A NEW VIEW OF ICA

G.E. Hinton, M. Welling, Y.W. Teh

Department of Computer Science
University of Toronto
10 Kings College Road, Toronto
Canada M5S 3G4

S. K. Osindero *

Gatsby Computational Neuroscience Unit
University College London
17 Queen Square, London
United Kingdom WC1N 3AR

ABSTRACT

We present a new way of interpreting ICA as a probability density model and a new way of fitting this model to data. The advantage of our approach is that it suggests simple, novel extensions to overcomplete, undercomplete and multilayer non-linear versions of ICA.

1. ICA AS A CAUSAL GENERATIVE MODEL

Factor analysis is based on a causal generative model in which an observation vector is generated in three stages. First, the activities of the factors (also known as latent or hidden variables) are chosen independently from one dimensional Gaussian priors. Next, these hidden activities are multiplied by a matrix of weights (the "factor loading" matrix) to produce a noise-free observation vector. Finally, independent Gaussian "sensor noise" is added to each component of the noise-free observation vector. Given an observation vector and a factor loading matrix, it is tractable to compute the posterior distribution of the hidden activities because this distribution is a Gaussian, though it generally has off-diagonal terms in the covariance matrix so it is not as simple as the prior distribution over hidden activities.

ICA can also be viewed as a causal generative model [1, 2] that differs from factor analysis in two ways. First, the priors over the hidden activities remain independent but they are non-Gaussian. By itself, this modification would make it intractable to compute the posterior distribution over hidden activities. Tractability is restored by eliminating sensor noise and by using the same number of factors as input dimensions. This ensures that the posterior distribution over hidden activities collapses to a point. Interpreting ICA as a type of causal generative model suggests a number of ways in which it might be generalized, for instance to deal with more hidden units than input dimensions. Most of these generalizations retain marginal independence of the hidden activities and add sensor noise, but fail to preserve the property that the posterior distribution collapses to a point. As a result inference is intractable and crude approximations are needed to model the posterior distribution, e.g., a MAP estimate in [3], a Laplace approximation in [4, 5] or more sophisticated variational approximations in [6].

* Funded by the Wellcome Trust and the Gatsby Charitable Foundation.

2. ICA AS AN ENERGY-BASED DENSITY MODEL

We now describe a very different way of interpreting ICA as a probability density model. In the next section we describe how we can fit the model to data. The advantage of our energy-based view is that it suggests different generalizations of the basic ICA algorithm which preserve the computationally attractive property that the hidden activities are a simple deterministic function of the observed data. Instead of viewing the hidden factors h_j as stochastic latent variables in a causal generative model, we view them as deterministic functions $h_j(\mathbf{d}; \mathbf{w}_j)$ of the data with parameters $\mathbf{w}_j$. The hidden factors are then used for assigning an energy $E(\mathbf{d})$, to each possible observation vector $\mathbf{d}$:

$$E(\mathbf{d}) = \sum_j E_j(h_j(\mathbf{d}; \mathbf{w}_j)) \tag{1}$$

where j indexes over the hidden factors. The probability of $\mathbf{d}$ is defined in terms of its energy in the following way:

$$p(\mathbf{d}) = \frac{e^{-E(\mathbf{d})}}{\int_{\mathbf{c}} e^{-E(\mathbf{c})}\, d\mathbf{c}} \tag{2}$$

Standard ICA with non-Gaussian priors $p_j(h_j)$ is implemented by having the same number of hidden factors as input dimensions and using

$$h_j(\mathbf{d}, \mathbf{w}_j) = \mathbf{w}_j^T \mathbf{d} \tag{3}$$

$$E_j(h_j) = -\log p_j(h_j) \tag{4}$$

Furthermore, in this special case of standard ICA the normalization term in Eq. 2 (the "partition function") is tractable and simplifies to

$$\int_{\mathbf{c}} e^{-E(\mathbf{c})}\, d\mathbf{c} = |1/\det(W)| \tag{5}$$

where the rows of W are the filters $\mathbf{w}_j$.

The above energy-based model suggests thinking about ICA as a recognition model, where observations are linearly *filtered*, instead of as a causal generative model, where independent factors are linearly *mixed*. In [7] these filters are interpreted as linear constraints, with the energies serving as costs for violating the constraints. Using energies corresponding to heavy tailed distributions with a sharp peak at zero means that the constraints should be "frequently approximately satisfied", but will not be strongly penalized if they are grossly violated. In this new view it is very natural to include more constraints than input dimensions, which implies however that the hidden units will no longer be marginally independent. It is also natural to extend the model to have non-linear constraints. This extension will be further described in the discussion section. In the rest of this paper we will assume linear constraints as in Eq. 3.

3. CONTRASTIVE DIVERGENCE LEARNING

We now describe a novel method of fitting models of the type defined by Eqs. 1 and 2 to data. The method is not particularly efficient in the standard ICA case, but is very easy to generalize to models where the number of hidden units is not the same as the dimensionality of the observation vector, and h_j is any non-linear function. The fitting method is based on the "contrastive divergence" objective function introduced in [8].

Maximizing the log likelihood of the observed data under a model is equivalent to minimizing the Kullback-Leibler (KL) divergence, $(P^0||P^\infty)$, between the distribution, P^0, of the data and the distribution, P^∞, defined by the model. For the model in Eq. 2 the gradient of the KL divergence is:

$$\frac{\partial(P^0||P^\infty)}{\partial w_{ij}} = \left\langle \frac{\partial E}{\partial w_{ij}} \right\rangle_{P^0} - \left\langle \frac{\partial E}{\partial w_{ij}} \right\rangle_{P^\infty} \quad (6)$$

where the derivative of the partition function, given by the second term, is an average of the gradient over the model distribution. In the standard ICA setting this term is equal to $-(W^{-T})_{ji}$. An exact computation of this average, however, is intractable in general. Instead it is possible to approximate the average by generating samples using a Markov chain whose equilibrium distribution is the distribution defined by the model. In practice however, the chain has to be run for many steps before it approaches the equilibrium distribution and it is hard to estimate how many steps are required (hence the ∞ in "P^∞"). This means that fitting the model by following the gradient of the log likelihood is slow and can be unreliable.

The idea of contrastive divergence is to avoid the need to approach the stationary distribution by starting the Markov chain at the distribution of the observed data and then watching how it begins to diverge from the data distribution. Even if the chain is only run for a few steps, any consistent tendency to move away from the data distribution provides valuable information that can be used to adapt the parameters of the model. Intuitively, we need to modify the energy function so that the Markov chain does not tend to move away from the data distribution. This can be achieved by lowering the energy of points in the data space that have high probability in the data distribution but lower probability after a few steps of the Markov chain, and raising the energy of points whose probability rises after a few steps of the Markov chain. This intuitive idea can be understood as a way of improving the following contrastive divergence cost function:

$$CD = (P^0||P^\infty) - (P^n||P^\infty) \quad (7)$$

where P^n is the distribution obtained after running n steps of the Markov chain starting from P^0. Since a Markov chain has the property that the KL divergence from the stationary distribution never increases, the contrastive divergence can never become negative, and will be zero exactly when $P^0 = P^\infty$. The gradient of $(P^n||P^\infty)$ is:

$$\frac{\partial(P^n||P^\infty)}{\partial w_{ij}} = \left\langle \frac{\partial E}{\partial w_{ij}} \right\rangle_{P^n} - \left\langle \frac{\partial E}{\partial w_{ij}} \right\rangle_{P^\infty} + \frac{\partial(P^n||P^\infty)}{\partial P^n}\frac{\partial P^n}{\partial w_{ij}} \quad (8)$$

and the gradient of the contrastive divergence therefore is:

$$\frac{\partial CD}{\partial w_{ij}} = \left\langle \frac{\partial E}{\partial w_{ij}} \right\rangle_{P^0} - \left\langle \frac{\partial E}{\partial w_{ij}} \right\rangle_{P^n} - \frac{\partial(P^n||P^\infty)}{\partial P^n}\frac{\partial P^n}{\partial w_{ij}} \quad (9)$$

The last term in Eq. 9 represents the effect that changes in w_{ij} have on CD via the effect on P^n. This term is typically small and simulations in [8] suggest that it can be safely ignored. The results later in this paper also show that it is safe to ignore this term.

In summary, we propose the following algorithm to learn the parameters of the energy-based model defined by Eq. 2.

1. Compute the gradient of the total energy with respect to the parameters and average over the data cases $\mathbf{d}_k$.
2. Run MCMC samplers for n steps, starting at every data-vector $\mathbf{d}_k$, keeping only the last sample $\mathbf{s}_k$ of each chain.
3. Compute the gradient of the total energy with respect to the parameters and average over the samples $\mathbf{s}_k$.
4. Update the parameters using,

$$\delta w_{ij} = -\frac{\eta}{N}\left(\sum_{\text{data } \mathbf{d}_k} \frac{\partial E(\mathbf{d}_k)}{\partial w_{ij}} - \sum_{\text{samples } \mathbf{s}_k} \frac{\partial E(\mathbf{s}_k)}{\partial w_{ij}}\right) \quad (10)$$

where η is the learning rate and N the number of samples in each mini-batch.

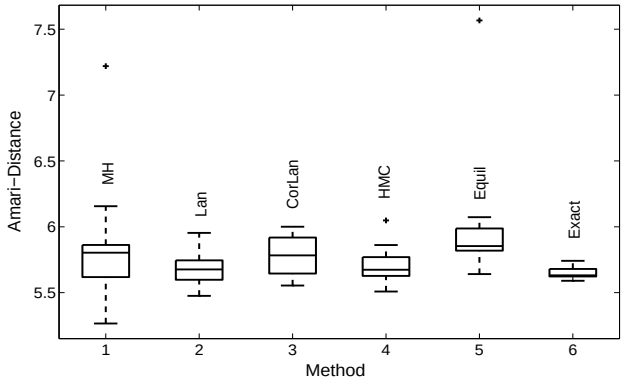

Fig. 1. Final Amari-Distance for the various algorithms, averaged over 10 runs. The boxes have lines at the lower quartile, median, and upper quartile values. The whiskers show the extent of the rest of the data. Outliers are denoted by "+". This plot shows that all the methods arrive at essentially the same result at the end of their runs, with the sampling methods having a higher variance than Exact. Although not shown here, we find that HMC is best able to scale well to high dimensions.

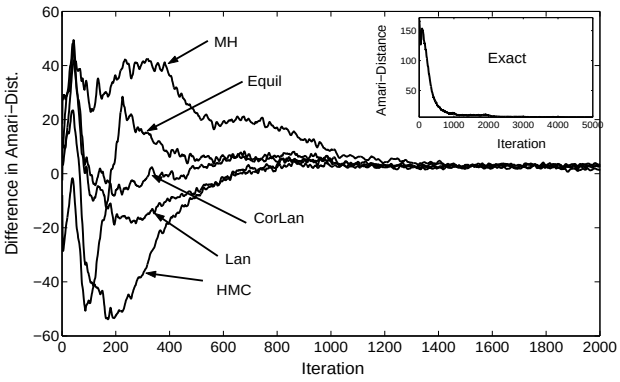

Fig. 2. Evolution of the Amari-Distance relative to Exact which is shown at the right upper corner. Note that the CD algorithms, which do *not* sample from the equilibrium distribution, converge just as fast as Equil.

If the shape of the energies E_j is parameterized as well, similar update rules as Eq. 9 can be used to fit them to data. For standard ICA, this corresponds to learning the shape of the prior densities.

4. EXPERIMENT: BLIND SOURCE SEPARATION

To assess the performance of the proposed contrastive divergence learning algorithm we compared 4 versions, differing in their MCMC-implementation, with an exact sampling algorithm as well as the Bell and Sejnowski algorithm on a standard "blind source separation" problem[1]. The model has the same number of hidden and visible units, and the energy of the model is defined as

$$E_j(h_j) = -\log(\sigma(h_j)(1-\sigma(h_j))) \qquad (11)$$

This model is strictly equivalent to the noiseless ICA model with sigmoidal outputs used by Bell and Sejnowski [9]. Below we will give a very brief description of the different MCMC methods, but refer to [10] for more details.

Algorithm HMC uses 1 step of hybrid Monte Carlo simulation to sample from $P^1(\mathbf{d})$. This involves sampling a momentum variable $\mathbf{k}$ from a standard normal distribution, followed by deterministic Hamiltonian dynamics starting at the data $\mathbf{d}$ such that the total energy $H = E + \frac{1}{2}|\mathbf{k}|^2$ is preserved. The dynamics is implemented by L leapfrog steps of size ϵ. A rejection rule is then used to correct for discretization error. CorLan resamples the momentum variable $\mathbf{k}$ after every leapfrog step. This is equivalent to using an isotropic normal distribution as the proposal distribution, with variance ϵ^2 but centered at $\mathbf{d} - \frac{1}{2}\epsilon^2 \frac{\partial E}{\partial \mathbf{d}}$. As $\epsilon \to 0$, we can show that the acceptance rate approaches 1. Hence assuming that step sizes are small enough, we can simplify and accept all proposals without incurring a loss in accuracy. This is called Langevin dynamics and implemented in Lan. On the other hand, if we use isotropic normal proposals centered on the current location $\mathbf{d}$, we get a simple Metropolis-Hastings sampler. This is used in MH. For noiseless ICA, it is possible to sample efficiently from the true equilibrium distribution. This is used in Equil. To be fair, we used a number of samples equal to the number of data vectors in each mini-batch. We can also compute the partition function using Eq. 5, and evaluate the second term of Eq. 6 exactly. This is precisely Bell and Sejnowski's algorithm and was implemented in Exact.

The data consisted of 16, 5-second stereo CD recordings of music, sampled at 44.1 kHz [2]. Each recording was down-sampled by a factor of 5, randomly permuted over the time-index and rescaled to unit variance. The resulting 88436 samples in 16 channels were linearly mixed using the standard *instamix* routine with $b = 0.5$ (1 on the diagonal and $1/9$ off the diagonal)[3], before presentation to the learning algorithm.

Parameter updates were performed on mini-batches of 100 data vectors. The learning rate was annealed from 0.05 down to 0.0005 in 5000 iterations of learning, while a momentum factor of 0.9 was used to speed up convergence. The initial weights were sampled from a Gaussian with std. 0.1. The number of sampling steps for MH, Lan, CorLan and HMC are 20 ,20, 10 and 20 respectively. The number of steps are chosen so that the amount of computation required for each algorithm is approximately equal. The step-sizes

[1] Note however that recovering more sound sources than input dimensions (sensors) is not possible with our energy-based model.

[2] http://sweat.cs.unm.edu/~ bap/demos.html

[3] http://sound.media.mit.edu/ica-bench/

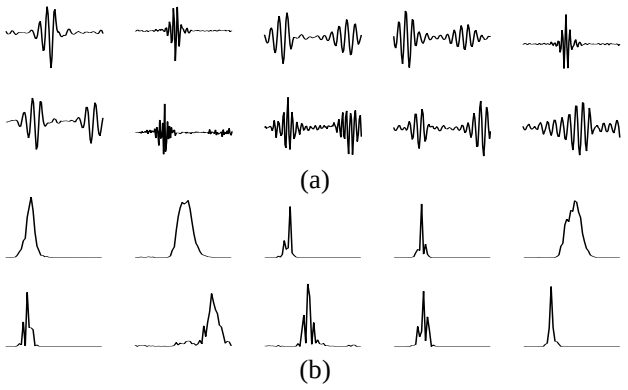

Fig. 3. (a) Filters found by the $2\times$ overcomplete HMC algorithm with 20 leapfrog steps and an acceptance rate of 90%. The first 5 filters are the ones with largest power, indicating that they represent important constraints. The last 5 filters are randomly drawn from the remaining 195 filters. (b) Corresponding power-spectra.

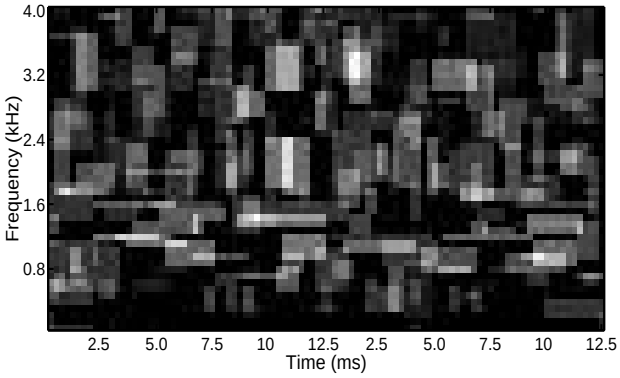

Fig. 4. Distribution of power over time and frequency. First the envelope of each filter (the absolute value of its Hilbert transform) was computed and squared. Next, the squared envelope and the power spectrum were thresholded by mapping all values greater than half the peak value to one and the rest to zero. Gaps smaller than 6 samples in time and 3 samples in frequency were filled in. Finally, the outer product of the two "templates" were computed, weighted by the total power of the filter, and added to the diagram.

of the MCMC samplers were adapted (at the end of each MC run) such that the acceptance rates were maintained at 0.5, 0.95, 0.5 and 0.9 respectively (for Lan this means that the step size was adapted such that if we were to use an accept/reject step, i.e. use CorLan instead, the acceptance rate would have been 0.95).

During learning we monitored the Amari-Distance[4] to the true unmixing matrix. In figures 1 and 2 we show the results of the various algorithms on the sound separation task. The main conclusion of this experiment is that we do not need to sample from the equilibrium distribution in order to learn the filters $\mathbf{W}$. This validates the ideas behind CD learning.

5. EXPERIMENT: INDEPENDENT COMPONENTS OF SPEECH

To test whether the model could extract meaningful filters from speech data we used recordings of 10 male speakers from the TIMIT database, uttering the sentence "Don't ask me to carry an oily rag like that". The sentences were downsampled to 8kHz and 50000, 12.5ms segments (corresponding to 100 samples) were extracted from random locations. Before presenting to the learning algorithm the data was centered and sphered.

The HMC implementation was used with 20 leapfrog steps. Mini-batches of size 100 were used, while the step size was annealed from 0.05 to 0.0005 in 20000 iterations. The filters were initialized at small random values and momentum was used to speed up convergence.

In figure 3 we show 10 of the 200 features that were extracted together with their power spectra. Recall that since there are 2 times more filters extracted as dimensions in the input space, the energy-based model is no longer equivalent to any ICA model. Figure 4 shows the distribution of power over time and frequency. There seems to be interesting structure around 1.5 kHz, where the filters are less localized and more finely tuned in frequency than average. This phenomenon is also reported in [12].

6. EXPERIMENT: NATURAL IMAGES

We tested our algorithm on the standard ICA task of determining the "independent" components of natural images. The data set used was obtained from van Hateren's website[5] [13]. The logarithm of the pixel intensities was first taken and then the image patches were centered and whitened with ZCA. There were 122880 patches and each patch was 16×16 in size.

We trained up a network with $256\times3=768$ features, with energy functions of the form

$$E_j(h_j(\mathbf{d},\mathbf{w}_j)) = \gamma_j \log\left(1+(\mathbf{w}_j^T\mathbf{d})^2\right) \qquad (12)$$

where the filters $\mathbf{w}_j$ and the weights γ_j were adapted with CD learning. The HMC algorithm was used with 30 leapfrog steps and an adaptive step size so that the acceptance rate is approximately 90%. Both $\mathbf{w}_j$ and γ_j are unconstrained, but a small weight decay of 10^{-4} was used for $\mathbf{w}_j$ to encourage the features to localize. The $\mathbf{w}_j$'s were initialized to random

[4]The Amari-Distance [11] measures a distance between two matrices A and B up to permutations and scalings: $\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{|(AB^{-1})_{ij}|}{\max_k|(AB^{-1})_{ik}|}+\frac{|(AB^{-1})_{ij}|}{\max_k|(AB^{-1})_{kj}|}\right)-2N^2$.

[5]ftp://hlab.phys.rug.nl/pub/samples/imlog

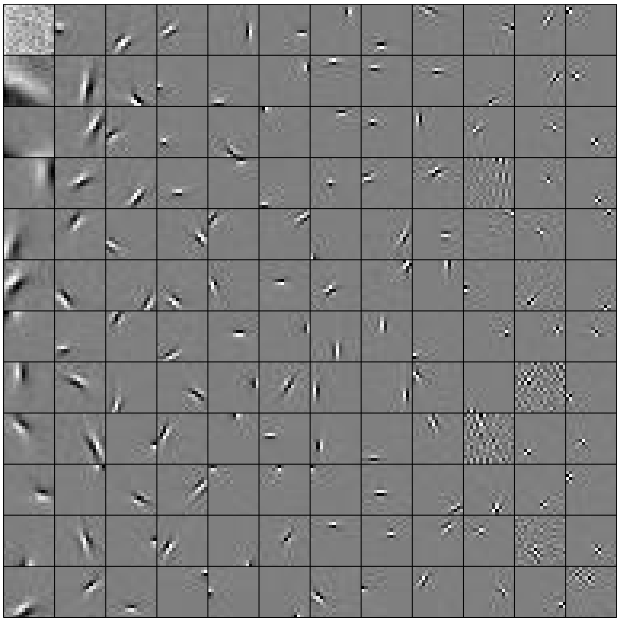

Fig. 5. Learned filters for natural images

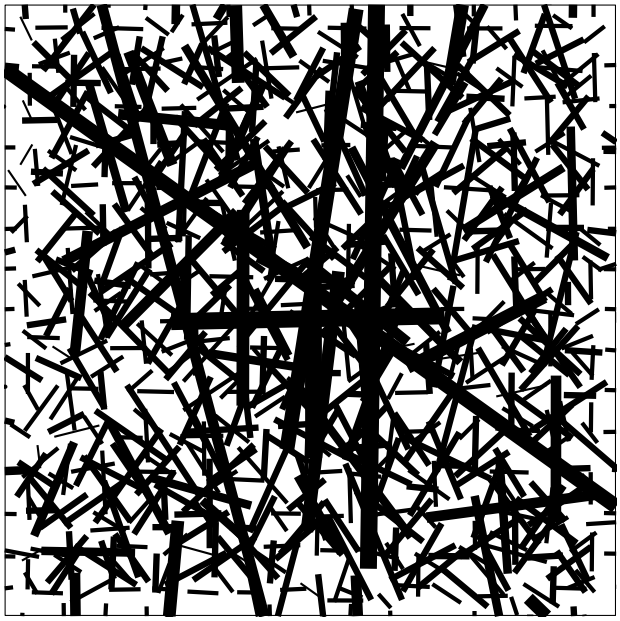

Fig. 6. The spatial layout and size of the filters, which are described by the position and size of the bars.

vectors of norm 1, while the γ_j's were initialized at 1. Both $\mathbf{w}_j$ and γ_j were learned with a learning rate of 0.01 and momentum factor of 0.9. We found however that the result is not sensitive to the settings of these parameters.

A random sample of 144 learned features is shown in figure 5. They were roughly ordered by increasing spatial frequency. By hand, we counted a total of 19 features which have not localized either in the spatial or frequency domain. Most of the features can be described well with Gabor functions. To further analyze the set of learned filters, we fitted a Gabor function of the form used in [5] to each feature and extracted parameters like spatial frequency, location and extent in the spatial and frequency domains. These are summarized in figures 6 and 7, and show that the filters form a nice tiling of both the spatial and frequency domains. We see from figures 5 and 7 that filters are learned at multiple scales, with larger features typically being of lower frequency. However we also see an over emphasis of horizontal and vertical filters. This effect has been observed in previous papers [13, 5], and is probably due to pixellation.

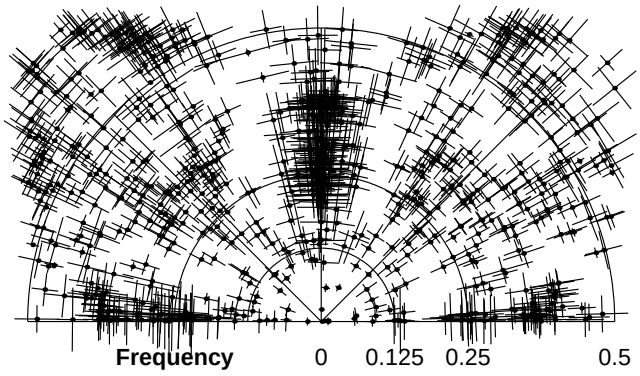

Fig. 7. A polar plot of frequency tuning and orientation selectivity of the learned filters, with the center of each cross at the peak frequency and orientation response, and crosshairs describing the $1/16$-bandwidth.

7. DISCUSSION

We have shown that ICA can be viewed as an energy-based probability density model, and can as such be trained using the contrastive divergence algorithm. Framing ICA in this new way suggests simple, novel extensions which retain many of the attractive properties of the original ICA algorithm. In particular the framework makes it very easy to deal with overcomplete and non-linear models.

This energy-based view of ICA stems from our previous work on products of experts (PoEs) [8]. In fact our model is a type of PoE in which each energy term corresponds to one expert. The PoE framework is related to the maximum entropy framework advocated by some researchers [14, 15]. The difference is that in PoEs the features come from a parameterized family and are fitted together with weights using CD-based gradient descent, while in the maximum entropy literature the features are usually either fixed or induced in an outer loop and the weights are learned with variants of iterative scaling. As a matter of fact in Eq. 12 the features are $\log(1+(\mathbf{w}_j^T\mathbf{d})^2)$ with γ_j being the correspond-

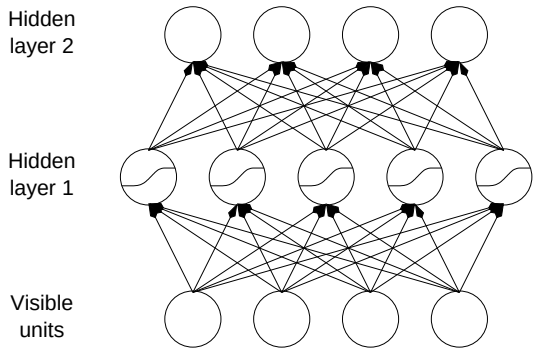

Fig. 8. Architecture of a hierarchical non-linear energy-based model. Non-linearities are indicated by sigmoidal units in hidden layer 1. Energies are contributed by hidden units in *both* layers, and the number of hidden units need not correspond to the number of visible units.

ing maximum entropy weights.

Our present work is actively exploring a hierarchical non-linear architecture (figure 8) in which the hidden activities are computed with a feed-forward neural network (see [16] for related work). To fit this model to data, back-propagation is used to compute gradients of the energy with respect to both the data vector (to be used in MCMC sampling), and the weights (to be used for weight updates).

8. ACKNOWLEDGEMENTS

We would like to thank Peter Dayan, Sam Roweis, Zoubin Ghahramani and Maneesh Sahani for helpful discussions, and Carl Rasmussen for making *minimize.m* available.

9. REFERENCES

[1] B. A. Pearlmutter and L. C. Parra, "A context-sensitive generalization of ICA," in *Proceedings of the International Conference on Neural Information Processing*, 1996.

[2] D. MacKay, "Maximum likelihood and covariant algorithms for independent components analysis," http://wol.ra.phy.cam.ac.uk/mackay/abstracts/ica.html, 1996.

[3] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[4] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations.," *Neural Computation*, vol. 12, pp. 337–365, 2000.

[5] M. S. Lewicki and B. A. Olshausen, "A probabilistic framework for the adaptation and comparison of image codes," *J. Opt. Soc. Am. A: Optics, Image Science, and Vision*, vol. 16, no. 7, pp. 1587–1601, 1999.

[6] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, pp. 803–851, 1998.

[7] G. E. Hinton and Y. W. Teh, "Discovering multiple constraints that are frequently approximately satisfied," in *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, 2001, pp. 227–234.

[8] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, 2001, In press.

[9] A. J. Bell and T. J. Sejnowski, "An information maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[10] R. M. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," Tech. Rep. CRG-TR-93-1, University of Toronto, Department of Computer Science, 1993.

[11] S. Amari, A. Cichocki, and H. Yang, "A new algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*, 1996, vol. 8, pp. 757–763.

[12] S. A. Abdallah and M. D. Plumbley, "If edges are the independent components of natural images, what are the independent components of natural sounds?," in *International Conference On Independent Component Analysis and Blind Signal Separation*, 2001.

[13] 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 B*, vol. 265, pp. 359–366, 1998.

[14] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[15] S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modelling," *Neural Computation*, vol. 9, no. 8, 1997.

[16] A. Hyvrinen and P. O. Hoyer, "A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images," *Vision Research*, vol. 41, no. 18, pp. 2413–2423, 2001.