2006). Individual higher order outputs pool one such pair of features with one strong negative and one positive weight. Using the identity $a^2 - b^2 = (a-b)(a+b)$, this can be interpreted as taking the product of the sum and difference of the two linear filters, which turns out to be the product of two individual Gabor filters. Thus the model can be interpreted as taking products of linear filter outputs, with results very similar to those observed previously in quadratic ICA (Lindgren & Hyvärinen, 2006). In the model with sparse positive and Gaussian negative outputs (results not shown), we report higher order features with one or a small number of highly active positive inputs, and a larger number of small negative inputs. This could possibly be interpreted as a surround-inhibition, or a gain control phenomenon.

Thus we see that the results depend very much on the choice of the nonlinearity f, and very different results can be obtained by changing the nonlinearity. It seems very difficult to to make a principled choice of f, because the usual measures of sparseness do not easily generalize to two-layer models. In future research, f could possibly be estimated from the data. In the current work, we choose to analyze only the results from the non-negative model, because it seems to be most in line with the visual processing in mammalian visual cortex, i.e. complex cell responses. Furthermore we view our model as a direct extension to models with non-negative second layers such as ISA and topographic ICA. These have previously been motivated by the observation of strong positive energy correlations between linear filter outputs, so non-negativity seems to be a most reasonable constraint.

6 Experiments on Audio data

In order to demonstrate the general applicability of our model to a variety of data sets, we also tested it on speech data from the TIMIT database. The model was estimated exactly the same way as for image data, and the data was preprocessed as follows: We took 100 short utterances from the database, and resampled them to 8kHz. The data was high-pass filtered with a cutoff at 100Hz and then normalized to unit variance. We sampled 10,000 random sound windows of 16ms length, which corresponds to 128-dimensional data and removed the DC component. We also applied our standard preprocessing consisting of whitening and contrast gain control, as described above for image data. Simultaneously we reduced the dimensionality from 128 to 120 which amounts to low-pass filtering and serves to eliminate artifacts due to the windowing procedure. The rows of second layer were again constrained to unit L_1-norm.

The results we obtained from the speech data are remarkably similar to those from image data and are presented in Figure 6. In *(a)* we show the first layer features in the time domain, which are tuned to specific frequency bands as well as positions and extend in time. The second layer in *(b)* is shown in the same way as for image data, and shows that a sparse connectivity has been learned between groups of first layer features. This is analyzed further in *(c)*, where we we show which first layer features are pooled in the second layer. We obtain higher order features where similar frequencies with slightly different temporal position are pooled. Interestingly the pooling size is considerably smaller than for image data, some of the outputs have as few as three contributing first order features. This seems to indicate that the breaking of the independence assumption

is not as severe for audio data as it is for images, and that residual dependencies after the first layer are smaller.

7 Discussion

In the past, relatively little work has been done with full two-layer models, since one very important feature is lost in the transition from a carefully designed second layer to a free one: While e.g. ISA has a pdf that is straightforward to normalize in closed form, this is not the case for a more general model. Therefore it is not possible to express the likelihood in simple terms (the partition function cannot be solved analytically), and learning becomes very difficult, possibly requiring stochastic methods such as Markov Chain Monte Carlo. Only recently it has become computationally feasible to estimate two-layer models. This new generation of models has the advantage that the models learn, rather than presuppose, dependencies between outputs of linear filters. The model that we have presented is estimated using Score Matching, which overcomes some of these difficulties and allows simultaneous estimation of both network layers. While including some previous extensions of ICA such as ISA and topographic ICA as special cases, our new model is far more general that those previous models. For example ISA forces the filters to group into subspaces of a constant size and with an equal contribution, and does not allow a single filter to be active in more than one higher order unit.

Two models have recently been proposed that have a similar hierarchical structure but are estimated differently. Most closely related to our work is the *hierarchical Product of experts* by Osindero et al. (Osindero et al., 2006). Instead of using the traditional "independent component" point of view, the model is defined as a "product of experts" model following Student-t distributions. The estimation is performed using contrastive divergence (CD), which was recently shown (Hyvärinen, 2007a) to be related to Score Matching. The results obtained are similar to those reported here: Estimating the model on natural images also lead to complex cell-like behavior in the outputs. While it is not explicitly mentioned in (Osindero et al., 2006), it is possible that the authors also observed sparse connectivity in the second layer. From our point of view the most surprising difference to our model is that the authors observe it makes little difference whether the second layer is estimated "on top of" a predetermined first layer. The change in first layer units, which leads a significantly more complex cell-like behavior of output units when the estimation is simultaneous, shows that this in not the case in our model. Rather, the first layer features adapt to the requirements of the second layer, and while learning with a fixed second layer provides a good initialization for the first layer, it is clearly advantageous to learn both layers simultaneously to optimize the model.

Another study, (Karklin & Lewicki, 2005), with its extension (Karklin & Lewicki, 2006) is to our knowledge the only previous model where a simultaneous estimation was shown to lead to significantly different results from sequential estimation. Similar to our results, the pooling patterns in the second layer change what constitutes the optimal linear filters in the first layer, and a change in the Gabor-like basis functions is observed. However in the original work (Karklin & Lewicki, 2005) where the authors analyze the pooling patters, a fixed first layer is used for computational simplicity, and in the later extension (Karklin & Lewicki, 2006) the authors focus purely on the change in first layer linear filters and do not comment on changes in the pooling patterns. In any case it would be difficult to compare the pooling patterns and the effect of simultaneous estimation directly to the model proposed here. In contrast to our work the authors report broadly tuned features in the second layer, describing global properties of the data, and do not show simple pooling patters with e.g. common orientations and spatial frequencies.

Thus the emergence of sparse connectivity is a key difference between the different models considered here. Karklin and Lewicki clearly do not obtain this type of connectivity, as seen from Fig. 5 in (Karklin & Lewicki, 2005). It seems likely that sparseness of the connectivity can be greatly influenced by small details in the model specifications, which can be seen from the effect the normalization of the higher order weight matrix in our model. We observed that the L_1-norm penalty is an important factor in obtaining a uniform population of phase invariant higher order cells which pool only few first order units. With the L_2-norm we observed a fragmentation into two populations of output cells with different pooling patterns. In addition to one very sparse population, a second population pools over a larger number of inputs, and looses some of the location selectivity, but retains orientation and spatial frequency tuning, still amounting to complex cell-like properties.

8 Conclusion

We have presented a two layer model that that can be used to learn the statistical structure of various kinds of data. By using Score Matching for the estimation, unsupervised learning in this type of model is made faster and more straightforward than with alternatives such as Monte Carlo methods. Contrary to ICA, higher order dependencies in the data can be captured to give better models of real world data. In contrast to similar models, we report that the tuning properties of the higher order features are dependent on a simultaneous estimation of both network layers, which in our results leads to increase in complex cell properties of the outputs.

Acknowledgments

Urs Köster is supported by a scholarship from the *Alfried Krupp von Bohlen und Halbach-Stiftung*

Appendix

Derivatives of the objective function

We need to evaluate the gradients in the objective function (equation 23) w.r.t. the elements of $\mathbf{W}$ and $\mathbf{V}$. Since the expression can readily separated into a sum of three terms, which we call A, B and C, we treat these separately. We get six terms $\frac{\partial A}{\partial \mathbf{W}}$, $\frac{\partial B}{\partial \mathbf{W}}$, etc. Writing these out, we get for the first term

$$\begin{aligned}
\frac{\partial A}{\partial w_c^d} &= \sum_{k=1}^{n}\sum_{h=1}^{o}\sum_{\ell=1}^{m} v_h^\ell \frac{\partial}{\partial w_c^d}\left[(w_\ell^k)^2 g_\ell''(.) f'(\sum_i v_h^i g_i(.))\right] && (24)\\
&= \sum_{h}^{o}\Bigg\{ v_h^c 2 w_c^d g_c''(.) f_h'(.) && (25)\\
&+ \sum_{k}^{n} v_h^c (w_c^k)^2 g_c'''(.) x_d f_h'(.) && (26)\\
&+ \sum_{k}^{n}\sum_{\ell}^{m} v_h^\ell (w_\ell^k)^2 g_\ell''(.) f_h'''(.) v_h^c g_c'(.) x_d \Bigg\} && (27)
\end{aligned}$$

For the second term we get

$$\begin{aligned}
\frac{\partial B}{\partial w_c^d} &= \sum_{k=1}^{n}\sum_{h=1}^{o} \frac{\partial}{\partial w_c^d} f_h''(.)\left[\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g_\ell'(.)\right]^2 && (28)\\
&= \sum_{h}^{o}\Bigg\{ \sum_{k}^{n} f_h'''(.) v_h^c g_c'(.) x_d \left[\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g_\ell'(.)\right]^2 && (29)\\
&+ f_h''(.) 2 \left[\sum_{\ell=1}^{m} w_\ell^d v_h^\ell g_\ell'(.)\right]\left[v_h^c g'(.)\right] && (30)\\
&+ \sum_{k}^{n} f_h''(.) 2 \left[\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g_\ell'(.)\right]\left[w_c^k v_h^c g_c''(.) x_d\right] \Bigg\} && (31)
\end{aligned}$$

and for the third term, the derivative is

$$\frac{\partial C}{\partial w_c^d} = \sum_{k=1}^{n}\left[\sum_{h=1}^{o}\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g_\ell'(.) f_h'(.)\right] \sum_{h=1}^{o}\sum_{\ell=1}^{m}\left[\frac{\partial}{\partial w_c^d} w_\ell^k v_h^\ell g_\ell'(.) f_h'(.)\right] \quad (32)$$

For better readability we substitute $\left[\sum_{h=1}^{o}\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g'_\ell(.) f'_h(.)\right] = A_k$ in all subsequent equations.

$$\frac{\partial C}{\partial w_c^d} = A_d \left[\sum_h v_h^c g'_c(.) f'_h(.)\right] \tag{33}$$

$$+ \sum_{k=1}^{n} A_k \left[\sum_h v_h^c w_c^k g''(w_c^d) x_d f'_h(.)\right] \tag{34}$$

$$+ \sum_{k=1}^{n} A_k \left[\sum_{h,\ell} v_h^l w_\ell^k g'_\ell(.) f''_h(.) v_h^c g'_c(.) x_d\right] \tag{35}$$

Next we evaluate the derivatives for V, for the first term:

$$\frac{\partial A_k}{\partial v_a^b} = \sum_{k=1}^{n}\sum_{h=1}^{o}\sum_{\ell=1}^{m} \frac{\partial}{\partial v_a^b}\left[(w_\ell^k)^2 v_h^\ell g''_\ell(.) f'_h(.)\right] \tag{36}$$

$$= \sum_{k}^{n}\left\{\sum_{\ell}^{m}(w_\ell^k)^2 v_a^\ell g''_\ell(.) f''(v_a^b g_b) g_b + (w_b^k)^2 g''_b(.) f'_h(.)\right\} \tag{37}$$

the second term:

$$\frac{\partial B_k}{\partial v_a^b} = \sum_{k=1}^{n}\sum_{h=1}^{o} \frac{\partial}{\partial v_a^b} f''_h(.)\left[\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g'_\ell(.)\right]^2 \tag{38}$$

$$= \sum_{k}^{n}\left\{f'''_a(.) g_b(.)\left[\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g'_\ell(.)\right]^2\right. \tag{39}$$

$$\left. + f''_a(.) 2\left[\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g'_\ell(.)\right] w_b^k g'_b(.)\right\} \tag{40}$$

and similar for the third term:

$$\frac{\partial C}{\partial v_a^b} = \sum_{k=1}^{n}\left[\sum_{h=1}^{o}\sum_{\ell=1}^{m} w_\ell^k v_h^\ell g'_\ell(.) f'_h(.)\right] \sum_{h=1}^{o}\sum_{\ell=1}^{m}\left[\frac{\partial}{\partial v_a^b} w_\ell^k v_h^\ell g'_\ell(.) f'_h(.)\right] \tag{41}$$

$$= \sum_{k=1}^{n} A_k \sum_{h=1}^{o}\sum_{\ell=1}^{m}\left[\frac{\partial}{\partial v_a^b} w_\ell^k v_h^\ell g'_\ell(.) f'_h(.)\right] \tag{42}$$

$$= \sum_{k=1}^{n}\left\{A_k w_b^k g'_b(.) f'_a(.) + A\sum_{\ell} w_l^k g'_l(.) v_a^l f''_a(.) g_b(.)\right\} \tag{43}$$

$$\tag{44}$$

References

Adelson, E., & Bergen, J. (1985). Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A 2*, (pp. 284 – 299).

Barlow, H. (1961). *Possible principles underlying the transformation of sensory messages*. Cambridge, MA: MIT Press. W. Rosenblith (Ed.) Sensory Communication.

Comon, P. (1994). Independent component analysis – a new concept? *Signal Processing*, *36*, 287–314.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.*, *14*(8), 1771–1800.

Hubel, D. H., & Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *J Physiol.*, *148*, 574 – 591.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in cat's visual cortex. *J Physiol.*, *160*, 106 – 154.

Hyvärinen, A. (2005). Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, *6:695–709*.

Hyvärinen, A. (2007a). Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on Neural Networks*, *18*(5), 1529–1531.

Hyvärinen, A. (2007b). Some extensions of score matching. *Computational Statistics and Data Analysis*, *51*, 2499–2512.

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

Hyvärinen, A., Hoyer, P., & Inki, M. (2001). Topographic independent component analysis. *Neural Computation.*, *13*(7), 1527–1558.

Hyvärinen, A., & Hoyer, P. O. (2001). A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, *41*, 2413 – 2423.

Hyvärinen, A., Hurri, J., & Hoyer, P. O. (2009). *Natural Image Statistics*. Springer-Verlag. In press.

Hyvärinen, A., & Köster, U. (2007). Complex cell pooling and the statistics of natural images. *Network: Computation in Neural Systems*, *18*, 81–100.

Jutten, C., & Herault, J. (1991). Blind separation of sources part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, *24*, 1–10.

Karklin, Y., & Lewicki, M. S. (2005). A hierarchical bayesian model for learning non-linear statistical regularities in non-stationary natural signals. *Neural Computation*, *17*(2), 397–423.

Karklin, Y., & Lewicki, M. S. (2006). Is early vision optimized for extracting higher-order dependencies? *Advances in Neural Information Processing Systems*, *18*, 625–642.

Köster, U., & Hyvärinen, A. (2007). A two-layer ICA-like model estimated by score matching. In *Artificial Neural Networks - ICANN 2007, Lecture Notes in Computer Science*, (pp. 798–807). Springer Berlin / Heidelberg.

Lindgren, J. T., & Hyvärinen, A. (2006). Emergence of conjunctive visual features by quadratic independent component analysis. *Advances in Neural Information Processing Systems*.

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

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, *37*(23), 3311–3325.

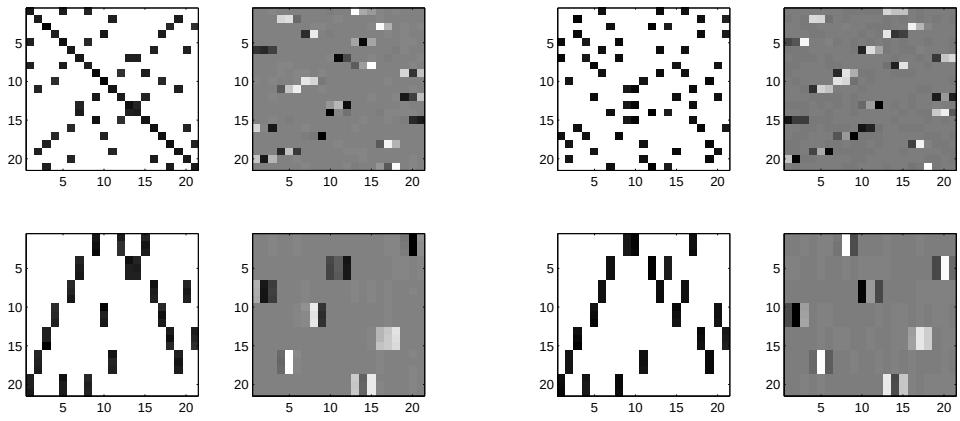
Osindero, S., Welling, M., & Hinton, G. E. (2006). Topographic product models applied to natural scene statistics. *Neural Computation*, *18*.

Pollen, D., & Ronner, S. (1983). Visual cortical neurons as localized spatial frequency filters. *IEEE Trans. on Systems, Man, and Cybernetics*, *13*, 907–916.

Schwartz, O., & Simoncelli, E. P. (2001). Natural signal statistics and sensory gain control. *Nature Neuroscience*, *4*(8), 819–825.

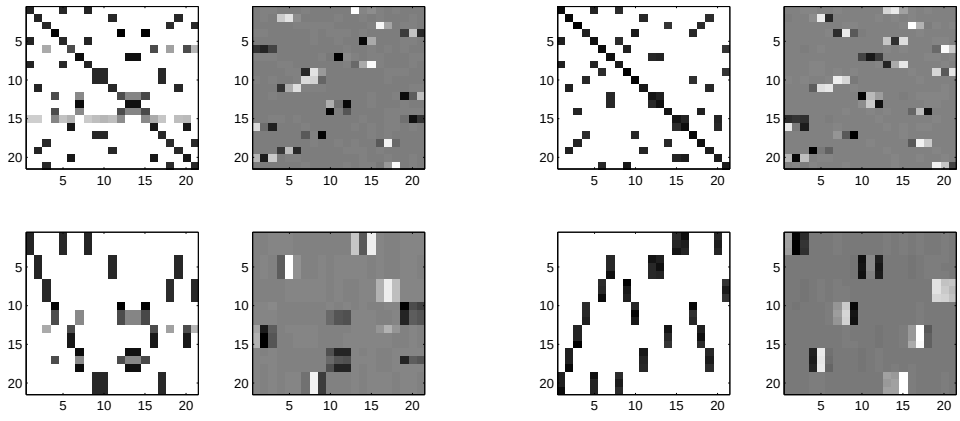
Spitzer, H., & Hochstein, S. (1985). A complex-cell receptive-field model. *J. Neurophysiol. 53*, (pp. 1266 – 1286).

Willmore, B., & Tolhurst, D. J. (2001). Characterizing the sparseness of neural codes. *Network: Computation in Neural Systems*, *12*, 255–270.



(a) Diagonal initialization, L_1-norm

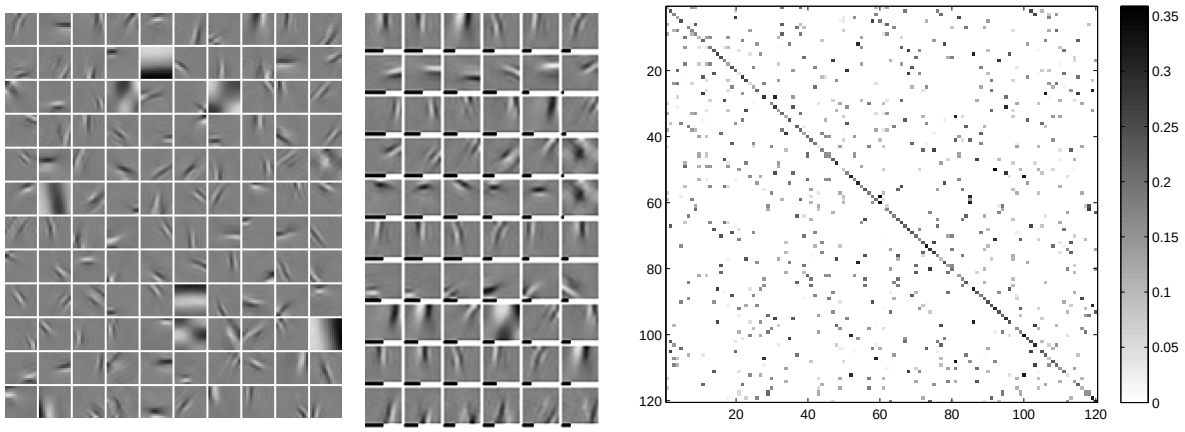
(b) Random initialization, L_1-norm



(c) Diagonal initialization, L_2-norm

(d) Separate learning of both layers, L_1-norm

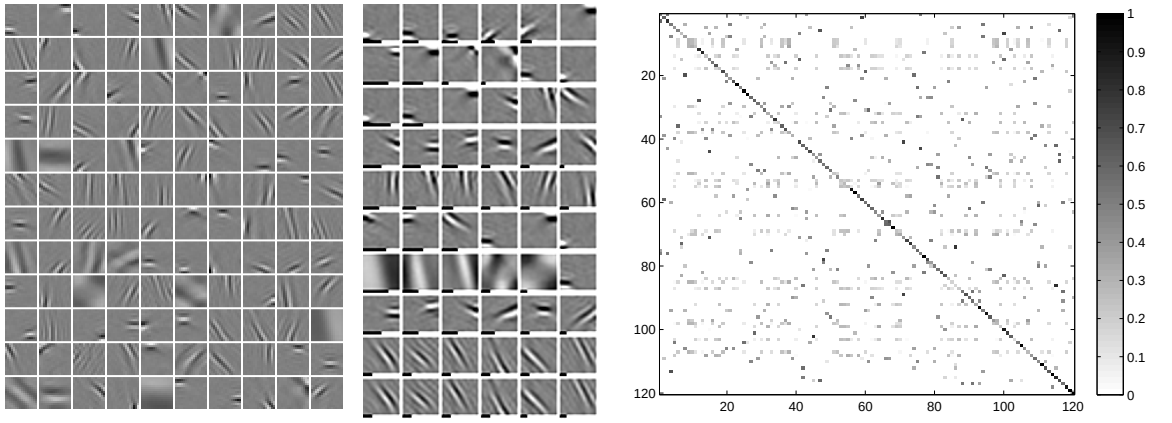
Figure 1: Simulations with synthesized data. For each of the four plots we show the second layer matrix $\mathbf{V}$ on the top left and the product $\mathbf{Z} = \mathbf{W} \times \mathbf{A}$ on the top right. The bottom row contains the same matrices as the top row, but with the vectors sorted for visualization purposes.
(a) Both layers estimated simultaneously with $\mathbf{W}$ initialized with Gaussian white noise and $\mathbf{V}$ with an identity matrix. The rows of $\mathbf{V}$ are constrained to unit L_1-norm.
(b) Like (a), but both weight layers initialized with white noise. Convergence takes nearly an order of magnitude longer, but the the global minimum is found nevertheless.
(c) Like (a), but with rows of $\mathbf{V}$ constrained to unit L_2-norm. Note that the second layer is not estimated correctly.
(d) The estimation can be simplified by estimating only $\mathbf{W}$ first, with $\mathbf{V}$ held constant. In the second step both layers are learned. While this does not seem useful with generated data, pre-learning the first layer like this is advantageous for natural data.



(a) First Layer, L_1-norm

(b) Some pooling patterns, L_1-norm

(c) Second Layer, L_1-norm

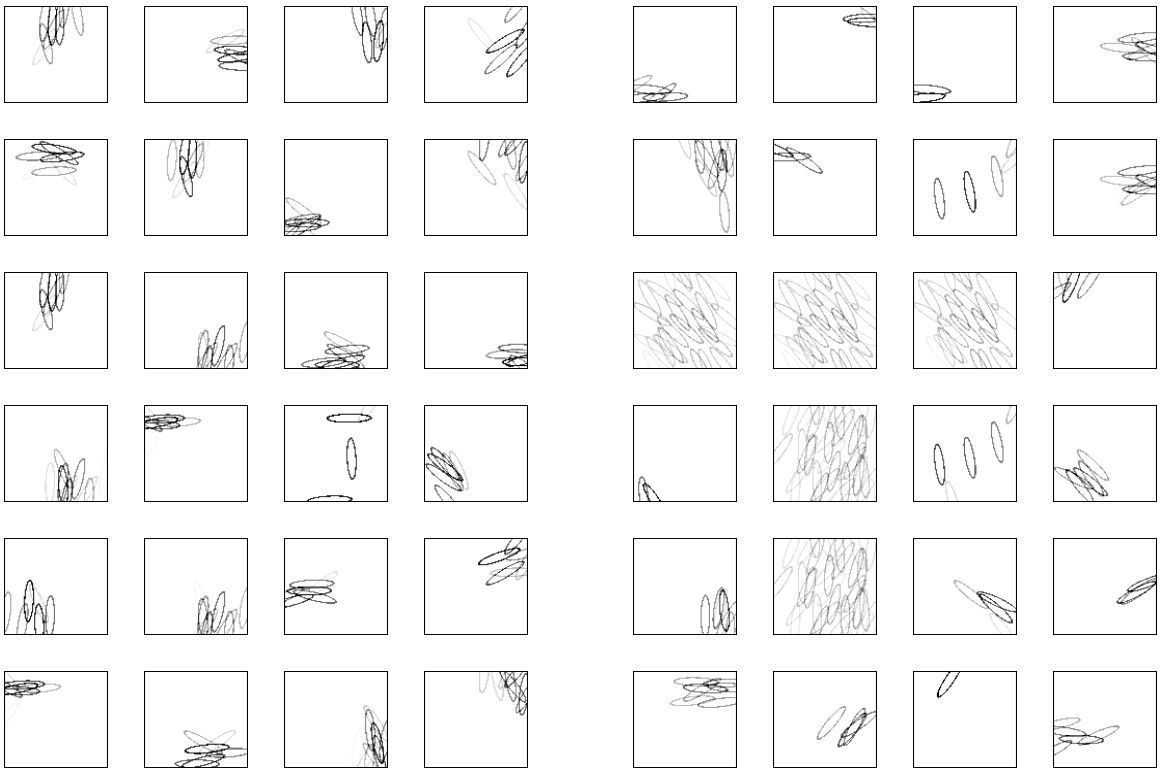


(d) First Layer, L_2-norm

(e) Some pooling patterns, L_2-norm

(f) Second Layer, L_2-norm

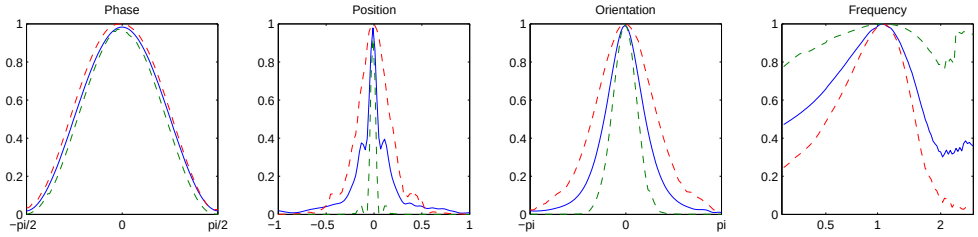
Figure 2: *(a)* First layer filters $\mathbf{W}$ show the classical simple cell type structure.
(b) A random selection of second-layer units where each row shows the most active first-layer contributers to the response with the black bars indicating "synaptic strength" (elements of $\mathbf{V}$), i.e. how strongly each filter contributes to the second-layer output.
(c) The second layer matrix $\mathbf{V}$. Connectivity is sparse, with connections between basis functions of similar tuning.
(d-f) Same as (a-c), but with L_2 rather than L_1-normalization.



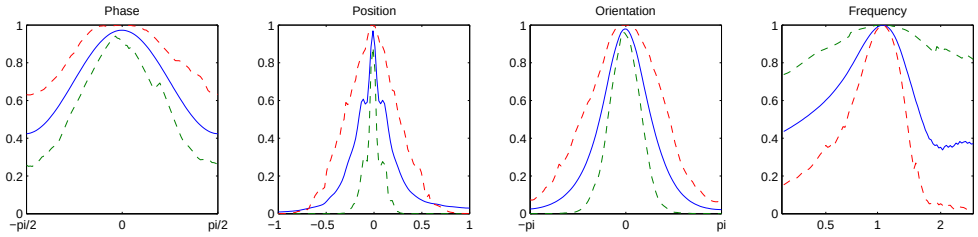
(a) Higher order features, L_1-norm

(b) Higher order features, L_2-norm

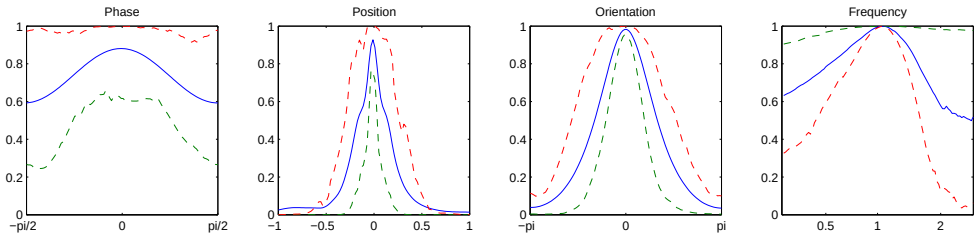
Figure 3: A random selection of 24 rows of $\mathbf{V}$, corresponding to individual higher order features, in an iconic representation. Each feature is represented by a number of ellipses corresponding to individual first layer basis functions with the same orientation and position as the ellipse. Spatial phase and frequency are not shown in this representation. Each unit can be seen to pool over a small number of basis functions that tend to be iso-oriented and co-localized. This is typical behavior for Complex Cell receptive fields.
While the L_1-norm penalty leads to a relatively uniform population of outputs which pool over approximately 5 linear filters, the features with an L_2-norm constraint show a distinct splitting into two sub-populations: Some features pool over a larger number of inputs and loose much of the location selectivity, while the rest of the features pool over fewer features than with the L_1-norm. In further analysis we therefore focused on the more uniform outputs obtained under the L_1-norm constraint.



(a) Estimating $\mathbf{W}$ only

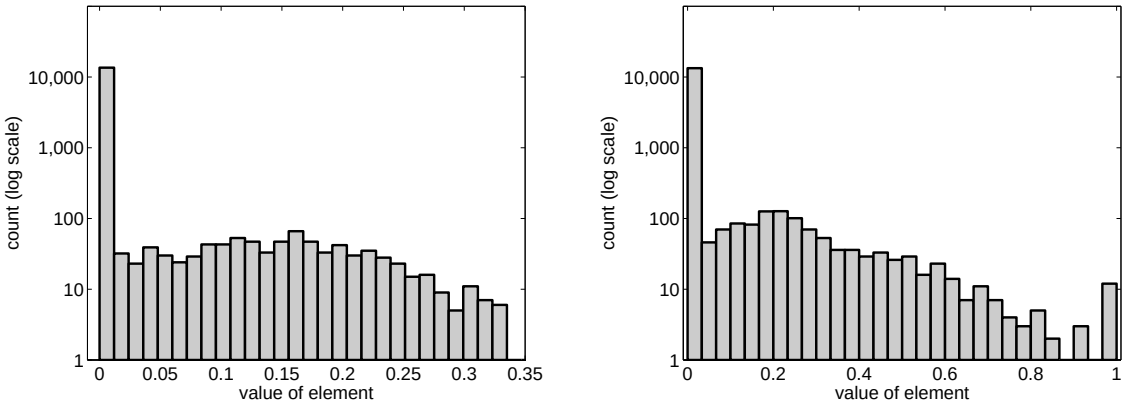


(b) Sequential estimation of $\mathbf{W}$ and $\mathbf{V}$



(c) Simultaneous estimation of $\mathbf{W}$ and $\mathbf{V}$

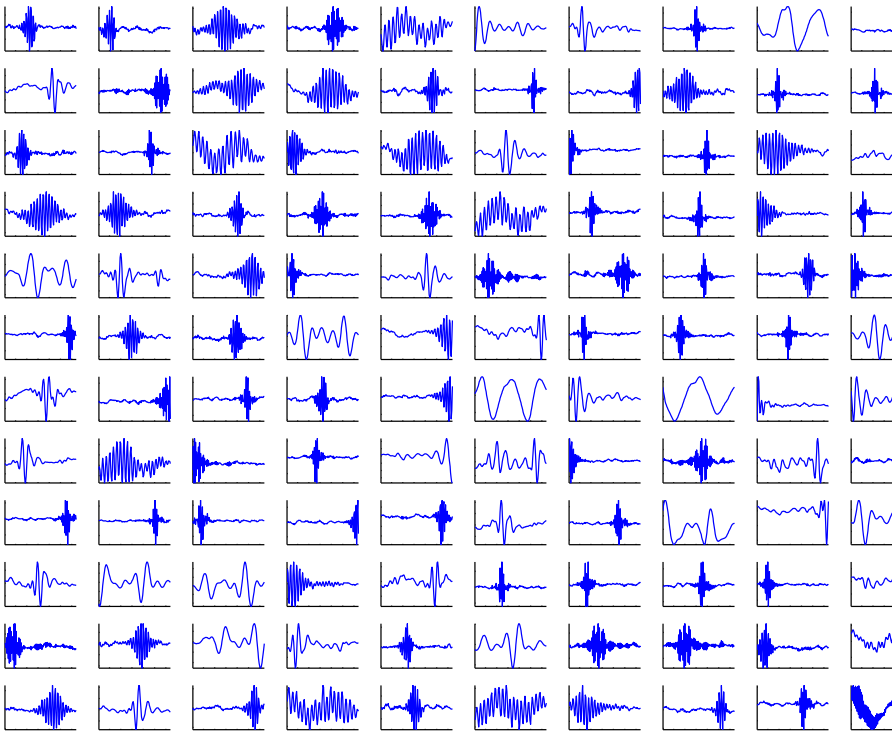
Figure 4: Analysis of complex cells properties of the second layer outputs, following (Hyvärinen & Hoyer, 2001). One parameter of the fitted Gabor was changed at a time, and the normalized response was plotted as a function of the tuning parameter. The solid line shows the mean response of 120 tested cells,the dashed lines give 10% and 90% quantiles.
(a) Only the first layer $\mathbf{W}$ was estimated and $\mathbf{V}$ was fixed to identity.
(b) After $\mathbf{W}$ had converged it was held constant and $\mathbf{V}$ was estimated using this constant first layer.
(c) $\mathbf{W}$ was initialized as above, but then both layers were estimated simultaneously. This shows significantly less phase-dependence in the tuning curves, indicating that $\mathbf{W}$ has adapted to the pooling imposed by $\mathbf{V}$.



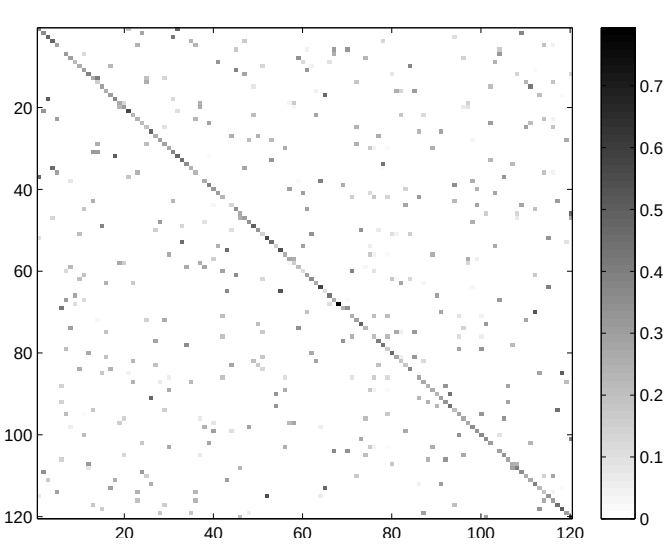
(a) Histogram of $\mathbf{V}$ estimated with L_1-norm constraint

(b) Histogram of $\mathbf{V}$ estimated with L_2-norm constraint

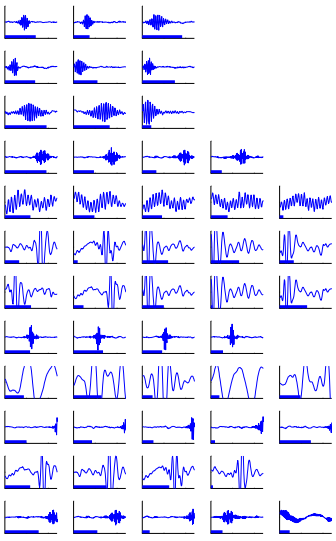
Figure 5: *(a)* Histogram of the values in the second layer matrix $\mathbf{V}$ estimated with L_1-norm constraint. The connectivity is clearly very sparse, only 6.0% of the elements are non-zero.
(b) Learning with an L_2-norm constraint on the second layer matrix $\mathbf{V}$ produces slightly different connectivity. The sparseness is still very high with 7.6% non-zero values, but these are distributed less uniformly than in the L_1-norm case. There is an increase in elements with a value close to one or zero, indicating higher order units with more heterogeneous pooling patterns.



(a) First Layer



(b) Second Layer



(c) Some pooling patterns

Figure 6: Experiments for speech data from the TIMIT database.
(a) The first layer gives outputs localized in both frequency and time.
(b) The second layer gives connections between features with dependencies of squares.
(c) A random selection of output units. Each row shows the active first layer filters in one row of $\mathbf{V}$.