Scaling in a Hierarchical Unsupervised Network[1]

Zoubin Ghahramani,[2] **Alexander T. Korenberg** and **Geoffrey E. Hinton**

Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR, U.K.

http://www.gatsby.ucl.ac.uk/

Abstract

A persistent worry with computational models of unsupervised learning is that learning will become more difficult as the problem is scaled. We examine this issue in the context of a novel hierarchical, generative model that can be viewed as a non-linear generalization of factor analysis and can be implemented in a neural network. The model performs perceptual inference in a probabilistically consistent manner by using top-down, bottom-up and lateral connections. These connections can be learned using simple rules that require only locally available information. We first demonstrate that the model can extract a sparse, distributed, hierarchical representation of global disparity from simplified random-dot stereograms. We then investigate some of the scaling properties of the algorithm on this problem and find that: (1) Increasing the image size leads to faster and more reliable learning; (2) Increasing the depth of the network from one to two hidden layers leads to better representations at the first hidden layer, and (3) Once one part of the network has discovered how to represent disparity, it "supervises" other parts of the network, greatly speeding up their learning.

Introduction

In order to understand how a perceptual system can learn without any supervision it is useful to define the notion of a generative model. A generative model is a probabilistic model of how the underlying physical properties of the world cause sensory data. For example, an imaging model relates surface properties, spatial relationships and lighting conditions to the intensities detected on the retina. The generative model provides a rigorous basis for perceptual inference. By inverting the generative model, the perceptual system can infer the probabilities of different causes for the sensory data. By adhering to its generative model it can allow top-down expectations to combine with bottom-up inputs while maintaining a probabilistically consistent interpretation of the world. A generative model also provides a sensible objective function for unsupervised learning. Learning can be viewed as maximizing the likelihood of the observed data under the generative model, which is mathematically equivalent to discovering efficient ways of coding the sensory data.

Many models of how cortex learns can be understood in terms of two relatively simple generative models developed by statisticians. On the one side are clustering models, typified by the mixture of Gaussians. In this model, the sensory data is assumed to have been generated by picking one of K possible prototypes and adding Gaussian noise. The goal of learning is to determine the K prototypes that best fit the data and the variance of the Gaussian noise for each of the sensory units. Competitive learning algorithms (*e.g.* Rumelhart and Zipser, 1985; Carpenter and Grossberg, 1988) can generally be viewed as ways of fitting mixture of Gaussian generative models. Kohonen's self-organizing maps (Kohonen, 1982) and Durbin and Willshaw's elastic net (1987) are variations of mixture of Gaussian models in which additional constraints are imposed that force neighboring hidden units to have similar generative weight vectors. These constraints typically lead to a model of the data that is worse when measured by the likelihood of the data.

On the other side are dimensionality reduction models, typified by factor analysis (Everitt, 1984). In factor analysis, the D-dimensional sensory data is assumed to have been generated by linearly combining K independent Gaussian variables, the factors, and then adding Gaussian noise. The goal of learning is to find the linear transformation from the K factors that maximizes the likelihood of the sensory data. This goal is only well-defined for $K < D$, and therefore the factors can be thought of as a reduced dimensionality representation of the sensory data. In the limit where the variance of the noise added to each of the D dimensions of the sensory data is assumed to go to zero, factor analysis reduces to principal components analysis (PCA). Unsupervised learning models based on Hebbian learning can generally be viewed as implementing variants of PCA (Oja, 1982). Models of this kind have been found to develop center-surround and orientation-selective properties similar to those of cells in the visual system (Linsker, 1988).

In this paper, we first describe the need for models that go beyond factor analysis and mixtures of Gaussians. The goal of these models is to discover hierarchical distributed representations that are non-linearly related to the perceptual data. We briefly review previous attempts at developing such a model. We then present a new model, the rectified Gaussian belief net (Hinton and Ghahramani, 1997). This model makes strong suggestions about the role of both top-down and lateral connections in cortex and it also suggests why topographic maps are so prevalent. Using simplified random-dot stereograms we show that this model discovers a hierarchical distributed representation of global disparity. Finally, we examine the scaling properties of the model on the stereogram

[1] A slightly shorter version of this paper will appear as Ghahramani, Z., Korenberg, A., and Hinton, G.E. (1999) Scaling in a Hierarchical Unsupervised Network. In *ICANN 99: Ninth international conference on Artificial Neural Networks*.

[2] Email: zoubin@gatsby.ucl.ac.uk

problem.

Sparse Distributed Representations

Factor analysis and mixtures of Gaussians are at opposite ends of a spectrum of possible learning algorithms. In factor analysis, the representation is "componential" or "distributed" because it involves states in all of the hidden units. However, it is also linear and is therefore limited to capturing the information in the pairwise covariances of the visible units. All the higher-order structure is invisible to it. At the other end of the spectrum, mixtures of Gaussians have localist representations because each data vector is assumed to be generated from a single hidden unit. This is an exponentially inefficient representation: each datapoint is represented by the identity of the winning hidden unit (*i.e.* the cluster it belongs to). So for the representation to contain, on average, n bits of information about the data, there must be at least 2^n hidden units. However, it is non-linear and with enough hidden units it can capture all of the higher-order structure in the data.

The really interesting generative models lie in the middle of the spectrum. They use non-linear distributed representations of the type advocated by Barlow (1989) and Olshausen and Field (1996). To see why such representations are needed, consider a typical image that contains multiple objects. To represent the pose and deformation of each object we want a componential representation of the object's parameters. To represent the multiple objects we need several of these componential representations at once.

The difficulty with such models lies in the computation of the posterior distribution over hidden states when given a datapoint. This distribution, or an approximation to it, is required both for learning the generative model and for perceptual inference once the model has been learned. Mixtures of Gaussians and factor analysis are standard statistical models precisely because the exact computation of the posterior distribution is tractable. For models with non-linear distributed representations, computing the posterior distribution (or even the most probable state) of the hidden units given a data point is in general intractable, as it involves considering all exponentially-many possible settings of the hidden units.

Scaling

One worry with such models is that while the approximations commonly used for learning and inference may work on small problems, they may not scale well to larger problems with more realistic datasets. This has certainly been the experience with combinatorial optimization problems (such as the travelling salesman problem) in which the best solution to one part of the problem is usually incompatible with the best solution to another part of the problem. This is called a "frustrated" system and is just what vision is not like. It is generally easier to interpret two neighboring patches of an image than to interpret one patch in isolation because context almost always facilitates interpretation. In the latter part of this paper we test the conjecture that, for a vision problem such as discovering depth from random-dot stereograms, scaling both the size of the input and the number of hidden layers will lead to faster rather than slower perceptual inference and learning.

Rectified Gaussian Belief Nets

We now describe a new model called the Rectified Gaussian Belief Net (RGBN) that combines sparse distributed representations with a hierarchical structure. The RGBN uses units with states that are either positive real values or zero, so it can represent real-valued latent variables directly. Its main disadvantage is that the recognition process involves Gibbs sampling which could be very time consuming. In practice, however, 10 to 20 samples per unit have proved adequate for some small but interesting tasks.

We first describe the RGBN without considering neural plausibility. Then we show how lateral interactions within a layer can be used to perform explaining away correctly[3]. This makes the RGBN far more plausible as a neural model and leads to a very natural explanation for the prevalence of topographic maps in cortex.

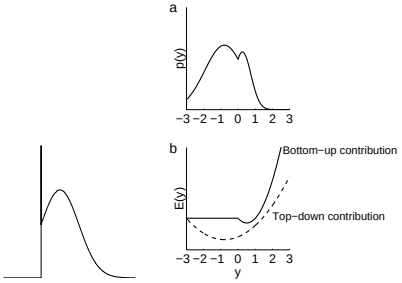


Figure 1: **(left)** The rectified Gaussian. **(right) a** Schematic of the posterior density of an unrectified state of a unit. **b** Bottom-up and top-down energy functions corresponding to **a**.

The generative model for RGBN's consists of multiple layers of units each of which has a real-valued unrectified state, y_j, and a rectified state, $\tilde{y}_j$, which is zero if y_j is negative and equal to y_j otherwise. This rectification is the only non-linearity in the network. The value of y_j is Gaussian distributed with a standard deviation σ_j and mean, $\hat{y}_j$ that is determined by the generative bias, g_{0j}, and the combined effects of the rectified states of units, k, in the layer above:

$$\hat{y}_j = g_{0,j} + \sum_k \tilde{y}_k g_{kj} \tag{1}$$

The rectified state $\tilde{y}_j$ therefore has a Gaussian distribution above zero, but all of the mass of the Gaussian that falls below zero is concentrated in an infinitely dense spike at zero as shown in figure 1. This infinite density creates problems if we attempt to use Gibbs sampling over the rectified states, so we perform Gibbs sampling on the unrectified states.

Sampling from the posterior distribution

Consider a unit, j, in some intermediate layer of a multilayer RGBN. Suppose that we fix the unrectified states of all the

[3]"Explaining away" refers to the situation where causes that are *a priori* independent become dependent conditioned on some observed effect. For example, having the sprinkler on and whether it rains or not may be *a priori* independent. Now, assume that we observe that the ground is wet. Either cause can explain away the observation. One of them is probably true, but both of them together are unlikely, and therefore the two causes may be *a posteriori* anti-correlated.

other units in the net. To perform Gibbs sampling, we need to stochastically select a value for y_j according to its posterior distribution given the unrectified states of all the other units.

If we think in terms of energy functions, which are equal to the negative log probabilities (up to a constant), the rectified states of the units in the layer above contribute a quadratic energy term by determining $\hat{y}_j$. The unrectified states of units, i, in the layer below contribute nothing if $\tilde{y}_j$ is 0, and if $\tilde{y}_j$ is positive they each contribute a quadratic term because of the effect of $\tilde{y}_j$ on $\hat{y}_i$.

$$E(y_j) = \frac{(y_j - \hat{y}_j)^2}{2\sigma_j^2} + \sum_i \frac{(y_i - \sum_h \tilde{y}_h g_{hi})^2}{2\sigma_i^2} \qquad (2)$$

where h is an index over all the units in the same layer as j including j itself. Terms that do not depend on y_j have been omitted from Eq. 2. For values of y_j below zero there is a quadratic energy function which leads to a Gaussian posterior distribution. The same is true for values of y_j above zero, but it is a different quadratic (see figure 1b). The Gaussian posterior distributions corresponding to the two quadratics must agree at $y_j = 0$ (figure 1a). Because the posterior distribution is piecewise Gaussian it is possible to perform Gibbs sampling exactly and fairly efficiently.

Learning the parameters of an RGBN

Given samples from the posterior distribution, the generative weights of a RGBN can be learned by using the online delta rule:

$$\Delta g_{ji} = \epsilon \tilde{y}_j (y_i - \hat{y}_i) \qquad (3)$$

The variance of the local Gaussian noise of each unit, σ_j^2, can be also learned by an online rule:

$$\Delta \sigma_j^2 = \epsilon[(y_j - \hat{y}_j)^2 - \sigma_j^2] \qquad (4)$$

Alternatively, σ_j^2 can be fixed at 1 for all hidden units and the effective local noise level can be controlled by scaling the generative weights.

The Role of Lateral Connections

Lee and Seung (1997) introduced a clever way of using lateral connections to handle explaining away effects. Consider the network shown in figure 2. One contribution, E, to the energy of the state of the network is the squared difference between the unrectified states of the units in the bottom layer, y_j, and the top-down expectations generated by the states of units in the layer above. Assuming the local noise models for the visible units all have unit variance, and ignoring biases and constant terms that are unaffected by the states of the units

$$E = \sum_j (y_j - \hat{y}_j)^2 = \sum_j (y_j - \sum_k y_k g_{kj})^2. \qquad (5)$$

This expression can be rearranged to give

$$E = \sum_j y_j^2 - 2\sum_k y_k \sum_j y_j g_{kj} - \sum_k \sum_l y_k y_l (-\sum_j g_{kj} g_{lj}). \qquad (6)$$

Setting $r_{jk} = g_{kj}$ and $m_{kl} = -\sum_j g_{kj} g_{lj}$ we get

$$E = \sum_j y_j^2 - 2\sum_k y_k \sum_j y_j r_{jk} - \sum_k y_k \sum_l y_l m_{kl}. \qquad (7)$$

This energy function can therefore be implemented in a network with recognition weights, r_{jk}, and symmetric lateral interactions, m_{kl}. The lateral and recognition connections allow a unit, k, to compute how E for the layer below depends on its own state and therefore they allow it to follow the gradient of E or to perform Gibbs sampling in E.

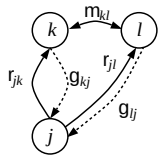


Figure 2: A network with lateral connections to handle explaining away effects.

If we are willing to use Gibbs sampling, Seung's trick allows a proper implementation of factor analysis in a neural network because it makes it possible to sample from the full covariance posterior distribution in the hidden state space. Seung's trick can also be used in an RGBN and it eliminates the most neurally implausible aspect of this model which is that a unit in one layer appears to need to send both its state y and the top-down prediction of its state $\hat{y}$ to units in the layer above. Using the lateral connections, the units in the layer above can, in effect, compute all they need to know about the top-down predictions.

In computer simulations, we can simply set each lateral connection m_{kl} to be $-\mathbf{g}_k \cdot \mathbf{g}_l$. It is also possible to learn these lateral connections in a more biologically plausible way by driving units in the layer below with unit-variance independent Gaussian noise and using a simple anti-Hebbian learning rule. Similarly, a purely local learning rule can be implemented to learn recognition weights equal to the generative weights. If units at one layer are driven by unit-variance independent Gaussian noise, and these in turn drive units in the layer below using the generative weights, then Hebbian learning between the two layers will learn the correct recognition weights (Hinton and Ghahramani, 1997).

There is one remaining difficulty that is a consequence of our decision to perform Gibbs sampling on the unrectified states. A unit needs to send its unrectified state to units in the layer above and its rectified state to units in the layer below. In the simulations we report in this paper we do not implement RGBNs using the lateral connection trick or the more biologically plausible learning rules. In another paper we have explored the use of lateral connections for inference (as described here) and also to induce topographic self-organisation of the features in the hidden layer (Ghahramani and Hinton, 1998).

Discovering disparity in simplified stereo pairs

A problem in which discovering the higher order structure of a dataset has presented difficulties for some previous unsupervised learning algorithms is the one-dimensional stereo disparity problem (Becker and Hinton, 1992). Consider

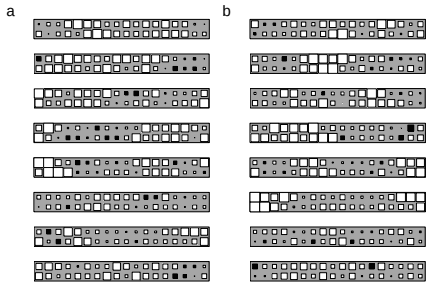
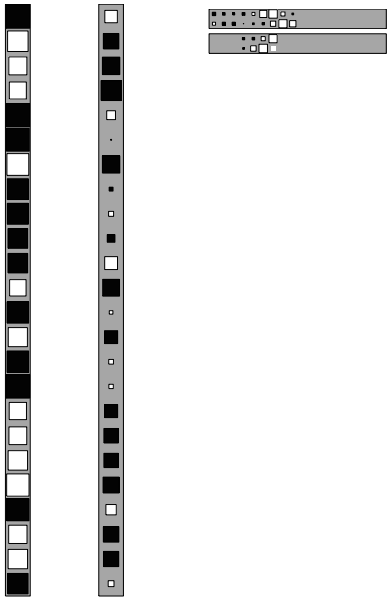


Figure 3: **a** Sample data from the simplified stereo disparity problem. The top and bottom row of each 2×18 image are the inputs to the left and right eye, respectively. **b** Sample outputs generated by the model after learning.



fect on learning of different connectivity patterns (i.e. fan-in) between visible and hidden units. This is of interest for two reasons. First, it is not clear a priori what the effect of scaling the number of inputs to each hidden unit will have. Second, we wished to determine an optimal fan-in to be used in subsequent experiments. For the 1-48-72 architecture, we varied the fan-in between 8 (4 connections from each eye) and 72 (full connectivity). Throughout learning, the networks were tested on test sets of 1000 noisy images to determine whether the top unit had discovered a representation of disparity. The learning curves, measured by the percent correct disparity inferred by the top unit, varied considerably as a function of fan-in (figure 5). A fan-in of 18 (9 connections from each eye) resulted in faster learning and higher asymptote than larger or smaller fan-ins. This is probably due to the fact that the basic correlation length of features in the image is on the order of 2-3 pixels. A fan-in of 8 (4 pixels per eye) gives a low probability that a hidden unit will have an entire feature in its receptive field, while fan-ins of 36 and 72 will recult in many features in each hidden unit's receptive field, and the hidden unit has to learn to ignore all but one of them. We used a fan-in of 18 for the remainder of the experiments.

Exp 2: Larger images improve learning

The goal of the second experiment was to determine the effect of scaling the number of visible and hidden units. We compared learning and inference on the stereo disparity problem in four different network sizes: 1-12-18, 1-24-36, 1-48-72, and 1-72-108, all with a fan-in of 18.

Using the same measure as in experiment 1, we evaluated learning of disparity for each of these architectures. The results clearly indicate that larger networks learn a respresentation of disparity faster and with a higher performance asymptote than smaller networks (figure 6). This experiment suggests that, for this simple vision problem, increasing the network size to accommodate a larger image leads to faster rather than slower learning.

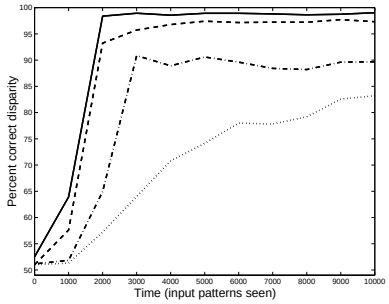


Figure 6: Percent correct disparity as a function of learning time averaged over 5 runs for networks of differing size: 1-72-108 (solid), 1-48-72 (dashed), 1-24-36 (dot-dashed), 1-12-18 (dotted).

We also examined the speed of perceptual inference, as measured by the convergence of the activity of the top layer unit, for these different network sizes after learning. The top unit activity converged to 0 or 1 more quickly and reliably for larger networks, as indicated by the mean and standard deviation of the activity as a function of Gibbs samples (figure 7). Since the amount of depth information is greater in larger networks this is not surprising. However, had we not found this scaling behaviour, it would have been difficult to justify using Gibbs sampling in larger networks.

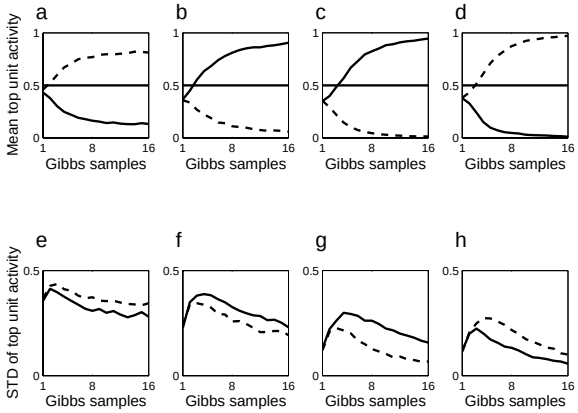


Figure 7: Mean **a–d** and standard deviation **e–h** activity of the top unit during Gibbs sampling after learning when the clamped image had leftward (solid) and rightward (dashed) disparity; averaged over 1000 images. Differing size networks: 1-12-18 (**a,e**), 1-24-36 (**b,f**), 1-48-72 (**c,g**), 1-72-108 (**d,h**).

Exp 3: Deeper networks learn better hidden representations

The goal of this experiment was to investigate the effect of network depth on learning. We compared a 1-24-36 network with a 0-24-36 network, i.e. an identical network lacking the unit at the top layer. Under one hypothesis, the presence of the top unit should slow learning since at first it is simply introducing noise and spurious correlations in the layer below. On the other hand, the network may be able to use the top unit to clean up representations at the level below and therefore speed learning.

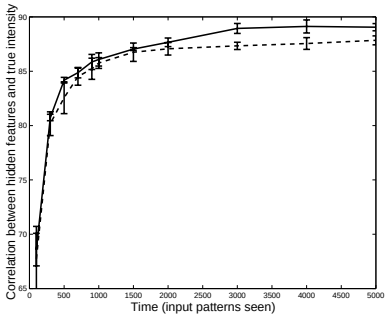


Figure 8: Average of 5 runs each of networks with a top layer hidden unit (solid lines) and without one (dashed lines).

To compare the networks with and without a unit at the top layer, we calculated the percent of variance accounted for by the best linear reconstruction of the noise-free features in the image that could be obtained from the middle hidden layer representations. This measures how faithfully the middle hidden units have captured the image features, independently of the generative weights from hidden to visible units. This measure increased very rapidly at the beginning of learning for both types of networks. After this initial phase, the networks with a top hidden unit had a consistently better middle

hidden representation than the networks without a top hidden unit, although the effect was small (figure 8).

We also looked at networks with 2 hidden units in the top layer, but the behaviour of these was not significantly different from networks with one hidden unit (data not shown).

Exp 4: One sub-network can teach another

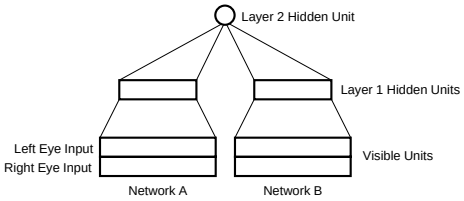


Figure 9: The double-network architecture, with two banks of 2×9 visible units, two banks of 12 first layer hidden units, and a single shared second layer hidden unit.

The aim of the last experiment was to see if connecting a network that had already learned to represent disparity from one part of the image to a second network with random weights would result in faster learning in the second network. We used a double network architecture consisting of two 1-12-18 networks sharing the top hidden unit, as shown in figure 9. The two networks had no interconnections (other than the shared top unit) and saw two different images with the same disparity. We will call one side of the double network the "teacher" and the other side the "student". We took a 1-12-18 network that had learned disparity (the teacher) and attached a similar network with random weights (the student). During learning, we compared the percent correct disparity inferred under three conditions: (1) inference using the combined student-teacher network (solid lines); (2) inference using only the student part of the network (dashed line); and (3) a control in which the student was trained on its own without a teacher (figure 10). The results suggest that the student network greatly benefits from having the top unit in common with the teacher network.

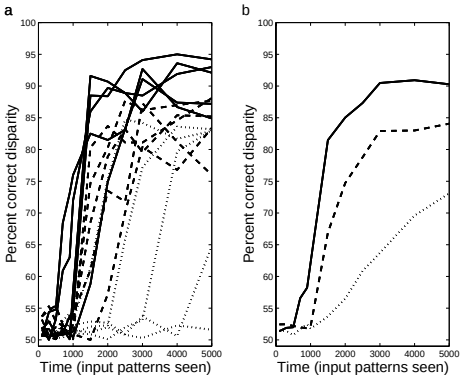


Figure 10: The percent correct disparity for the combined student-teacher network (solid lines), for the student tested alone (dashed lines), and for a control network of the same size as the student (dotted line). We show all five runs of each configuration (**a**) and the averages (**b**).

Discussion

In this paper we have shown empirically that increasing the width and depth of a hierarchical network can result in faster learning and inference. To our knowledge, this is the first systematic study of scaling properties of an unsupervised learning algorithm for a hierarchical generative model. It is important not to overstate the generalisability of these results: we have explored a single learning algorithm on a single problem. However, we believe that these results are significant and encouraging for the following reason: in everyday perception there is a great deal of redundancy across space, time, and different modalities. Plausible models of unsupervised learning in the brain and sensible unsupervised pattern recognition systems should be able to make good use of this redundancy. The experiments in this paper provide evidence that nonlinear hierarchical networks fit this criterion.

Acknowledgements

ZG and GEH were funded by grants from the Canadian NSERC, the Ontario ITRC and the Gatsby Charitable Foundation. ATK was funded by the Wellcome Trust 4 year PhD in Neuroscience Programme.

References

Barlow, H. (1989). Unsupervised learning. *Neur. Comp.*, 1:295–311.

Becker, S. and Hinton, G. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355:161–163.

Carpenter, G. and Grossberg, S. (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, pages 77–88.

Durbin, R. and Willshaw, D. (1987). An analogue approach to the travelling salesman problem using an elastic net method. *Nature*, 326(16):689–691.

Everitt, B. S. (1984). *An Introduction to Latent Variable Models*. Chapman and Hall, London.

Ghahramani, Z. and Hinton, G. E. (1998). Hierarchical non-linear factor analysis and topographic maps. In *Adv.in Neural Information Processing Systems 10*. MIT Press, Cambridge, MA.

Hinton, G. E. and Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Phil. Trans. Roy. Soc. London B: Biol. Sci.*, 352:1177–1190.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Lee, D. D. and Seung, H. S. (1997). Unsupervised learning by convex and conic coding. In *Adv. in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.

Linsker, R. (1988). Self-organization in a perceptual network. *IEEE Computer*, 21:105–117.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *J. of Mathematical Biology*, 15(3):267–273.

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

Rumelhart, D. and Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, 9:75–112.