The Helmholtz Machine

Peter Dayan[1] Geoffrey E Hinton[1] Radford M Neal[1] Richard S Zemel[2]

[1]Department of Computer Science
University of Toronto
6 King's College Road
Toronto, Ontario M5S 1A4
Canada

[2]CNL
The Salk Institute
PO Box 85800
San Diego, CA 92186-5800
USA

14[th] December 1994

Abstract

Discovering the structure inherent in a set of patterns is a fundamental aim of statistical inference or learning. One fruitful approach is to build a parameterised stochastic generative model, independent draws from which are likely to produce the patterns. For all but the simplest generative models, each pattern can be generated in exponentially many ways. It is thus intractable to adjust the parameters to maximize the probability of the observed patterns, We describe a way of finessing this combinatorial explosion by maximising an easily computed lower bound on the probability of the observations. Our method can be viewed as a form of hierarchical self-supervised learning that may relate to the function of bottom-up and top-down cortical processing pathways.

1 Introduction

Following Helmholtz, we view the human perceptual system as a statistical inference engine whose function is to infer the probable causes of sensory input. We show that a device of this kind can learn how to perform these inferences without requiring a teacher to label each sensory input vector with its underlying causes. A *recognition* model is used to infer a probability distribution over the underlying causes from the sensory input, and a separate *generative* model, which is also learned, is used to train the recognition model (Zemel, 1994; Hinton & Zemel, 1994; Zemel & Hinton, 1994).

As an example of the generative models in which we are interested, consider the shift patterns in Figure 1, which are on four 1×8 rows of binary pixels. These were produced by a two-level stochastic hierarchical generative process described in the figure caption. The task of learning is to take a set of examples generated by such a process and induce the model. Note that underlying any pattern there are multiple simultaneous causes. We call each possible set of causes an *explanation* of the pattern. For this particular example, it is possible to infer a unique set of causes for most patterns, but this need not always be the case.

For general generative models, the causes need not be immediately evident from the surface form of patterns. Worse still, there can be an exponential number of possible explanations underlying each pattern. The computational cost of considering all of these explanations makes standard maximum likelihood approaches such as the Expectation–Maximisation algorithm (Dempster *et al*, 1977) intractable. In this paper we describe a tractable approximation to maximum likelihood learning implemented in a layered hierarchical connectionist network.

2 The Recognition Distribution

The log probability of generating a particular example, d, from a model with parameters θ is

$$\log p(d|\theta) = \log \left(\sum_\alpha p(\alpha|\theta) p(d|\alpha, \theta) \right) \tag{1}$$

where the α are explanations. If we view the alternative explanations of an example as alternative configurations of a physical system there is a precise analogy with statistical physics. We define the energy of explanation α to be

$$E_\alpha(\theta, d) = -\log p(\alpha|\theta) p(d|\alpha, \theta) \tag{2}$$

The posterior probability of an explanation given d and θ is related to its energy by the equilibrium or Boltzmann distribution, which at a temperature of 1 gives:

$$P_\alpha(\theta, d) = \frac{p(\alpha|\theta) p(d|\alpha, \theta)}{\sum_{\alpha'} p(\alpha'|\theta) p(d|\alpha', \theta)} = \frac{e^{-E_\alpha}}{\sum_{\alpha'} e^{-E_{\alpha'}}} \tag{3}$$

where indices θ and d in the last expression have been omitted for clarity. Using E_α and P_α equation 1 can be rewritten in terms of the Helmholtz free energy, which is the difference between the expected energy of an explanation and the entropy of the probability distribution across explanations.

$$\log p(d|\theta) = -\left[\sum_\alpha P_\alpha E_\alpha - \left(-\sum_\alpha P_\alpha \log P_\alpha \right) \right] \tag{4}$$

So far, we have not gained anything in terms of computational tractability because we still need to compute expectations under the posterior distribution P which, in general, has exponentially many terms and cannot be factored into a product of simpler distributions. However, we know (Thompson, 1988) that any probability distribution over the explanations will have at least as high a free energy as the Boltzmann distribution (equation 3). Therefore we can restrict ourselves to some class of tractable distributions and still have a lower bound on the log probability of the data. Instead of using the true posterior probability distribution, P, for averaging over explanations, we use a more convenient probability distribution, Q. The log probability of the data can then be written as

$$\begin{aligned} \log p(d|\theta) &= -\sum_\alpha Q_\alpha E_\alpha - \sum_\alpha Q_\alpha \log Q_\alpha \quad + \sum_\alpha Q_\alpha \log[Q_\alpha / P_\alpha] \\ &= \qquad -F(d;\theta,Q) \qquad\qquad + \sum_\alpha Q_\alpha \log[Q_\alpha / P_\alpha] \end{aligned} \tag{5}$$

where F is the free energy based on the incorrect or non-equilibrium posterior Q.

Making the dependencies explicit, the last term in equation 5 is the Kullback-Leibler divergence between $Q(d)$ and the posterior distribution, $P(\theta, d)$ (Kullback, 1959). This term cannot be negative, so by ignoring it we get a lower bound on the log probability of the data given the model.

In our work, distribution Q is produced by a separate *recognition* model that has its own parameters, ϕ. These parameters are optimized at the same time as the parameters of the generative model, θ, to maximise the overall fit function $-\mathcal{F}(d;\theta,\phi) = -F(d;\theta,Q(\phi))$. Figure 2 shows graphically the nature of the approximation we are making and the relationship between our procedure and the EM algorithm. From equation 5, maximising $-F$ is equivalent to maximising the log probability of the data minus the Kullback-Leibler divergence, showing that this divergence acts like a penalty on the traditional log probability. The recognition model is thus encouraged to be a good approximation to the true posterior distribution P. However, the same penalty also encourages the generative model to change so that the true posterior distributions will be close to distributions that can be represented by the recognition model.

3 The Deterministic Helmholtz Machine

A *Helmholtz Machine* (figure 3) is a simple implementation of these principles. It is a connectionist system with multiple layers of neuron-like binary stochastic processing units connected hierarchically by two sets of weights. Top-down connections θ implement the generative model. Bottom-up connections ϕ implement the recognition model.

The key simplifying assumption is that the recognition distribution for a particular example d, $Q(\phi, d)$, is *factorial* (separable) in each layer. If there are h stochastic binary units in a layer ℓ, the portion of the distribution $P(\theta, d)$ due to that layer is determined by $2^h - 1$ probabilities. However, $Q(\phi, d)$ makes the assumption that the actual activity of any one unit in layer ℓ is independent of the activities of all the other units in that layer, given the activities of all the units in the lower layer, $\ell - 1$, so the recognition model needs only specify h probabilities rather than $2^h - 1$. The independence assumption allows $\mathcal{F}(d; \theta, \phi)$ to be evaluated efficiently, but this computational tractability is bought at a price, since the true posterior is unlikely to be factorial: the log probability of the data will be underestimated by an amount equal to the Kullback-Leibler divergence between the true posterior and the recognition distribution.

The generative model is taken to be factorial in the same way, although one should note that factorial generative models rarely have recognition distributions that are themselves exactly factorial.

Recognition for input example d entails using the bottom-up connections ϕ to determine the probability $q_j^\ell(\phi, d)$ that the jth unit in layer ℓ has activity $s_j^\ell = 1$. The recognition model is inherently stochastic – these probabilities are functions of the $0, 1$ activities $s_i^{\ell-1}$ of the units in layer $\ell - 1$. We use:

$$q_j^\ell(\phi, \mathbf{s}^{\ell-1}) = \sigma(\sum_i s_i^{\ell-1} \phi_{i\ ,j}^{\ell-1,\ell}) \tag{6}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the conventional sigmoid function, and $\mathbf{s}^{\ell-1}$ is the vector of activities of the units in layer $\ell - 1$. All units have recognition biases as one element of the sums, all the activities at layer ℓ are calculated after all the activities at layer $\ell - 1$, and s_i^1 are the activities of the input units. It is essential that there are no feedback connections in the recognition model.

In the terms of the previous section, α is a complete assignment of s_j^ℓ for all the units in all the layers other than the input layer (for which $\ell = 1$). The multiplicative contributions to the probability of choosing that assignment using the recognition weights are q_j^ℓ for units that are on and $1 - q_j^\ell$ for units that are off:

$$Q_\alpha(\phi, d) = \prod_{\ell > 1} \prod_j \left(q_j^\ell(\phi, \mathbf{s}^{\ell-1}) \right)^{s_j^\ell} \left(1 - q_j^\ell(\phi, \mathbf{s}^{\ell-1}) \right)^{1 - s_j^\ell} \tag{7}$$

The Helmholtz free energy $\mathcal{F}$ depends on the generative model through $E_\alpha(\theta, d)$ in equation 2. The top-down connections θ use the activities $\mathbf{s}^{\ell+1}$ of the units in layer $\ell + 1$ to determine the factorial generative probabilities $p_j^\ell(\theta, \mathbf{s}^{\ell+1})$ over the activities of the units in layer ℓ. The obvious rule to use is the sigmoid:

$$p_j^\ell(\theta, \mathbf{s}^{\ell+1}) = \sigma(\sum_i s_k^{\ell+1} \theta_{k\ ,j}^{\ell+1,\ell}). \tag{8}$$

including a generative bias (which is the only contribution to units in the topmost layer). Unfortunately this rule did not work well in practice for the sorts of inputs we tried. Appendix A discusses the more complicated method that we actually used to determine $p_j^\ell(\theta, \mathrm{s}^{\ell+1})$. Given this, the overall generative probability of α is:

$$p(\alpha|\theta) = \prod_{\ell>1} \prod_j \left(p_j^\ell(\theta, \mathrm{s}^{\ell+1})\right)^{s_j^\ell} \left(1 - p_j^\ell(\theta, \mathrm{s}^{\ell+1})\right)^{1-s_j^\ell} . \tag{9}$$

We extend the factorial assumption to the input layer $\ell = 1$. The activities s^2 in layer 2 determine the probabilities $p_j^1(\theta, \mathrm{s}^2)$ of the activities in the input layer. Thus

$$p(d|\alpha, \theta) = \prod_j \left(p_j^1(\theta, \mathrm{s}^2)\right)^{s_j^1} \left(1 - p_j^1(\theta, \mathrm{s}^2)\right)^{1-s_j^1} , \tag{10}$$

Combining equations 2, 9 and 10, and omitting dependencies for clarity,

$$\begin{aligned} E_\alpha(\theta, d) &= -\log p(\alpha|\theta) p(d|\alpha, \theta) & (11) \\ &= -\sum_{\ell \geq 1} \sum_j s_j^\ell \log p_j^\ell + \left(1 - s_j^\ell\right) \log \left(1 - p_j^\ell\right) & (12) \end{aligned}$$

Putting together the two components of $\mathcal{F}$, an unbiased estimate of the value of $\mathcal{F}(d; \theta, \phi)$ based on an explanation α drawn from Q_α is:

$$\begin{aligned} \mathcal{F}_\alpha(d; \theta, \phi) &= E_\alpha + logQ_\alpha & (13) \\ &= \sum_\ell \sum_j s_j^\ell \log \frac{q_j^\ell}{p_j^\ell} + \left(1 - s_j^\ell\right) \log \frac{1 - q_j^\ell}{1 - p_j^\ell} & (14) \end{aligned}$$

One could perform stochastic gradient ascent in the negative free energy across all the data $-\mathcal{F}(\theta, \phi) = -\sum_d \mathcal{F}(d; \theta, \phi)$ using equation 14 and a form of REINFORCE algorithm (Barto & Anandan, 1986; Williams, 1992; Dayan *et al*, in preparation). However, for the simulations in this paper, we made a number of mean-field inspired approximations, in that we replaced the stochastic binary activities s_j^ℓ by their mean values under the recognition model q_j^ℓ. We took:

$$q_j^\ell(\phi, \mathbf{q}^{\ell-1}) = \sigma(\sum_i q_i^{\ell-1} \phi_{i\ \ ,j}^{\ell-1,\ell}), \tag{15}$$

we made a similar approximation for p_j^ℓ which we discuss in appendix A, and we then averaged the expression in equation 14 over α to give the overall free energy:

$$-\mathcal{F}(\theta, \phi) = \sum_d \sum_\ell \sum_j \mathrm{KL}[q_j^\ell(\phi, \mathbf{q}^{\ell-1}), p_j^\ell(\theta, \mathbf{q}^{\ell+1})] \tag{16}$$

where the innermost term in the sum is the Kullback-Leibler divergence between generative and recognition distributions for unit j in layer ℓ for example d:

$$\mathrm{KL}[q,p] = q \log \frac{q}{p} + (1-q) \log \frac{1-q}{1-p}.$$

Weights θ and ϕ are trained by following the derivatives of $\mathcal{F}(\theta,\phi)$ in equation 16. Since the generative weights θ do not affect the actual activities of the units, there are no cycles, and so the derivatives can be calculated in closed form using the chain rule. Appendix B gives the appropriate recursive formulæ.

Note that this deterministic version introduces a further approximation by ignoring correlations arising from the fact that under the real recognition model, the actual activities at layer $\ell+1$ are a function of the actual activities at layer ℓ rather than their mean values.

Figure 4 demonstrates the performance of the Helmholtz Machine in a hierarchical learning task (Becker & Hinton, 1992), showing that it is capable of extracting the structure underlying a complicated generative model. The example shows clearly the difference between the generative (θ) and the recognition (ϕ) weights, since the latter often include negative side-lobes around their favoured shifts, which are needed to prevent incorrect recognition.

4 The Stochastic Helmholtz Machine

The derivatives required for learning in the deterministic Helmholtz machine are quite complicated because they have to take into account the effects that changes in an activity at one layer will have on activities in higher layers. However, by borrowing an idea from the Boltzmann machine (Hinton & Sejnowski, 1986; Ackley, Hinton & Sejnowski, 1985), we get a very simple learning scheme for layered networks of stochastic binary units that approximates the correct derivatives (Hinton *et al*, in preparation).

Learning in this scheme is separated into two phases. During the wake phase, data d from the world are presented at the lowest layer and binary activations of units at successively *higher* layers are picked according to the recognition probabilities, $q_j^\ell(\phi, \mathrm{s}^{\ell-1})$, determined by the bottom-up weights. The top-down generative weights from layer $\ell+1$ to layer ℓ are then altered to reduce the Kullback-Leibler divergence between the actual activations and the generative probabilities $p_j^\ell(\theta, \mathrm{s}^{\ell+1})$. In the sleep phase, the recognition weights are turned off and the top-down weights are used to activate the units. Starting at the top layer, activities are generated at successively *lower* layers based on the current top-down weights θ.

The network thus generates a random instance from its generative model. Since it has generated the instance, it knows the true underlying causes, and therefore has available the target values for the hidden units that are required to train the bottom-up weights. If the bottom-up and the top-down activation functions are both sigmoid (equations 6 and 8), then both phases use exactly the same learning rule, the purely local delta rule (Widrow & Stearns, 1985).

Unfortunately, there is no single cost function that is reduced by these two procedures. This is partly because the sleep phase trains the recognition model to invert the generative model for input vectors that are distributed according to the generative model rather than according to the real data and partly because the sleep phase learning does not follow the correct gradient. Nevertheless, $Q_\alpha = P_\alpha$ at the optimal end point, if it can be reached. Preliminary results by Brendan Frey (personal communication) show that this algorithm works well on some non-trivial tasks.

5 Discussion

The Helmholtz machine can be viewed as a hierarchical generalization of the type of learning procedure described by Zemel (1994) and Hinton and Zemel (1994). Instead of using a fixed independent prior distribution for each of the hidden units in a layer, the Helmholtz machine makes this prior more flexible by deriving it from the bottom-up activities of units in the layer above. In related work, Zemel and Hinton (1994) show that a system can learn a redundant population code in a layer of hidden units, provided the activities of the hidden units are represented by a point in a multidimensional constraint space with pre-specified dimensionality. The role of their constraint space is to capture statistical dependencies among the hidden unit activities and this can again be achieved in a more uniform way by using a second hidden layer in a hierarchical generative model of the type described here.

The old idea of analysis-by-synthesis assumes that the cortex contains a generative model of the world and that recognition involves inverting the generative model in real time. This has been attempted for non-probabilistic generative models (MacKay, 1956; Pece, 1992). However, for stochastic ones it typically involves Markov chain Monte Carlo methods (Neal, 1992). These can be computationally unattractive, and their requirement for repeated sampling renders them unlikely to be employed by the cortex. In addition to making learning tractable, its separate recognition model allows a Helmholtz machine to recognise without iterative sampling, and makes it much easier to see how generative models could be implemented in the cortex without running into serious time constraints. During recognition, the generative model is superfluous, since the recognition model contains all the information that is required. Nevertheless, the generative model plays an essential role in defining the objective function $\mathcal{F}$ that allows the parameters ϕ of the recognition model to be learned.

The Helmholtz machine is closely related to other schemes for self-supervised learning that use feedback as well as feedforward weights (Carpenter & Grossberg, 1987; Luttrell, 1992; 1994; Ullman, 1994; Kawato *et al*, 1993; Mumford, 1994). By contrast with Adaptive Resonance Theory (Carpenter & Grossberg, 1987) and the Counter-Streams model (Ullman, 1994), the Helmholtz machine treats self-supervised learning as a statistical problem — one of ascertaining a generative model which accurately captures the structure in the input examples. Luttrell (1992; 1994) discusses multilayer self-supervised learning aimed at faithful vector quantisation in the face of noise, rather than our aim of maximising the likelihood. The outputs of his separate low level coding networks are combined at higher levels, and thus their optimal coding choices become mutually dependent. These networks can be given a coding interpretation that is very similar to that of the Helmholtz machine. However, we are interested in distributed rather than local representations at each level (multiple cause rather than single cause models), forcing the approximations that we use. Kawato *et al* (1993) consider forward (generative) and inverse (recognition) models (Jordan & Rumelhart, 1992) in a similar fashion to the Helmholtz machine, but without this probabilistic perspective. The recognition weights between two layers do not just invert the generation weights between those layers, but also take into account the prior activities in the upper layer. The Helmholtz machine fits comfortably within the framework of Grenander's Pattern Theory (Grenander, 1976) in the form of Mumford's (1994) proposals for the mapping onto the brain.

As described, the recognition process in the Helmholtz machine is purely bottom-up — the top-down generative model plays no direct role and there is no interaction between units in a single layer. However, such effects are important in real perception and can be implemented using iterative recognition, in which the generative and recognition activations interact to produce the final activity of a unit. This can introduce substantial theoretical complications in ensuring that the activation process is stable and converges adequately quickly, and in determining how the weights should change so as to capture input examples more accurately. An interesting first step towards towards interaction within layers would be to organize their units into small clusters with local excitation and longer-range inhibition, as is seen in the columnar structure of the brain. Iteration would be confined within layers, easing the complications.

Acknowledgements

We are very grateful to Drew van Camp, Brendan Frey, Geoff Goodhill, Mike Jordan, David MacKay, Mike Revow, Virginia de Sa, Nici Schraudolph, Terry Sejnowski and Chris Williams for helpful discussions and comments, and particularly to Mike Jordan for extensive criticism of an earlier version of this paper. This work was supported by NSERC and IRIS. GEH is the Noranda Fellow of the Canadian Institute for Advanced Research. The current address for RSZ is Baker Hall 330, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA 15213

Appendices

A The Imaging Model

The sigmoid activation function given in equation 8 turned out not to work well for the generative model for the input examples we tried, such as the shifter problem (figure 1). Learning almost invariably got caught in one of a variety of local minima. In the context of a one layer generative model and without a recognition model, Saund (1994a;b) discussed why this might happen in terms of the underlying imaging model – which is responsible for turning binary activities in what we call layer 2 into probabilities of activation of the units in the input layer. He suggested using a noisy-or imaging model (Pearl, 1986), for which the weights $0 \leq \theta_{k\;\;,j}^{\ell+1,\ell} \leq 1$ are interpreted as probabilities that $s_j^\ell = 1$ if unit $s_k^{\ell+1} = 1$, and are combined as:

$$p_j^\ell(\theta, \mathbf{s}^{\ell+1}) = 1 - \prod_k \left(1 - s_k^{\ell+1}\theta_{k\;\;,j}^{\ell+1,\ell}\right). \tag{17}$$

The noisy-or imaging model worked somewhat better than the sigmoid model of equation 8, but it was still prone to fall into local minima. Dayan & Zemel (1994) suggested a yet more competitive rule based on the Integrated Segmentation and Recognition architecture of Keeler *et al* (1991). In this, the weights $0 \leq \theta_{k\;\;,j}^{\ell+1,\ell}$ are interpreted as the odds that $s_j^\ell = 1$ if unit $s_k^{\ell+1} = 1$, and are combined as:

$$p_j^\ell(\theta, \mathbf{s}^{\ell+1}) = 1 - \frac{1}{1 + \sum_k s_k^{\ell+1}\theta_{k\;\;,j}^{\ell+1,\ell}} \tag{18}$$

For the deterministic Helmholtz machine, we need a version of this activation rule that uses the probabilities $\mathbf{q}^{\ell+1}$ rather than the binary samples $\mathbf{s}^{\ell+1}$. This is somewhat complicated, since the obvious expression $1 - 1/(1 + \sum_k q_k^{\ell+1}\theta_{k\;\;,j}^{\ell+1,\ell})$ turns out not to work. In the end (Dayan & Zemel, 1994) we used a product of this term and the deterministic version of the noisy-or:

$$p_j^\ell(\theta, \mathbf{q}^{\ell+1}) = \left(1 - \frac{1}{1 + \sum_k q_k^{\ell+1}\theta_{k\;\;,j}^{\ell+1,\ell}}\right)\left(1 - \prod_k (1 - q_k^{\ell+1}\frac{\theta_{k\;\;,j}^{\ell+1,\ell}}{1 + \theta_{k\;\;,j}^{\ell+1,\ell}})\right) \tag{19}$$

Appendix B gives the derivatives of this. We used the exact expected value of equation 18 if there were only three units in layer $\ell+1$ because it is computationally inexpensive to work it out.

For convenience, we used the same imaging model (equations 18 and 19) for all the generative connections. In general one could use different types of connections between different levels.

B The Derivatives

Write $\mathcal{F}(d;\theta,\phi)$ for the contribution to the overall error in equation 16 for input example d, including the input layer:

$$\mathcal{F}(d;\theta,\phi)=\sum_{\ell}\sum_{j}-q_j^{\ell}\log p_j^{\ell}-\left(1-q_j^{\ell}\right)\log\left(1-p_j^{\ell}\right)+q_j^{\ell}\log q_j^{\ell}+\left(1-q_j^{\ell}\right)\log\left(1-q_j^{\ell}\right)$$

Then the total derivative for input example d with respect to the activation of a unit in layer ℓ is:

$$\begin{aligned}\frac{\partial\mathcal{F}(d;\theta,\phi)}{\partial q_j^{\ell}} &= \log\frac{1-p_j^{\ell}}{p_j^{\ell}}\frac{1-q_j^{\ell}}{q_j^{\ell}}+\sum_i\frac{p_i^{\ell-1}-q_i^{\ell-1}}{p_i^{\ell-1}\left(1-p_i^{\ell-1}\right)}\frac{\partial p_i^{\ell-1}}{\partial q_j^{\ell}}\\ &\quad+\sum_{\wp>\ell}\sum_k\frac{\partial\mathcal{F}(d;\theta,\phi)}{\partial q_k^{\wp}}\frac{\partial q_k^{\wp}}{\partial q_j^{\ell}}\end{aligned}\tag{20}$$

since changing q_j^{ℓ} affects the generative priors at layer $\ell-1$, and the recognition activities at all layers higher than ℓ. These derivatives can be calculated in a single backward propagation pass through the network, accumulating $\partial\mathcal{F}(d;\theta,\phi)/\partial q_k^{\wp}$ as it goes. The use of standard sigmoid units in the recognition direction makes $\partial q_k^{\wp}/\partial q_j^{\ell}$ completely conventional. Using equation 19 makes:

$$\begin{aligned}\frac{\partial p_i^{\ell-1}}{\partial q_j^{\ell}} &= \frac{\theta_{j,i}^{\ell,\ell-1}}{\left(1+\sum_a q_a^{\ell}\theta_{a,i}^{\ell,\ell-1}\right)^2}\left(1-\prod_a\left(1-q_a^{\ell}\frac{\theta_{a,i}^{\ell,\ell-1}}{1+\theta_{a,i}^{\ell,\ell-1}}\right)\right)+\\ &\quad\left(1-\frac{1}{1+\sum_a q_a^{\ell}\theta_{a,i}^{\ell,\ell-1}}\right)\frac{\theta_{j,i}^{\ell,\ell-1}}{1+\theta_{j,i}^{\ell,\ell-1}}\prod_{a\neq j}\left(1-q_a^{\ell}\frac{\theta_{a,i}^{\ell,\ell-1}}{1+\theta_{a,i}^{\ell,\ell-1}}\right)\end{aligned}\tag{21}$$

One also needs the derivative:

$$\begin{aligned}\frac{\partial p_i^{\ell-1}}{\partial \theta_{j,i}^{\ell,\ell-1}} &= \frac{q_j^{\ell}}{\left(1+\sum_a q_a^{\ell}\theta_{a,i}^{\ell,\ell-1}\right)^2}\left(1-\prod_a\left(1-q_a^{\ell}\frac{\theta_{a,i}^{\ell,\ell-1}}{1+\theta_{a,i}^{\ell,\ell-1}}\right)\right)+\\ &\quad\left(1-\frac{1}{1+\sum_a q_a^{\ell}\theta_{a,i}^{\ell,\ell-1}}\right)\frac{q_j^{\ell}}{\left(1+\theta_{j,i}^{\ell,\ell-1}\right)^2}\prod_{a\neq j}\left(1-q_a^{\ell}\frac{\theta_{a,i}^{\ell,\ell-1}}{1+\theta_{a,i}^{\ell,\ell-1}}\right)\end{aligned}\tag{22}$$

This is exactly what we used for the imaging model in equation 19. However, it is important to bear in mind that $p_j^{\ell}(\theta,\mathrm{s}^{\ell+1})$ should really be a function of the stochastic choices of the units in layer $\ell+1$. The contribution to the expected cost $\mathcal{F}$ is a function of $\left\langle\log p_j^{\ell}(\theta,\mathrm{s}^{\ell+1})\right\rangle$ and $\left\langle\log\left(1-p_j^{\ell}(\theta,\mathrm{s}^{\ell+1})\right)\right\rangle$, where $\langle\quad\rangle$ indicates averaging over the recognition distribution. These are not the same as $\log\left\langle p_j^{\ell}(\theta,\mathrm{s}^{\ell+1})\right\rangle$ and $\log\left(1-\left\langle p_j^{\ell}(\theta,\mathrm{s}^{\ell+1})\right\rangle\right)$, which is what the deterministic machine uses. For other imaging models, it is possible to take this into account.

References

[1] Ackley, DH, Hinton, GE & Sejnowski, TJ (1985). A learning algorithm for Boltzmann machines. *Cognitive Science,* **9**, 147-169.

[2] Barto, AG & Anandan, P (1985). Pattern recognizing stochastic learning automata. *IEEE Transactions on Systems, Man and Cybernetics,* **15**, 360-374.

[3] Becker, S & Hinton, GE (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature,* **355**, 161-163.

[4] Carpenter, G & Grossberg, S (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing,* **37**, 54-115.

[5] Dayan, P & Zemel, RS (1994). Competition and multiple cause models. *Neural Computation,* in press.

[6] Dempster, AP, Laird, NM & Rubin, DB (1976). Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society,* 1-38.

[7] Grenander, U (1976-1981). *Lectures in Pattern Theory I, II and III: Pattern Analysis, Pattern Synthesis and Regular Structures.* Berlin: Springer-Verlag., Berlin, 1976-1981).

[8] Hinton, GE & Sejnowski, TJ (1986). Learning and relearning in Boltzmann machines. In DE Rumelhart, JL McClelland and the PDP research group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations,* Cambridge, MA: MIT Press, 282-317.

[9] Hinton, GE & Zemel, RS (1994). Autoencoders, minimum description length and Helmholtz free energy. In JD Cowan, G Tesauro and J Alspector, editors, *Advances in Neural Information Processing Systems 6.* San Mateo, CA: Morgan Kaufmann, 3-10.

[10] Jordan, MI & Rumelhart, DE (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science,* **16**, 307-354.

[11] Kawato, M, Hayakama, H & Inui, T (1993). A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network,* **4**, 415-422.

[12] Kullback, S (1959). *Information Theory and Statistics.* New York: Wiley.

[13] Luttrell, SP (1992). Self-supervised adaptive networks. *IEE Proceedings Part F,* **139**, 371-377.

[14] Luttrell, SP (1994). A Bayesian analysis of self-organizing maps. *Neural Computation,* **6**, 767-794.

[15] MacKay, DM (1956). The epistemological problem for automata. In CE Shannon & J McCarthy, editors, *Automata Studies,* Princeton, NJ: Princeton University Press, 235-251.

[16] Mumford, D (1994). Neuronal architectures for pattern-theoretic problems. In C Koch and J Davis, editors, *Large-Scale Theories of the Cortex.* Cambridge, MA: MIT Press, 125-152.

[17] Neal, RM (1992). Connectionist learning of belief networks. *Artificial Intelligence,* **56**, 71-113.

[18] Neal, RM & Hinton, GE (1994). A new view of the EM algorithm that justifies incremental and other variants. Submitted to *Biometrika.*

[19] Pearl, J (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann.

[20] Pece, AEC (1992). Redundancy reduction of a Gabor representation: a possible computational role for feedback from primary visual cortex to lateral geniculate nucleus. In I Aleksander & J Taylor, editors, *Artificial Neural Networks, 2.* Amsterdam: Elsevier, 865-868.

[21] Saund, E (1994a). Unsupervised learning of mixtures of multiple causes in binary data. In JD Cowan, G Tesauro & J Alspector, editors, *Advances in Neural Information Processing Systems, 6.* San Mateo, CA: Morgan Kaufmann, 27-34.

[22] Saund, E (1994b). A multiple cause mixture model for unsupervised learning. *Neural Computation,* in press.

[23] Thompson, CJ (1988). *Classical Equilibrium Statistical Mechanics.* Oxford: Clarendon Press.

[24] Ullman, S (1994). Sequence seeking and counterstreams: A model for bidirectional information flow in the cortex. In C Koch and J Davis, editors, *Large-Scale Theories of the Cortex.* Cambridge, MA: MIT Press, 257-270.

[25] Widrow, B & Stearns, SD (1985). *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall.

[26] Williams, RJ (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning,* **8**, 229-56.

[27] Zemel, RS (1994). *A Minimum Description Length Framework for Unsupervised Learning.* PhD Dissertation, Computer Science, University of Toronto, Canada.

[28] Zemel, RS & Hinton, GE (1994). Learning population codes by minimizing description length. *Neural Computation,* in press.

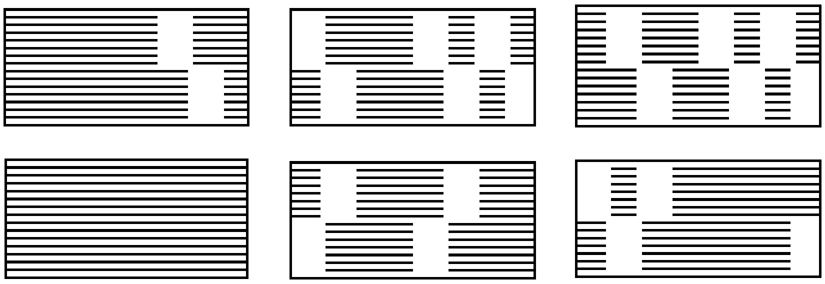

Figure 1: Shift Patterns. In each of these six patterns the bottom row of square pixels is a random binary vector, the top row is a copy shifted left or right by one pixel with wraparound, and the middle two rows are copies of the outer rows. The patterns were generated by a two-stage process. First the direction of the shift was chosen, with left and right being equiprobable. Then each pixel in the bottom row was turned on (white) with a probability of 0.2, and the corresponding shifted pixel in the top row and the copies of these in the middle rows were made to follow suit. If we treat the top two rows as a left retina and the bottom two rows as a right retina, detecting the direction of the shift resembles the task of extracting depth from simple stereo images of short vertical line segments. Copying the top and bottom rows introduces extra redundancy into the images which facilitates the search for the correct generative model.

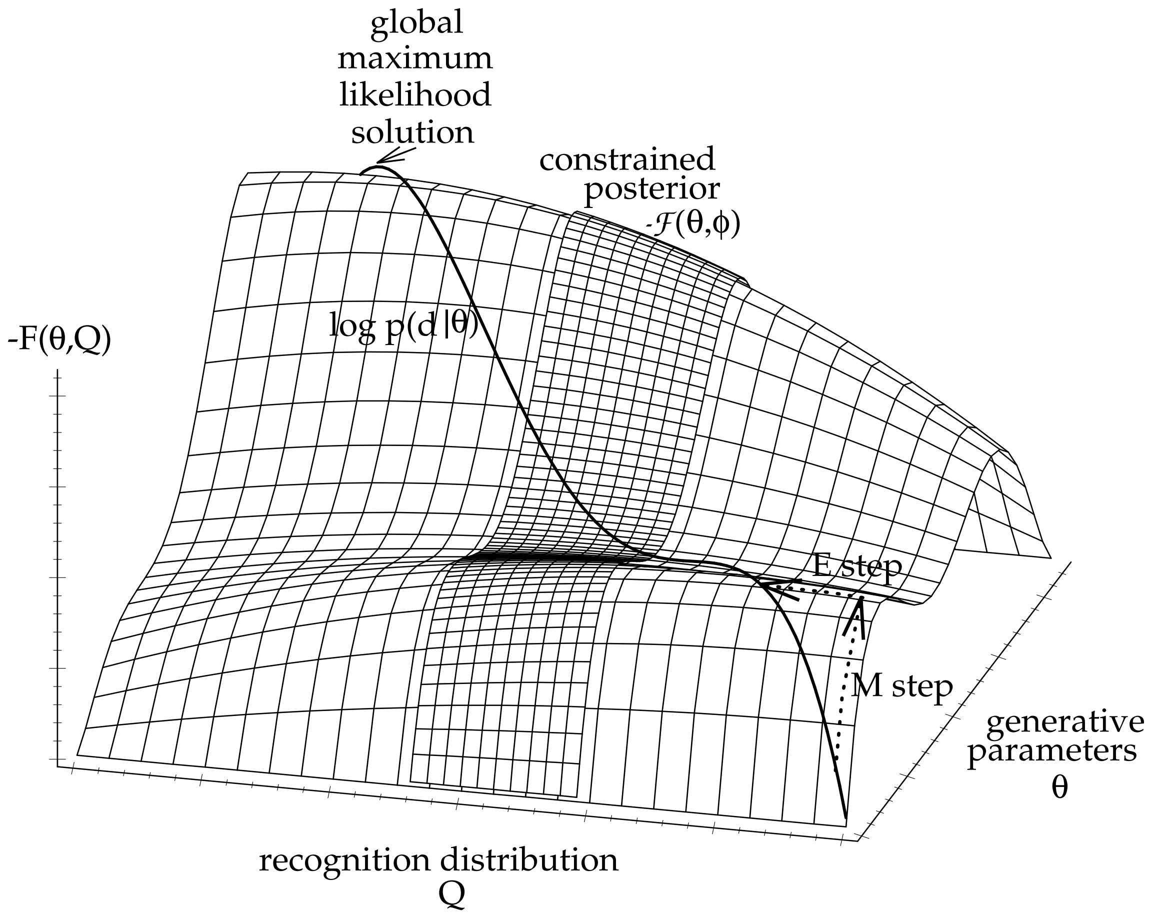

Figure 2: Graphical view of our approximation. The surface shows a simplified example of $-F(\theta, Q)$ as a function of the generative parameters θ and the recognition distribution Q. As discussed by Neal and Hinton (1994), the Expectation-Maximisation algorithm ascends this surface by optimising alternately with respect to θ (the M-step) and Q (the E-step). After each E-step, the point on the surface lies on the line defined by $Q_\alpha = P_\alpha$, and on this line, $-F = \log p(d|\theta)$. Using a factorial recognition distribution parameterised by ϕ restricts the surface over which the system optimises (labeled "constrained posterior"). We ascend the restricted surface using a conjugate gradient optimisation method. For a given θ, the difference between $\log p(d|\theta) = \max_Q \{-F(\theta, Q)\}$ and $-F(\theta, Q)$ is the Kullback-Leibler penalty in equation 5.

That EM gets stuck in a local maximum here is largely for graphical convenience, although neither it, nor our conjugate gradient procedure, is guaranteed to find their respective global optima. Showing the factorial recognition as a connected region is an arbitrary convention; the actual structure of the recognition distributions cannot be preserved in one dimension.

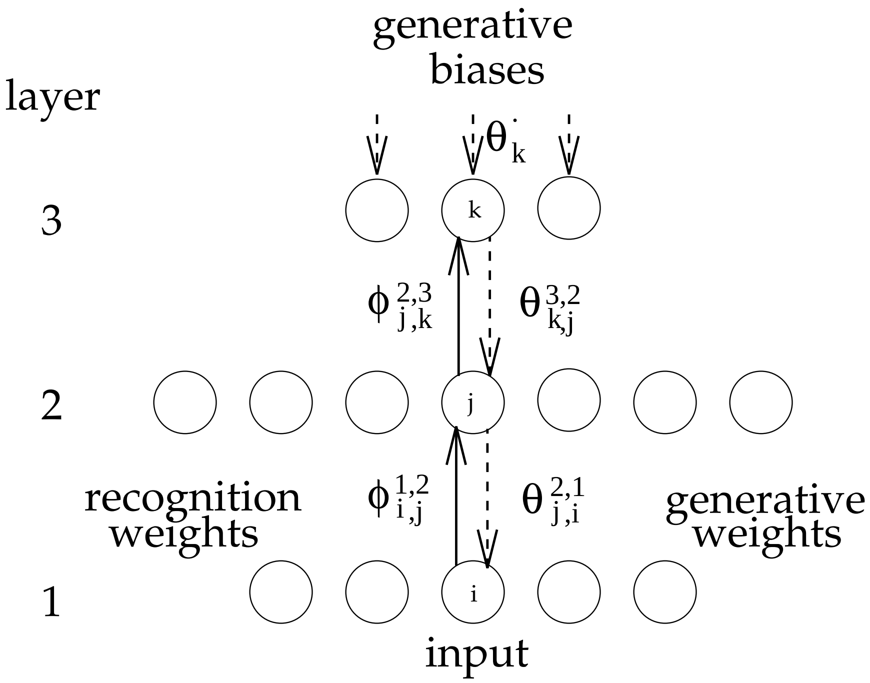

Figure 3: A Helmholtz Machine.

A simple three layer Helmholtz machine modelling the activity of 5 binary inputs (layer 1) using a two stage hierarchical model. Generative weights (θ) are shown as dashed lines, including the generative biases, the only such input to the units in the top layer. Recognition weights (ϕ) are shown with solid lines. Recognition and generative activation functions are described in the text.

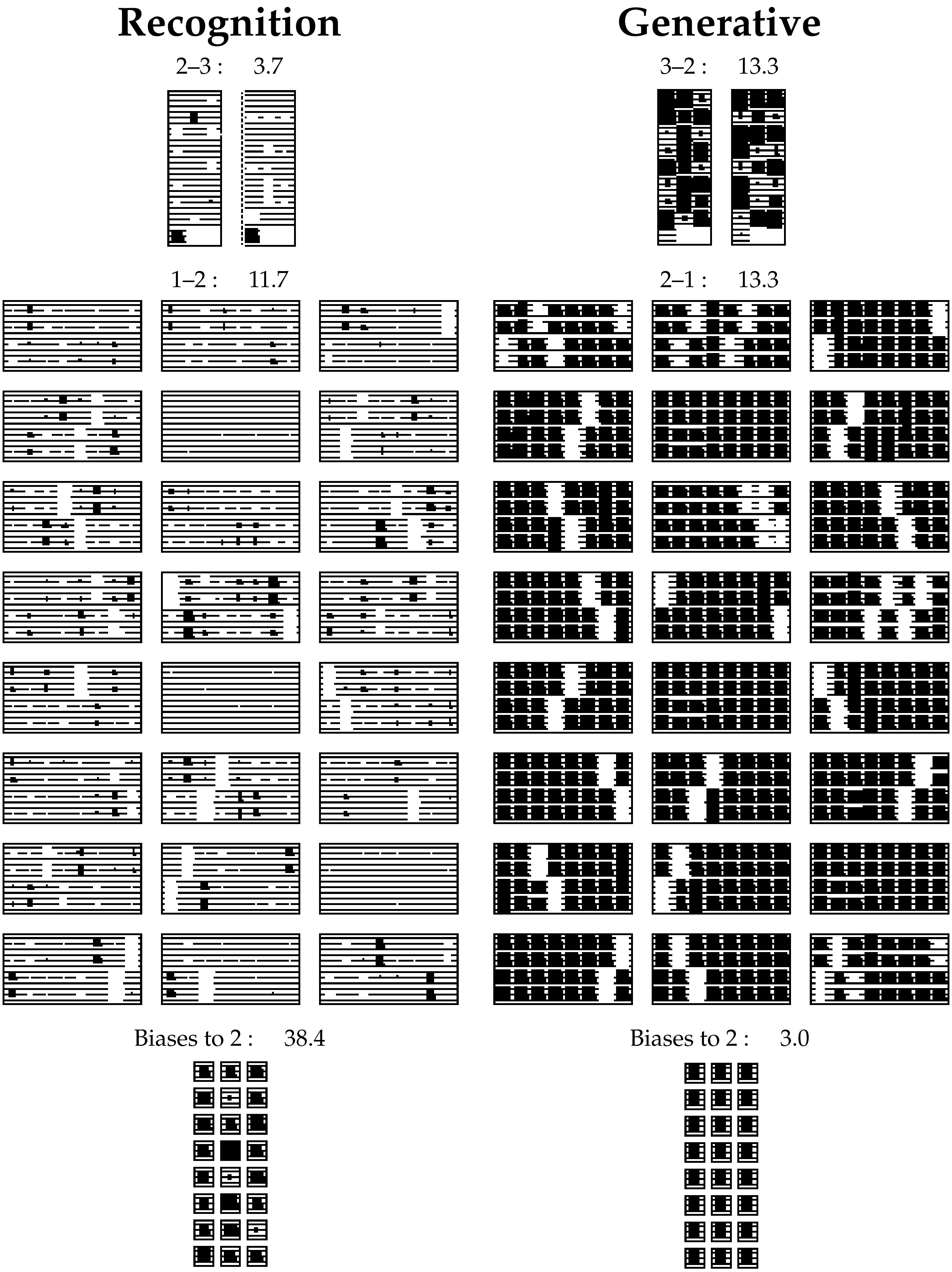

Figure 4: The Shifter – for legend see over

Legend to figure 4

Recognition and generative weights for a three layer Helmholtz machine's model for the shifter problem (see figure 1 for how the input patterns are generated). Each weight diagram shows recognition or generative weights between the given layers (1–2, 2–3, *etc*) and the number quoted is the magnitude of the largest weight in the array. White is positive, black negative, but the generative weights shown are the natural logarithms of the ones actually used. The lowest weights in the 2–3 block are the biases to layer 3; the biases to layer 2 are shown separately because of their different magnitude.

All the units in layer 2 are either silent, or respond to one or two pairs of appropriately shifted pairs of bits. The recognition weights have inhibitory side lobes to stop their units from responding incorrectly. The units in layer 3 are shift tuned, and respond to the units in layer 2 of their own shift direction. Note that under the imaging model (equation 18 or 19), a unit in layer 3 cannot specify that one in layer 2 should be off, forcing a solution that requires two units in layer 3. One aspect of the generative model is therefore not correctly captured.

Finding weights equivalent to those shown is hard, requiring many iterations of a conjugate gradient algorithm. To prevent the units in layers 2 and 3 from being permanently turned off early in the learning they were given fixed, but tiny generative biases ($\theta = 0.05$). Additional generative biases to layer 3 are shown in the figure; they learn the overall probability of left and right shifts.