Recognition in Hierarchical Models

Peter Dayan

Department of Brain and Cognitive Sciences
E25-210, MIT
Cambridge, MA 02139

Abstract. Various proposals have recently been made which cast cortical processing in terms of hierarchical statistical generative models (Mumford, 1994; Kawato, 1993; Hinton & Zemel, 1994; Zemel, 1994; Hinton *et al*, 1995; Dayan *et al*, 1995; Olshausen & Field, 1996; Rao & Ballard, 1995). In the case of vision, these claim that top-down connections in the cortical hierarchy capture essential aspects of how the activities of neurons in primary sensory areas are generated by the contents of visually observed scenes. The counterpart to a generative model is its statistical inverse, called a *recognition* model (Hinton & Zemel, 1994). This takes low-level activities and produces probability distributions over the entities in the world that could have led to them, expressed as activities of neurons in higher visual areas that model the image generation process. Even if a generative model is computationally tractable, its associated recognition model may not be. In this paper, we study various different types of exact, sampling-based and approximate recognition models in the light of computational and cortical constraints.

I Introduction

There are two popular notions as to the major on-line (as opposed to learning) mode of cortical processing. One concentrates on *discrimination* or *classification* of input from the sensory epithelium. For instance, if images contain a single handwritten digit, then the task for cortex in recognising or interpreting an image is to produce a probability distribution reporting which digit might be present. Invariances of various sorts are key – successive layers are taken as ignoring ever more information present in the image but irrelevant to the digit class, such as the style of the digit (*eg* italic or roman), the thickness of the strokes, the position on the page, *etc*. Purely bottom-up processing in the cortical hierarchy is typically thought of as implementing classification, justified by results such as Perrett *et al's* (1982) on the speed of face recognition, whose calculations imply that there little, if any, time for lateral or top-down influences. There is also substantial statistical theory underlying discrimination. However, its cortical instantiation is somewhat problematical. First, it is not clear how the sort of supervised training that underlies most classification systems could be arranged. Second, it is not clear how relevant prior information (such as that the particular writer for an image tends to favour particular curly strokes) can be properly incorporated into the recognition process.

The contending notion suggests that cortex builds a *model* (usually a probability density model) of the input it receives (Grenander, 1976; Mumford, 1994). The model captures the statistical structure of the observed input, according probabilities to particular inputs commensurate with their frequency in the world. In ideal circumstances, the model will reflect accurately the actual process by which images are created – *eg* for the images of handwritten digits, the model will include explicit choices for the identity of the digit, the style, the thickness of the strokes, *etc*. In cortex, these choices should be instantiated in the activities of particular amongst groups or populations of neurons. Recent statistical models for cortex have taken note of its layered structure (Felleman & Van Essen, 1991), and suggested that top-down and/or lateral connections contain the generative model. The model represents a (possibly complicated) probabilistic *prior* over observable scenes.

Given such a model, the most general task in interpreting a particular image is to blend information from the senses with this prior information (consistent with Bayes theorem) to report a posterior distribution over the various generative choices – *ie* analysis by synthesis. Alternatively, given some loss function, single values might be produced that summarise the posterior probability distribution. If one of the generative choices is the identity of the handwritten digit, then interpreting an image entails reporting the distribution over the digits it might contain. This is a characteristic inverse problem (Marroquin, 1985) – regularisation theory is an alternative way of describing the same operations (Poggio & Torre, 1984) – and is also the conventional way that maximum likelihood models (strictly maximum *a posteriori* models) are used for classification or discrimination. The generative mode for cortex therefore requires the discrimination mode too. This paper studies the discriminative phase, called *recognition* (Hinton & Zemel, 1994), that emerges as the Bayesian inverse to top-down generation.

If top-down and/or lateral weights in cortex are involved in the generative model, it is natural to conclude that the bottom-up weights are concerned with recognition. Of course, the other weights could also be involved – there are many cases for which top-down influences over perception are strong, and others such as binocular rivalry (Leopold & Logothetis, 1996; Logothetis *et al*, 1996) for which there appears to be an on-going interaction between bottom-up and top-down processing (Dayan, 1996). This paper studies different ways that recognition, or approximations to recognition, can be implemented for various sorts of generative model.

In some cases the recognition phase is computationally straightforward. Two important examples are when it is linear, which is the case of factor analysis (FA) discussed in detail in the next section, and when it involves a one-of-n operation, as in a mixture model such as the popular mixture of Gaussians (Nowlan, 1991) and mixture of experts (Jacobs *et al*, 1991) architectures. Even if exact recognition is tractable, we will see that there are different ways of implementing it, mixing combinations of bottom-up, top-down and lateral processing.

For many other generative models, even ones that are simple to specify, recognition is computationally challenging. Two examples are the unsupervised version of the Boltzmann machine (BM; Hinton & Sejnowski, 1986) and causal belief networks (*eg* Pearl, 1988). For the BM, both the generative and the recognition distributions are computationally intractable to calculate. For directed belief networks, their structure makes it easy to calculate the prior probability over a set of generative choices. However, calculating the posterior probability distribution given an observation is again difficult.

If exact recognition is intractable, one has two options. Markov chain Monte-Carlo methods such as Gibbs sampling can be used to collect samples that (at least asymptotically) reflect the exact recognition inverse (Neal, 1992;1993).[1] In cases like the BM or belief nets, Gibbs sampling specifies a Markov chain whose stationary distribution is the true posterior. Typically, one needs to run the chain for a while until transients are sure to have decayed, and then take samples of the states of the chain as being samples from the true recognition distribution. The disadvantage of using Gibbs sampling is the time it takes for transients to decay, and also the number of independent runs necessary if there are large energy barriers between states (or, equivalently, the high variance in the samples). Various methods for overcoming these problems have been suggested, in particular forms of annealing.

If Monte-Carlo sampling based on the true generative model is not to be used, then some form of approximation to the true recognition inverse is needed. Various different such schemes have been suggested, each with its own characteristics. The Helmholtz machine (HM; Hinton *et al*, 1995; Dayan *et al*, 1995) has a top-down belief-net generative model leading to a lowest layer, which represents the direct sensory report of images. The HM uses a bottom-up belief net to instantiate an approximation to the recognition inverse. In one version, the recognition distribution is described through samples that are generated stochastically (which is computationally easy) – and parameters of this bottom-up net are learnt during a training phase to make its samples appropriate. An alternative is to use mean-field methods (Saul *et al*, 1996; Jaakkola *et al*, 1996). Here, a parameterised form is chosen for the approximation to the whole inverse distribution for a particular image, where the particular parameterisation is chosen to make calculations easy. The parameters are then updated to minimise a Kullback-Leibler based measure of the difference between the approximate and the actual recognition distributions. A further alternative to the Helmholtz machine or mean field methods is to abandon the requirement of finding the true posterior distribution, and rather look for just its maximum.

This paper studies aspects of these different recognition choices. We are particularly interested in the relationship between the information contained in the bottom-up weights and that contained in the top-down or generative weights. Iterative recognition schemes that employ top-down weights turn out to require that the bottom-up weights are essentially the transpose of the generative weights (as is also true for principal components analysis). Recognition schemes that concentrate on feedforward processing require bottom-up weights that are

[1] The inverse is a whole distribution, and therefore can be specified by samples as in Monte-Carlo methods, or through a parameterisation, as in mean-field methods.

not purely the transpose of the top-down weights. In general, there remains an unresolved conflict between having fast and feedforward recognition, as inspired by Perrett *et al's* (1982) results, and having iterative recognition that blends bottom-up and top-down information in an appropriate way, including handling so-called explaining away effects in non-linear generative models in which either one generative cause for an input is active, or another one is active, but probably not both. The role of lateral processing is also unclear. There is neither the experimental evidence nor the computational compulsion to adopt one scheme in particular at present. Other issues are also important, particularly the way in which the efficacies of the connections are specified through experience, but they are not the current focus.

The next section studies in depth the factor analysis case of a two-layer and linear generative model with Gaussian noise, and comments on its multilayer extension, as in Chou, Willsky & Benveniste (1994) and Chou, Willsky & Nikoukhah (1994); and section 3 looks briefly at causal belief networks with binary units, which raise such problems as explaining away.

II Factor Analysis

A simple two-layer generative model is shown in figure 1 and is given by:

$$\mathbf{y} \sim \mathcal{N}[0, \Phi] \qquad \mathbf{x} \sim \mathcal{N}\left[\mathcal{G}^T\mathbf{y}, \Psi\right] \qquad \Psi = \operatorname{diag}\left(\tau_1^2, \ldots, \tau_n^2\right) \tag{1}$$

where $\mathcal{N}(\mathbf{a}, \Gamma)$ is a multivariate Gaussian distribution with mean $\mathbf{a}$ and covariance matrix Γ, $\mathbf{y} \in \Re^m$, $\mathbf{x} \in \Re^n$, and $\mathcal{G}$ is an $m \times n$ matrix of generative weights. Equations 1 imply that the components x_i of $\mathbf{x}$ are mutually independent, given the values of their belief net parents $\mathbf{y}$. Components y_i are called the *factors* underlying the observed examples $\mathbf{x}$.

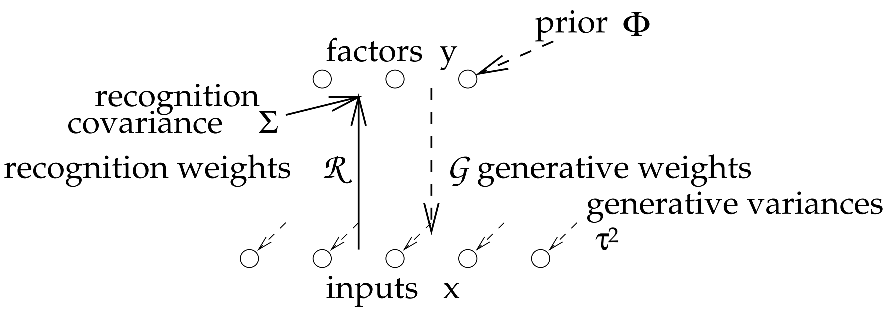

Fig. 1. Factor Analysis. The dotted lines on the right show the elements of the linear and Gaussian generative model in equations 1, involving factors $\mathbf{y}$ and observables $\mathbf{x}$. The solid lines on the left show the elements of the recognition model of equations 3 that is the inverse of the generative model.

This generative model is exactly that underlying the statistical technique of factor analysis (see Everitt 1984; Jolliffe, 1986 for introductions; Dempster, Laird & Rubin, 1977, and Rubin & Thayer, 1982 for more proximal analysis). It was pointed out as being a linear version of the Helmholtz machine by Neal (personal communication; Neal & Dayan, 1996) and was used for to model images of digits by Hinton *et al* (1996). Given this generative model, and a particular example $\mathbf{x}$, the role of recognition is to calculate the posterior distribution $\mathcal{P}[\mathbf{y}|\mathbf{x}]$ over the generators $\mathbf{y}$ given the observed image $\mathbf{x}$, or perhaps instead to calculate some particular value $\mathbf{y}^*$ that summarises this posterior distribution.

The joint distribution over $\mathbf{x}$ and $\mathbf{y}$ is Gaussian

$$\mathcal{P}[\mathbf{x},\mathbf{y}] \propto \exp\{-\frac{1}{2}\left[\mathbf{y}^T \Phi^{-1}\mathbf{y} + \left(\mathbf{x}-\mathcal{G}^T\mathbf{y}\right)^T \Psi^{-1}\left(\mathbf{x}-\mathcal{G}^T\mathbf{y}\right)\right]\}, \tag{2}$$

therefore the posterior distribution $\mathcal{P}[\mathbf{y}|\mathbf{x}]$ is also Gaussian $\mathcal{N}[\mathcal{R}^T\mathbf{x}, \Sigma]$, where

$$\mathcal{R} = \Psi^{-1}\mathcal{G}\left(\Phi^{-1} + \mathcal{G}\Psi^{-1}\mathcal{G}^T\right)^{-1} \qquad \Sigma^{-1} = \Phi^{-1} + \mathcal{G}\Psi^{-1}\mathcal{G}^T \tag{3}$$

The maximum *a posteriori* value $\mathbf{y}^{\text{MAP}}$ comes from minimising $-\log \mathcal{P}[\mathbf{x},\mathbf{y}]$:

$$\mathcal{E}[\mathbf{y}] = \mathbf{y}^T\Phi^{-1}\mathbf{y} + \left(\mathbf{x}-\mathcal{G}^T\mathbf{y}\right)^T\Psi^{-1}\left(\mathbf{x}-\mathcal{G}^T\mathbf{y}\right), \qquad ie \quad \mathbf{y}^{\text{MAP}} = \mathcal{R}^T\mathbf{x} \tag{4}$$

as it must be, since the posterior distribution is Gaussian and therefore unimodal. The maximum likelihood value $\mathbf{y}^{\text{ML}}$ comes from setting $\Phi^{-1} = 0$ in equation 4, which is dimensionally reasonable in the case in which there are fewer factors than input variables, *ie* $m < n$.

Top-down information for a particular case could be seen as changing the prior over $\mathbf{y}$ in equation 1 in two ways. If it changes the prior covariance matrix for the factors, then $\mathcal{R}$ must change too. If it just specifies a non-zero unconditional mean for the factors: $\mathbf{y} \sim \mathcal{N}[\bar{\mathbf{y}}, \Phi]$, then the new posterior is just

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left[\mathcal{R}^T\mathbf{x} + \left(\Phi^{-1} + \mathcal{G}\Psi^{-1}\mathcal{G}^T\right)^{-1}\Phi^{-1}\bar{\mathbf{y}}, \Sigma\right] \tag{5}$$

We are now in a position to describe some of the various proposals for recognition. For none of them is it quite clear how the posterior covariance matrix Σ might be represented in cortex (see Neal & Dayan, 1996). In the case that Φ is rotationally invariant, the factors are infamously rotationally underspecified. This allows us the cortically convenient option of taking Σ as being just a diagonal matrix. An additional problem with this linear factor analysis model is that it is only well determined (even up to rotation) if there are sufficiently fewer factors than input dimensions. This is not true of cortex, and is not necessary for non-linear factor analysis models (*eg* Olshausen & Field, 1996) or models that include temporal effects (*eg* Rao & Ballard, 1995).

Bottom-Up Method

The factor analysis version of the Helmholtz machine (Neal & Dayan, 1996; Hinton *et al*, 1996) is similar to the standard version of the Helmholtz machine in that it devotes a set of parameters to a feedforward belief net structure that is intended to approximate the recognition inverse using only bottom-up processing. This uses the feedforward weights $\mathcal{R}$ shown in figure 1, and an explicitly parameterised feedforward recognition covariance matrix Σ. This covariance matrix specifies the parameters of the noise corrupting the mean value $\mathcal{R}^T\mathbf{x}$, and so can be used to generate samples from the recognition model.

Note that recognition only requires a linear operation on the input pattern $\mathbf{x}$, but that the relationship between the top-down weights $\mathcal{G}$ and the bottom-up weights $\mathcal{R}$ is obscured by the priors Φ and Ψ. This is characteristic of methods that calculate the posterior distribution (or samples from it) based purely on bottom-up processing. The obscuring factor, $(\Phi^{-1} + \mathcal{G}\Psi^{-1}\mathcal{G}^T)^{-1}$ balances the prior variability of a factor (from Φ^{-1}) with the extent to which that factor might be responsible for inputs, modulated by the extent to which noise might be responsible instead (from $\mathcal{G}\Psi^{-1}\mathcal{G}^T$).

In principal components analysis (PCA), the value y_j would be the projection of the input $\mathbf{x}$ onto the jth orthonormal eigenvector of the covariance matrix of all the inputs. For PCA, the generative and recognition weight matrices are just transposes of each other ($\mathcal{R} = \mathcal{G}^T$), both containing the relevant eigenvectors. Most of the methods discussed below that employ top-down weights during processing also have $\mathcal{R} = \mathcal{G}^T$, but they do not contain the eigenvectors.

The advantage of the one-shot method in giving the mean of the posterior distribution in a single feedforward operation is offset by the disadvantage that it is not clear what to do if some particular input value x_k is not available on a particular case, for instance due to occlusion. $\mathcal{R}$ is tailored to the fact that all the inputs will be available. Also, $\mathcal{R}$ implicitly incorporates knowledge about the prior Φ over the factors, and so if top-down information can specify a different prior covariance matrix $\Phi' \neq \Phi$ on some particular occasion, then the bottom-up weights $\mathcal{R}$ will be incorrect. If top-down information only changes the unconditional mean, then the expression for the posterior mean in equation 5 shows that $\mathcal{R}$ is still appropriate.

Top-Down Method

More in the spirit of mean field methods, it is also possible to derive the posterior distribution of equations 3 in an iterative manner, using constitutively the top-down weights that define the generative model in the first place. A good way to understand this is through the same minimum description length (MDL; Rissanen, 1989) coding argument that motivates the Helmholtz machine. Consider using a distribution $\hat{\mathcal{N}} \equiv \mathcal{N}[\hat{\mathbf{y}}, \hat{\Sigma}]$ as a stochastic code for example $\mathbf{x}$. This means that a sample $\mathbf{y}^s$ is drawn from $\hat{\mathcal{N}}$, is coded itself using the prior $\mathcal{N}[\mathbf{0}, \Phi]$ over $\mathbf{y}$, and is used to provide a conditional prior $\mathcal{N}[\mathcal{G}^T\mathbf{y}^s, \Psi]$ over the actual image $\mathbf{x}$. The net mean description length for example $\mathbf{x}$ using this code has two additive components. The first is the cost of coding the sample $\mathbf{y}^s$ from $\hat{\mathcal{N}}$ (minus the bits back, Hinton & Zemel, 1994), and, on average, is:

$$\mathcal{F}_1 = KL\left\{\mathcal{N}\left[\hat{\mathbf{y}}, \hat{\Sigma}\right], \mathcal{N}\left[\mathbf{0}, \Phi\right]\right\} = \frac{1}{2}(\log\frac{|\Phi|}{|\hat{\Sigma}|} + \mathrm{tr}\left(\hat{\Sigma}\Phi^{-1}\right) + \hat{\mathbf{y}}^T\Phi^{-1}\hat{\mathbf{y}}) \qquad (6)$$

where $KL\{\mathcal{P}, \mathcal{Q}\}$ is the Kullback-Leibler divergence from $\mathcal{P}$ to $\mathcal{Q}$. The second component is the cost of coding the image $\mathbf{x}$ given $\mathbf{y}^s$, which, on average, is:

$$\mathcal{F}_2 = \frac{1}{2}((\mathbf{x} - \mathcal{G}^T\hat{\mathbf{y}})^T \Psi^{-1}\left(\mathbf{x} - \mathcal{G}^T\hat{\mathbf{y}}\right) + \mathrm{tr}\left(\mathcal{G}\Psi^{-1}\mathcal{G}^T\hat{\Sigma}\right) + \textstyle\sum_j \log\tau_j^2) + \mathcal{K} \qquad (7)$$

where $\mathcal{K}$ is a constant that does not depend on $\hat{\mathbf{y}}$ or $\hat{\Sigma}$.

Shannon's theorem guarantees that the description length $\mathcal{F} \equiv \mathcal{F}_1 + \mathcal{F}_2$ for any choice of $\hat{\mathbf{y}}$ and $\hat{\Sigma}$ is greater than or equal to $-\log\mathcal{P}[\mathbf{x}]$ under the generative model, and equality holds when the coding distribution $\hat{\mathcal{N}}$ is the true recognition distribution (*ie* the true probabilistic inverse to the generative distribution). Consider, therefore, minimising $\mathcal{F}$ with respect to $\hat{\mathbf{y}}$ and $\hat{\Sigma}$. The linear and Gaussian nature of the generative model makes the two minimisations independent. As might be expected from equation 3, the optimisation of $\hat{\Sigma}$ is also independent of the input $\mathbf{x}$ and therefore can be done once and for all. Optimising $\hat{\mathbf{y}}$ is more interesting. The parts of $\mathcal{F}$ that depend on $\hat{\mathbf{y}}$ comprise a quadratic form, with $\nabla_{\hat{\mathbf{y}}}\mathcal{F} = \Phi^{-1}\hat{\mathbf{y}} - \mathcal{G}\Psi^{-1}\left(\mathbf{x} - \mathcal{G}^T\hat{\mathbf{y}}\right)$. We can therefore either read off the optimal $\hat{\mathbf{y}}$ as implied by equation 3 (by solving for $\nabla_{\hat{\mathbf{y}}}\mathcal{F} = \mathbf{0}$), or implement gradient descent in $\mathcal{F}$ using the dynamical system (Olshausen & Field, 1996; Rao & Ballard, 1995):

$$\tau\frac{d\hat{\mathbf{y}}}{dt} = -\nabla_{\hat{\mathbf{y}}}\mathcal{F} = -\Phi^{-1}\hat{\mathbf{y}} + \mathcal{G}\Psi^{-1}\left(\mathbf{x} - \mathcal{G}^T\hat{\mathbf{y}}\right) \qquad (8)$$

where τ is a time constant. The interesting aspect of this equation is its implications for the bottom-up weights and processing in the $\mathbf{x}$ layer. Equation 8 suggests calculating the prediction error $\left(\mathbf{x} - \mathcal{G}^T\hat{\mathbf{y}}\right)$ in the $\mathbf{x}$ layer (this is the difference between the actual image $\mathbf{x}$ and the image that would be predicted from the mean top-down activities $\hat{\mathbf{y}}$), down-weighting this prediction error in the $\mathbf{x}$ layer by the noise magnitudes $1/\tau_j^2$ along each $\mathbf{x}$ dimension, and then propagating it through the transpose of the generative weights $\mathcal{G}$ to change $\hat{\mathbf{y}}$. Note the two major differences from the one-shot approach: a) the system is iterative, based on calculating the prediction errors; and b), as in PCA, the bottom-up weights are the transpose of the generative weights, rather than being dependent also on Ψ and Φ. The generative weights $\mathcal{G}$ that minimise $\mathcal{F}$ will nevertheless in general not be the same as the ones calculated by PCA, *ie* they will differ from the eigenvectors of the covariance matrix of the images.

If there is top-down information that changes the unconditional mean of $\mathbf{y}$, then this just adds an extra term $\Phi^{-1}\bar{\mathbf{y}}$ to the update equations. Changing the unconditional covariance matrix Φ requires just a change to the update within the $\mathbf{y}$ layer, and not a change to the bottom-up weights. Further, if the value of some input dimension is not specified on some particular occasion, then it should make no contribution in the term $\mathcal{G}\Psi^{-1}\left(\mathbf{x}-\mathcal{G}^T\hat{\mathbf{y}}\right)$. This will be the case if the result of top-down influences *on the* $\mathbf{x}$ *layer* is to set the relevant component of $\mathbf{x}$ equal to its top-down mean, in the absence of any information from the scene. Top-down inference therefore avoids all the problems alluded to for bottom-up inference, at the expense of requiring iterations to satisfy $d\hat{\mathbf{y}}/dt = 0$.

Olshausen & Field (1996) developed a dynamical system like that of equation 8 from the starting point of minimising the cost:

$$\mathcal{C}(\hat{\mathbf{y}},\mathcal{G}) = \sum_i f\left(\hat{y}_i\right) + \left(\mathbf{x}-\mathcal{G}^T\hat{\mathbf{y}}\right)^T \Psi^{-1}\left(\mathbf{x}-\mathcal{G}^T\hat{\mathbf{y}}\right) \tag{9}$$

which is closely related to $\mathcal{F}$.[2] In equation 9, $\hat{\mathbf{y}}$ is again the cortical representation of $\mathbf{x}$, and is also chosen to balance two costs. The first term is intended to encourage sparseness in the $\hat{\mathbf{y}}$, using a penalty term $f(y)$ such as $f(y) = \log\left(1+y^2\right)$ which encourages $\mathbf{y}$ units to be silent. Just like component $\mathcal{F}_1$ in equation 6, this penalty term is essentially equivalent to that coming from a prior $\mathcal{P}[y] \propto e^{-f(y)}$ for the activities in the $\mathbf{y}$ layer, in which they are mutually independent. The term $-\Phi^{-1}\hat{\mathbf{y}}$ in equation 8 is replaced in $\nabla_{\hat{\mathbf{y}}}\mathcal{C}(\hat{\mathbf{y}},\mathcal{G})$ by a vector whose components are $-f'\left(\hat{y}_i\right)$. However, making $f(y)$ non-quadratic means that there is no longer a separation in the minimisations with respect to $\hat{\mathbf{y}}$ and $\hat{\Sigma}$ (just as certainty equivalence in control theory only holds in the linear case), and so minimising $\mathcal{C}$ with respect to $\hat{\mathbf{y}}$ becomes itself an approximation. One can write down the equivalent of $\mathcal{F}$, but such a simple dynamical system can only find (local) maximum *a posteriori* values and not the true Bayesian conditional mean.

Just like components $\mathcal{F}_2$ in equation 7, the second term in equation 9 encourages $\hat{\mathbf{y}}$ to provide a good model for the image $\mathbf{x}$, through the medium of the generative weights $\mathcal{G}$. Olshausen & Field (1996) showed that realistic generative receptive fields emerge for the $\mathbf{y}$ units when recognition is based on choosing $\hat{\mathbf{y}}^* = \mathrm{argmin}_{\hat{\mathbf{y}}}\mathcal{C}(\hat{\mathbf{y}},\mathcal{G})$, and the generative weights are altered based on these values. Reasonable recognition receptive fields for the $\mathbf{y}$ units are also observed, but they depend on the images that are presented and require calculating $\hat{\mathbf{y}}^*$.

If $f(y)$ is quadratic, as in Rao & Ballard (1995), and effectively also in $\mathcal{F}_1$, then there is actually no incentive for $\hat{\mathbf{y}}^*$ to be sparse. For instance, if there were two units y_1 and y_2 with equal generative weights, then $\hat{y}_1^2+\hat{y}_2^2+(9-[\hat{y}_1+\hat{y}_2])^2$ is minimised with $\hat{y}_1^* = \hat{y}_2^* = 3$; whereas $\log(1+\hat{y}_1^2)+\log(1+\hat{y}_2^2)+(9-[\hat{y}_1+\hat{y}_2])^2$ is minimised at $(\hat{y}_1^*,\hat{y}_2^*) = (0.1, 8.8)$ or $(8.8, 0.1)$.

Lateral Method

Olshausen (personal communication) has pointed out an equivalent form of equation 8, for minimising $\mathcal{F}$ with respect to $\hat{\mathbf{y}}$:

$$\tau\frac{d\hat{\mathbf{y}}}{dt} = \mathcal{G}\Psi^{-1}\mathbf{x} - \left(\Phi^{-1}+\mathcal{G}\Psi^{-1}\mathcal{G}^T\right)\hat{\mathbf{y}}. \tag{10}$$

[2] Olshausen & Field (1996) used $\Psi = \mathcal{I}$, based on the reasonable assumption that all errors in the $\mathbf{x}$ layer are equivalent.

This suggests a slightly different calculation scheme from equation 8, in which the image $(\mathbf{x})$, rather than the prediction error for the image $\left(\mathbf{x} - \mathcal{G}^T \hat{\mathbf{y}}\right)$ is down-weighted by Ψ^{-1} and propagated through a weight matrix $\mathcal{G}$ which again is just the transpose of the generative weights. Now, though, computation in the $\mathbf{y}$ layer is more complicated, including requiring connections that are sensitive to the noise magnitudes Ψ^{-1} in the $\mathbf{x}$ layer. This shares the disadvantage of the bottom-up scheme in terms of requiring weights that include information about the priors Φ^{-1} and $\mathcal{G}\Psi^{-1}\mathcal{G}^T$, but does at least allow the non-specification of some particular input, provided, as also for equation 8 that the input is set at exactly its top-down predicted value at all times.

Combined Bottom-Up and Top-Down Method

The final method is based on multiplying the update in equation 10 by the positive definite matrix $\Sigma = (\Phi^{-1} + \mathcal{G}\Psi^{-1}\mathcal{G}^T)^{-1}$. In this case, one can also implement the dynamics:

$$\tau \frac{d\hat{\mathbf{y}}}{dt} = \left(\Phi^{-1} + \mathcal{G}\Psi^{-1}\mathcal{G}^T\right)^{-1} \mathcal{G}\Psi^{-1}\mathbf{x} - \hat{\mathbf{y}} = \mathcal{R}^T\mathbf{x} - \hat{\mathbf{y}} \tag{11}$$

This version has the attractive characteristic that if all the inputs are specified, then the feedforward information is instantly correct, and so iteration would in theory not be required (if the time constant $\tau = 1$). If some inputs are not specified, then, by the same reasoning as above, the system will still find the correct conditional mean for the factors.

In this simple linear and Gaussian case, there is therefore an update form that is based on: a) the one-shot method used in the conventional Helmholtz machine, and b) the iterative scheme employed in Rao & Ballard (1995) and also in mean field methods for the non-linear case (Jaakkola *et al*, 1996; Olshausen & Field, 1996).

Hierarchical Factor Analysis

Figure 2 shows a slightly more complicated, but still linear and Gaussian model which has three layers, two of which are spatially separated. This model, and yet more complicated versions of it, is due to Chou and Willsky and their colleagues (Chou, Willsky & Benveniste, 1994; Chou, Willsky & Nikoukhah, 1994; Luettgen & Willsky, 1995). These authors were interested in using Kalman filters in scale rather than in time to build tractable models of inputs that naturally live in multiple dimensions (such as images) rather than one dimension (such as auditory waveforms). The most natural generative model in two-dimensions is a Markov random field (MRF; Kinderman & Snell, 1980), but inference in MRFs is notoriously intractable. Chou *et al* (1994) develop a sophisticated theory of recognition in such structures, and we will mostly just cite the relevant results in our own terms, without writing out the more notionally gruesome ones.

One-Pass Method

The net prior distribution for $\mathbf{y}^a$ is Gaussian, with mean $\mathbf{0}$ and covariance matrix $\Xi^a + \mathcal{H}^{a^T} \Phi \mathcal{H}^a$. One can therefore use the results of the previous section to work out the conditional mean and variance of the Gaussian distributions $\mathbf{y}^a|\mathbf{x}^a$, and equivalently $\mathbf{y}^b|\mathbf{x}^b$. Only the means depend on the images $\mathbf{x}^a$ and $\mathbf{x}^b$. It turns out that one can also combine the means of these conditional distributions in a linear manner to work out the mean of the Gaussian distribution $\mathbf{z}|\mathbf{x}^a, \mathbf{x}^b$, including information from both halves of the input. These linear feedforward operations are closely related to those in equation 3, only with added complexity because of the separated inputs.

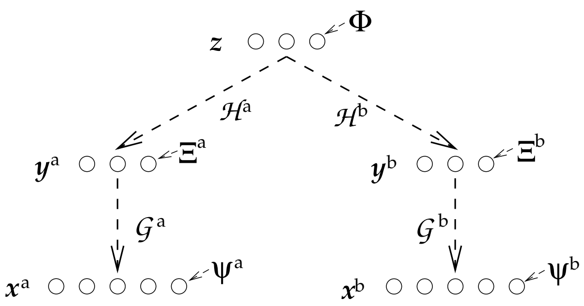

Fig. 2. Hierarchical factor analysis, after Chou *et al* (1994). The three-layer linear Gaussian generative model for inputs $\mathbf{x}^a$ and $\mathbf{x}^b$ via factors $\mathbf{z}$ and $\mathbf{y}^a$ and $\mathbf{y}^b$. Φ is the covariance of $\mathbf{z}$, Ξ^a is the covariance of the noise corrupting $\mathbf{y}^a$ from $\mathcal{H}^{a^T}\mathbf{z}$, and Ψ^a the noise corrupting $\mathbf{x}^a$ from $\mathcal{G}^{a^T}\mathbf{y}^a$. Only the generative weights are shown.

Although one can calculate $\mathbf{z}|\mathbf{x}^a, \mathbf{x}^b$ using purely bottom-up calculations, and $\mathbf{y}^a$ is conditionally independent of $\mathbf{y}^b$ and therefore $\mathbf{x}^b$ *given* $\mathbf{z}$,, the structure of the generative model makes it clear that $\mathbf{y}^a$ is conditionally *dependent* on $\mathbf{x}^b$ in the recognition circumstance in which we are given only $\mathbf{x}^a$ and $\mathbf{x}^b$ and not $\mathbf{z}$. In fact, $\mathbf{y}^a, \mathbf{y}^b$ and $\mathbf{z}$ are joinly Gaussian, with a hyper-elliptical covariance structure. The Kalman filter framework can be used for smoothing as well as filtering, in this case, feeding information back from $\mathbf{z}|\mathbf{x}^a, \mathbf{x}^b$ to update the distribution $\mathbf{y}^a|\mathbf{x}^a$ to $\mathbf{y}^a|\mathbf{x}^a, \mathbf{x}^b$. Chou *et al* (1994) show how to do this efficiently using a single top-down pass, in a generalization of the Rauch-Tung-Striebel smoothing algorithm to this tree-like case. Chou *et al* (1994) also point out a slightly different variant on this two pass algorithm in which the bottom-up phase calculates terms such as $\mathbf{y}^{a^{ML}}$ ignoring the prior, and then the top-down phase applies all the information about the priors.

Iterative Methods

Although, just as for the case of the two-layer model, there is a substantially efficient non-iterative algorithm, Rao & Ballard (1995) pointed out that one can calculate the means of the distributions $\mathbf{y}^a|\mathbf{x}^a, \mathbf{x}^b$, $\mathbf{y}^b|\mathbf{x}^a, \mathbf{x}^b$ and $\mathbf{z}|\mathbf{x}^a, \mathbf{x}^b$ using a dynamical system closely related to those in equations 8 and 10. The quadratic form comprising the terms in the description length that depend on the means (written as $\hat{\mathbf{y}}^a$, *etc*) is given by

$$\begin{aligned} 2\mathcal{F}' = {} & \hat{\mathbf{z}}^T \Phi^{-1} \hat{\mathbf{z}} + \\ & \left(\hat{\mathbf{y}}^a - \mathcal{H}^{a^T}\hat{\mathbf{z}}\right)^T \Xi^{a^{-1}} \left(\hat{\mathbf{y}}^a - \mathcal{H}^{a^T}\hat{\mathbf{z}}\right) + \left(\hat{\mathbf{y}}^b - \mathcal{H}^{b^T}\hat{\mathbf{z}}\right)^T \Xi^{b^{-1}} \left(\hat{\mathbf{y}}^b - \mathcal{H}^{b^T}\hat{\mathbf{z}}\right) + \\ & \left(\mathbf{x}^a - \mathcal{G}^{a^T}\hat{\mathbf{y}}^a\right)^T \Psi^{a^{-1}} \left(\mathbf{x}^a - \mathcal{G}^{a^T}\hat{\mathbf{y}}^a\right) + \left(\mathbf{x}^b - \mathcal{G}^{b^T}\hat{\mathbf{y}}^b\right)^T \Psi^{b^{-1}} \left(\mathbf{x}^b - \mathcal{G}^{b^T}\hat{\mathbf{y}}^b\right). \end{aligned}$$

Using gradient descent to solve for $\nabla\mathcal{F}' = \mathbf{0}$ leads to an analogue of equation 8 (Rao & Ballard, 1995):

$$\begin{aligned} \tau\frac{d\hat{\mathbf{z}}}{dt} &= -\Phi^{-1}\hat{\mathbf{z}} + \mathcal{H}^a \Xi^{a^{-1}} \left(\hat{\mathbf{y}}^a - \mathcal{H}^{a^T}\hat{\mathbf{z}}\right) + \mathcal{H}^b \Xi^{b^{-1}} \left(\hat{\mathbf{y}}^b - \mathcal{H}^{b^T}\hat{\mathbf{z}}\right) \\ \tau\frac{d\hat{\mathbf{y}}^a}{dt} &= -\Xi^{a^{-1}} \left(\hat{\mathbf{y}}^a - \mathcal{H}^{a^T}\hat{\mathbf{z}}\right) + \mathcal{G}^a \Psi^{a^{-1}} \left(\mathbf{x}^a - \mathcal{G}^{a^T}\hat{\mathbf{y}}^a\right) \end{aligned} \tag{12}$$

These share with equation 8 the characteristic of how top-down prediction errors at the various levels ($\hat{\mathbf{y}}^a - \mathcal{H}^{a^T}\hat{\mathbf{z}}$ and $\mathbf{x}^a - \mathcal{G}^{a^T}\hat{\mathbf{y}}^a$) are downweighted by the noise covariances and propagated bottom-up through the transpose of the generative weight matrices. Since the overall joint distribution $\mathbf{z}, \mathbf{y}^a, \mathbf{y}^b|\mathbf{x}^a, \mathbf{x}^b$ is elliptical, the use of these means is slightly tricky. Again, in a non-linear or non-Gaussian case, such as a multilayer analogue of Olshausen & Field's (1996) sparsity prior, the separation between means and covariances would disappear, and it would no longer be possible to use these equations to work out the true posterior means.

There is also an analogue of equation 10:

$$\begin{aligned} \tau\frac{d\hat{\mathbf{z}}}{dt} &= \mathcal{H}^a \Xi^{a^{-1}} \hat{\mathbf{y}}^a + \mathcal{H}^b \Xi^{b^{-1}} \hat{\mathbf{y}}^b - \left(\Phi^{-1} + \mathcal{H}^a \Xi^{a^{-1}} \mathcal{H}^{a^T} + \mathcal{H}^b \Xi^{b^{-1}} \mathcal{H}^{b^T}\right)\hat{\mathbf{z}} \\ \tau\frac{d\hat{\mathbf{y}}^a}{dt} &= \mathcal{G}^a \Psi^{a^{-1}} \mathbf{x}^a + \Xi^{a^{-1}} \mathcal{H}^{a^T} \hat{\mathbf{z}} - \left(\Xi^{a^{-1}} + \mathcal{G}^a \Psi^{a^{-1}} \mathcal{G}^{a^T}\right)\hat{\mathbf{y}}^a \end{aligned} \tag{13}$$

which implies the use of lateral operations. The equivalent of equation 11 is more complicated, however, because the bottom up estimate of $\hat{\mathbf{y}}^a$ given just $\mathbf{x}^a$ is different from the final estimate of $\hat{\mathbf{y}}^a$ given both $\mathbf{x}^a$ and $\mathbf{x}^b$. Unlike equation 10, it is not clear how to specify a single set of bottom-up weights that can conveniently be used for both one-pass and iterative inference.

III Non-linear Models

The previous section considered the case of linear and Gaussian generative models for which there are computationally tractable ways of calculating the exact recognition inverse distributions. Even though there may be purely bottom-up ways of doing this in regular cases, for which there is no occlusion and no top-down information relevant to inference for a particular image, iterative methods can also be used, and have certain demonstrable advantages. However, linear and Gaussian models are unlikely to suffice, even for the most primitive datasets. One class of non-linear models that has been studied in some depth is that of binary belief networks with sigmoidal activation functions (Neal, 1992; Hinton *et al*, 1995; Saul *et al*, 1996). These preserve the top-down structure of the generative model, but, for a single unit x_1, they have (*cf* equation 1):

$$x_1 \sim \mathcal{B}\left[\sigma\left(\left[\mathcal{G}^T \mathbf{y}\right]_1\right)\right] \tag{14}$$

where $\mathcal{B}[p]$ is the binomial distribution with mean p and σ is a sigmoid function whose output lies between 0 and 1. The activities in layer $\mathbf{y}$ are set similarly, except, for a two-layer network, that the only input to the binomial distribution is a bias term. An extra component of $\mathbf{y}$ is treated as a bias for determining $\mathbf{x}$. A three layer network is shown in figure 3a.

Examples of layered belief networks with this structure are given by Hinton *et al* (1995), in particular the comparatively large networks (with $4-16-16-64$ units in three hidden layers and one input layer) that model 8×8 binary images of handwritten digits. A key aspect of these generative models is that the activities of units *within* a layer are mutually independent, given the activities of the units in the layer above. This makes specifying the generative probabilities very simple. In such cases, it is again necessary to calculate the recognition inverse to this generative model – now the recognition distribution assigns probabilities to the 2^n binary states of the n hidden units in the network, given the activities in the lowest input layer. For general networks, this distribution is not tractably computable, and therefore sampling or approximations are necessary. Even though in the generative model, activities of units within a layer are independent given the activities in the layer above, activities of units within a layer in the recognition model need not be mutually independent given the activities in the layer below.

The non-linear belief net model makes for much richer generative and recognition models than the linear Gaussian models of factor analysis. Two very simple but revealing generative models are shown in figure 3b;c. The left example shows a case of explaining away (Pearl, 1988). All the units are most likely to be off (0). However, the occurrence of x being on (1) requires explanation by either $y_1 = 1$ or $y_2 = 1$. The *a priori* unlikelihood that the $\mathbf{y}$ units are active makes it unlikely that $y_1 = y_2 = 1$; having $y_1 = 1$ *explains away* $x = 1$ and so obviates the need for $y_2 = 1$ as well. Figure 3c is a modification of this example in which y_1 tends to generate $x_1 = x_2 = 1$.

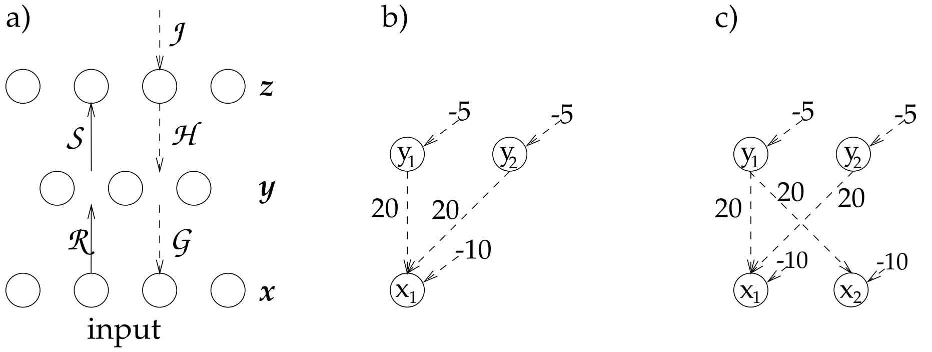

Fig. 3. a) Three layer sigmoidal belief net. The layered structure shows the conditional independencies in the net $\mathcal{P}[\mathbf{x}, \mathbf{y}, \mathbf{z}] = \mathcal{P}[\mathbf{z}]\mathcal{P}[\mathbf{y}|\mathbf{z}]\mathcal{P}[\mathbf{x}|\mathbf{y}]$ where the units within a layer are mutually independent given the activities in the layer above and are set according to equation 14 using the generative weights $\mathcal{H}$, $\mathcal{G}$ (and the biases $\mathcal{J}$ for the $\mathbf{z}$ layer that receives no other input). $\mathcal{R}$ and $\mathcal{S}$ are bottom-up parameters for recognition. For the Helmholtz machine, recognition is based on a bottom-up belief network which specifies $\mathcal{P}[\mathbf{y}, \mathbf{z}|\mathbf{x}] = \mathcal{P}[\mathbf{y}|\mathbf{x}]\mathcal{P}[\mathbf{z}|\mathbf{y}]$ (which is true) with an approximation of mutual independence within the $\mathbf{y}$ layer and within the $\mathbf{z}$ layer given the $\mathbf{y}$ activities (which is generally not true). b,c) Two non-linear generative models. Both figures give the generative connections allowing $\mathbf{y}$ units to describe activity patterns over $\mathbf{x}$ units. All units are stochastic binary units, with activation functions as given in the text. The weights with no originating nodes are biases. b) Explaining away. $x_1 = 1$ requires explaining by either $y_1 = 1$ or $y_2 = 1$ but not both (*ie* $\mathcal{P}[\mathbf{y} = \{0,0\}|x = 1] = 0.0033; \mathcal{P}[\mathbf{y} = \{1,1\}|x = 1] = 0.0033$ and $\mathcal{P}[\mathbf{y} = \{0,1\}|x = 1] = \mathcal{P}[\mathbf{y} = \{1,0\}|x = 1] = 0.4967$. c) if $y_1 = 1$ also generates $x_2 = 1$, then at question is whether or not there should be a direct negative influence from x_2 to y_2 in the recognition model.

Bottom-Up Method

The Helmholtz machine (see figure 3a) suggests building a bottom-up recognition model that is also a sigmoidal belief net and (at least in the stochastic version) drawing samples from this net to approximate the full recognition distribution. Two simplifications are made: a) despite the caveat above, the activities of units within a layer are forced to be independent given the activities in the layer below, to avoid having to parameterise and manipulate the full conditional distribution within each layer,[3] and b) the connections of the recognition belief net are set using an incorrect training procedure that chooses them to minimise (locally) a *wrong* error measure that is nonetheless computationally convenient. Write the states of all the units other than the inputs ($\mathbf{x}$) as α. Then, if the probability

[3] Note, however, that this does not mean that *all* the units are mutually independent.

accorded to a particular α by the recognition model is $\mathcal{Q}_\alpha(\mathcal{R})$, where $\mathcal{R}$ are the weights of the non-linear recognition model, then $\mathcal{R}$ should correctly be chosen to minimise $KL\{\mathcal{Q}_\alpha(\mathcal{R}), \mathcal{P}[\alpha|\mathbf{x}]\}$ (Hinton & Zemel *et al*, 1994). Instead, in the sleep learning procedure (Hinton *et al*, 1995), they are chosen to minimise $KL\{\mathcal{P}[\alpha|\mathbf{x}], \mathcal{Q}_\alpha(\mathcal{R})\}$. Since the Kullback-Leibler divergence is not symmetric, these quantities are not the same. If the divergence cannot be forced to be 0 (as is likely given the approximations employed), then the main difference is that minimising the first requires that $\mathcal{P}[\alpha|\mathbf{x}]$ be small whenever $\mathcal{Q}_\alpha(\mathcal{R})$ is small, whereas minimising the second requires that $\mathcal{Q}_\alpha(\mathcal{R})$ be small whenever $\mathcal{P}[\alpha|\mathbf{x}]$ is small.

This bottom-up method for approximating the recognition model works quite well in practice (see Frey *et al*, 1996). However, in cases such as explaining away (including the simple example of figure 3b) it fails. It assigns independent bottom-up probabilities of 0.5 to $y_1 = 1$ and $y_2 = 1$, and so 50% of the time chooses settings for $\mathbf{y}$ that are incorrect.

The bottom-up recognition inverse to the generative model in figure 3c is correct. The point is that if $x_1 = x_2 = 1$ for a particular case, then it must be that $y_1 = 1$ rather than $y_2 = 1$. Even though the generative weight from y_2 to x_2 is 0, the recognition weight from x_2 to y_2 is negative. A more interesting case of this is seen in the weight patterns for the 8-bit shifter problem in Dayan *et al* (1995). Units in the first hidden layer generate the activity of single pixels within both eyes, shifted by one pixel left or right with respect to each other. The recognition weights for these units have positive values for the pixels whose activities are actually generated, but inhibitory side-lobes from the neighbouring pixels, to avoid spurious activation.

It has been suggested that lateral connections within a layer might be used as in a Boltzmann machine solely for the recognition model (Dayan & Hinton, 1996). The generative model would still involve only top-down influences as in figure 3a (to ensure that the the generative model is still tractable), but the recognition model would employ lateral links to circumvent the requirement that the units within a layer be independent. The disadvantage is that computationally complex Gibbs sampling within a layer has to be used to instantiate recognition. Also, in cases such as explaining away, there would have to be an explicit negative lateral connection between y_1 and y_2.

Top-Down Methods

Mean-field The major alternatives to the bottom-up recognition model described above are mean field methods, pioneered for belief nets by Saul, Jaakkola & Jordan (1996) and Jaakkola, Saul & Jordan (1996). These effectively choose a parameterisation for an approximation to the recognition distribution for a particular case, and optimise the parameters to minimise the equivalent of the correct Kullback-Leibler divergence. Most approximations force *all* the units to be mutually independent (not just those units within a single layer, given the activities in the layer below). One can treat the linear and Gaussian case from the previous section exactly in mean-field terms, using a parameterisation with means for all the units and a particular covariance structure. Minimising the Kullback-Leibler divergence turns out to require satisfying a set of self-consistency equations at each unit, and there are algorithms that descend monotonically in the divergence whilst updating units asynchronously using only information local to a unit and its incoming and outgoing connections. In the linear Gaussian case, solving these self-consistency equations is exactly solving for the correct mean values.

The mean field theory of Jaakkola *et al* (1996) is a suitable non-linear counterpart. It uses as its sigmoid activation function the normal distribution function $\sigma(a) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{a} e^{-b^2/2} db$. This leads to a cost function $\mathcal{C}$ for minimisation for which the contribution from the terms in the $\mathbf{y}$ layer is $2\mathcal{C}_{\hat{\mathbf{y}}} =$:

$$\sum_{ij} \mathcal{G}_{ji}^2 \sigma(\hat{y}_j)\sigma(-\hat{y}_j) + \sum_j (\hat{y}_j - \sum_k \mathcal{H}_{kj}\sigma(\hat{z}_k))^2 + \sum_i (\hat{x}_i - \sum_j \mathcal{G}_{ji}\sigma(\hat{y}_j))^2 \quad (15)$$

summing for i over units in $\mathbf{x}$, for j over $\mathbf{y}$, for k over $\mathbf{z}$, where $\sigma(\hat{y}_j)$ is the mean activity of unit y_j, and $\hat{x}_i$ for input unit i is set so that $\sigma(\hat{x}_i)$ is very close to 0 or 1 as appropriate. Equivalent contributions to $\mathcal{C}$ come from the other layers in the network. This cost function is related to that for the linear and Gaussian case (equations 6 and 7) and for Olshausen & Field (1996) (equation 9). The contribution to $-\nabla_{\hat{y}_j}\mathcal{C}_{\hat{\mathbf{y}}}$ from the second and third (*ie* the prediction error) terms of equation 15 is

$$-(\hat{y}_j - \sum_k \mathcal{H}_{kj}\sigma(\hat{z}_k)) + \sum_i \mathcal{G}_{ji}\sigma'(\hat{y}_j)(\hat{x}_i - \sum_{j'} \mathcal{G}_{j'i}\sigma(\hat{y}_{j'}))$$

which has a very close parallel in the equivalent term for the linear and Gaussian hierarchical model as in $d\hat{\mathbf{y}}^a/dt$ in equation 12. Minimising $\mathcal{C}$ can either be accomplished using gradient descent, as in equation 12, or by updating $\hat{y}_j$ such that $\nabla_{\hat{y}_j}\mathcal{C} = 0$. In both cases only local information is used at each unit, and the bottom-up connections from units in a layer to units in the layer above must be the transpose of the generative weights in the opposite direction.

Mean field approximations such as this can cope well with inverting the generative model of figure 3b when $x = 1$, since there are two stable minima, one with $y_1 = 1; y_2 = 0$ and the other with $y_1 = 0; y_2 = 1$. Moreover, unlike the lateral Boltzmann machine, they do not require the existence of an explicit connection between y_1 and y_2.

Gibbs sampling An alternative approach to calculating the recognition distribution is to use a Markov-chain Monte-Carlo method such as Gibbs sampling. Neal (1992) discusses Gibbs sampling in sigmoidal belief nets in some detail, including the use of such samples for learning the generative weights. Here, we point out a simple approximation in the limit of small generative weights that shows the similarity with the other top-down uses of bottom-up connections.

In the case in figure 3a, with $\sigma(a) = 1/(1 + e^{-a})$, we are interested in sampling from $\mathcal{P}[\mathbf{y}, \mathbf{z}|\mathbf{x}]$. In the simplest version of Gibbs sampling, we visit each of the units in layers $\mathbf{y}$ and $\mathbf{z}$ in some random sequence, and choose a new state stochastically, taking into account influences from its parents (just the generative biases for the $\mathbf{z}$ units) and its children. Take unit y_1, writing $\bar{\mathbf{y}} = \{y_2, \ldots, y_n\}$.

We should set its new state to 1 according to $\mathcal{P}[y_1 = 1|\bar{\mathbf{y}}, \mathbf{z}, \mathbf{x}]$. It turns out to be most convenient to calculate instead

$$\rho_1 = \log \frac{\mathcal{P}[y_1 = 1|\bar{\mathbf{y}}, \mathbf{z}, \mathbf{x}]}{\mathcal{P}[y_1 = 0|\bar{\mathbf{y}}, \mathbf{z}, \mathbf{x}]} = \log \frac{\mathcal{P}[\mathbf{z}]\mathcal{P}[y_1 = 1|\mathbf{z}]\mathcal{P}[\bar{\mathbf{y}}|\mathbf{z}]\mathcal{P}[\mathbf{x}|y_1 = 1, \bar{\mathbf{y}}]}{\mathcal{P}[\mathbf{z}]\mathcal{P}[y_1 = 0|\mathbf{z}]\mathcal{P}[\bar{\mathbf{y}}|\mathbf{z}]\mathcal{P}[\mathbf{x}|y_0 = 1, \bar{\mathbf{y}}]}$$

$$= \textstyle\sum_k \mathcal{H}_{k1} z_k + \sum_i \log \frac{\mathcal{P}[x_i|y_1=1,\bar{\mathbf{y}}]}{\mathcal{P}[x_i|y_1=0,\bar{\mathbf{y}}]} \text{ as } \mathcal{P}[y_1 = 1|\mathbf{z}] = \sigma\left(\sum_k \mathcal{H}_{k1} z_k\right) \quad (16)$$

Since

$$\mathcal{P}[x_i|y_1 = 1, \bar{\mathbf{y}}] = \frac{x_i + (1 - x_i)e^{-\mathcal{G}_{1i} - \sum_{j \neq 1} \mathcal{G}_{ji} y_j}}{1 + e^{-\mathcal{G}_{1i} - \sum_{j \neq 1} \mathcal{G}_{ji} y_j}}$$

then, if $\mathcal{G}_{1i}$ is small compared with $\sum_{j \neq 1} \mathcal{G}_{ji} y_j$, it turns out that

$$\log \frac{\mathcal{P}[x_i|y_1 = 1, \bar{\mathbf{y}}]}{\mathcal{P}[x_i|y_1 = 0, \bar{\mathbf{y}}]} = \mathcal{G}_{1i}\left(x_i - \mathcal{P}[x_i = 1|y_1 = 0, \bar{\mathbf{y}}]\right).$$

Since $\mathcal{G}_{1i}$ is small, to zeroth order

$$\mathcal{P}[x_i = 1|y_1 = 0, \bar{\mathbf{y}}] = \mathcal{P}[x_i = 1|y_1 = 1, \bar{\mathbf{y}}] = \mathcal{P}[x_i = 1|\mathbf{y}]$$

is the top down prediction that $x_i = 1$ whatever the state of y_1. Using this in equation 16, we have:

$$\rho_1 \approx \sum_k \mathcal{H}_{k1} z_k + \sum_i \mathcal{G}_{1i}\left(x_i - \mathcal{P}[x_i = 1|\mathbf{y}]\right) \quad (17)$$

which consists of the obvious top-down influence from $\mathbf{z}$ and a bottom-up influence that tends to reduce the prediction error $(x_i - \mathcal{P}[x_i = 1|\mathbf{y}])$ for the state of x_i. Just as in the other top-down cases (such as the dynamic system in equation 8 or the mean field theory of Jaakkola *et al*, 1996) the prediction error is propagated bottom-up through the transpose of the generative weights $\mathcal{G}$. Gibbs sampling then requires that y_1 be set to 1 with probability $\sigma(\rho_1)$. As for mean field methods, explaining away can be handled by Gibbs sampling (although not quite as in equation 17, since the weights are not insubstantial), again without recourse to direct connections between y_1 and y_2.

For the generative model in figure 3c, note how making the recognition weight from x_2 to y_2 zero rather than negative makes it more complicated to work out that if $x_1 = x_2 = 1$, then $y_1 = 1$ rather than $y_2 = 1$.

IV Discussion

In this paper, we have studied the issue of inverting various sorts of directed belief net generative models. Such generative models are attractive as ways of capturing the essence of the hierarchical structure of cortex, where the activities of neurons in successively higher cortical areas represent the generation of successively more abstract entities in scenes. Inverting the generative models, *ie* going from sensory input to the activities of the neurons that represent its likely generators, is essential to interpret scenes, and also (though this has not been stressed here) to learn appropriate generative models. The inverse operation, called recognition, is akin to discrimination or classification, and, in many cases, doing it exactly is computationally intractable. We assume that the bottom-up weights in the cortex (from V1 to V2, *etc*) are important for recognition, but that they may operate in conjunction with the top-down and lateral weights. We have also seen some of the relationships amongst recent suggestions as to how cortex might implement a generative model.

Various schemes for performing recognition have been suggested. One class uses only bottom-up connections, either for exact recognition, as for factor analysis, or for approximate recognition, as for the Helmholtz machine. In the latter case, it is tractable to draw samples stochastically from the bottom-up model. Although purely bottom-up models are fast, a strong requirement suggested by evidence on the speed of processing images of objects, they suffer from a number of disadvantages in terms of the difficulty of incorporating top-down information, coping with occlusion, and integrating information from disparate parts of a scene further than the credible spread of feedforward connections. Further, in important cases such as explaining away, bottom-up models that make reasonable approximations, such as that the activities of units in a layer are mutually independent given the activities in the layer below, are incompetent.

An alternative scheme is suggested by mean field methods (Saul *et al*, 1996). Here a parameterised form is chosen for the recognition distribution for a particular case, and the parameters are set by an optimisation process. For suitable parameterisations, optimisation is achieved by a set of local operations. Although there are no bottom-up weights as such for mean field methods, we saw various cases in which the influence of the units in one layer on those in the layer above is calculated by passing some form of prediction error (*ie* the difference between their actual activation and the activation predicted on the basis of the states of the units in the layer above) through the transpose of the generative weights. Optimisation is iterative – this solves the problems mentioned for the purely bottom-up method, but raises questions as to the time required. We saw simple cases in which there is a difference between the bottom-up weights implied by purely bottom-up recognition and the bottom-up weights implied by this top-down scheme.

A further alternative was to use lateral weights within each layer. This was either to eliminate the requirement for iteration between layers (as in equation 8), or to fix problems with approximations made for bottom-up inference, as in explaining away.

Different generative models impose different requirements on their recognition inverses. Linear models with Gaussian noise are particularly simple – even in the case of Chou *et al* (1994) with multiple layers and only partial connectivity, there are algorithms for working out the true recognition inverses which require nothing more than one bottom-up and one top-down pass. Non-linear models, such as the layered belief nets of figure 3a do not possess such tractable inverses, and so approximations are necessary.

This analysis is unsatisfyingly incomplete. Foremost, the interaction between bottom-up, top-down and lateral connections is still open. Given the structural differences between lateral and bottom-up weights, one might expect to find a computational difference too. Top-down influences must clearly be felt during recognition; however it is not apparent how to have bottom-up weights that implement one-shot recognition in ideal circumstances, but can be used for interacting bottom-up and top-down processing in less ideal cases. The role of the lateral weights is also mysterious. One suggestion is that they help repair problems with too restrictive approximations in the bottom-up recognition model, but this use imposes strong requirements on the presence of dense connections (so that all cases of explaining away in the generative model can be properly handled) and/or on the time available for Gibbs sampling. If the lateral weights are also involved in specifying the generative model, then the whole system becomes a form of Boltzmann machine.

An issue raised by the Kalman filter models of section 2 is the various covariance matrices for the activities at different layers. In the linear cases with Gaussian noise that we have considered, the covariance matrices are fixed once the parameters of the generative model are fixed, and they do not depend on the input for a particular scene. In non-linear cases, and in cases which include temporal effects (Rao & Ballard, 1995), this is not true. Retaining just the diagonal terms of the covariance matrix of the activities is simple (Sutton, 1992). Retaining the off-diagonal terms is more complicated because of their numerosity and the complexity of the calculations that lead to them.

Apart from capturing its general hierarchical characteristics, none of the models in this paper is very faithful to the real details of cortical processing. Apart from the many complexities of the structure of lower and higher visual processing areas, important issues are that the anatomical spread of top-down connections is *broader* than that of the bottom-up connections, and that there appears to be a difference over developmental time in their specification. For instance, the top-down connections from V2 to V1 in humans wait in the lowest cortical layers in V1 for a few months, and only migrate to what becomes their main targets in layer 2 at the same time that the intracortical lateral connections within layer 2 are also maturing (Burkhalter, 1993). This might favour a different class of models (Luttrell, 1995) from the ones discussed here in which the bottom-up recognition weights are actually primary, heeding some other developmental call, and the top-down and lateral weights merely build the generative model that is most consistent with whatever activities the recognition model ultimately specifies.

Acknowledgements

I am most grateful to Brendan Frey, Geoff Hinton, Tommi Jaakkola, Mike Jordan, Zhaoping Li, Radford Neal, Bruno Olshausen and Lawrence Saul for helpful discussions. This work was funded by NIMH grant R29-MH55541-01. All opinions expressed are those of the author.

References

1. Burkhalter, A (1993). Development of forward and feedback connections between areas V1 and V2 of human visual cortex. *Cerebral Cortex,* **3**, 476-87.
2. Chou, KC, Willsky, AS & Benveniste, A (1994). Multiscale recursive estimation, data fusion, and regularization. *IEEE Transactions on Automatic Control,* **39**, 464-478.
3. Chou, KC, Willsky, AS & Nikoukhah, R (1994). Multiscale systems, Kalman filters, and Riccati equations. *IEEE Transactions on Automatic Control,* **39**, 479-492.
4. Dayan, P (1996). A Hierarchical model of visual rivalry. Submitted to *Neural Information Processing Systems, 9.*
5. Dayan, P & Hinton, GE (1996). Varieties of Helmholtz Machine. *Neural Networks,* in press.
6. Dayan, P, Hinton, GE, Neal, RM & Zemel, RS (1995). The Helmholtz machine. *Neural Computation,* **7**, 889-904.
7. Dempster, AP, Laird, NM & Rubin, DB (1977). Maximum likelihood from incomplete data via the EM algorithm. *Proceedings of the Royal Statistical Society,* **B 39**, 1-38.
8. Everitt, BS (1984). *An Introduction to Latent Variable Models.* London: Chapman and Hall.
9. Felleman DJ & Van Essen DC (1991) Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex,* **1**, 1-47.
10. Frey, BJ, Hinton, GE & Dayan, P (1995). Does the wake-sleep algorithm produce good density estimators? *Advances in Neural Information Processing Systems, 8,* forthcoming.
11. 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).
12. Hinton, GE, Dayan, P, Frey, BJ & Neal, RM (1995). The wake-sleep algorithm for unsupervised neural networks. *Science,* **268**, 1158-1160.
13. Hinton, GE, Dayan, P & Revow, M (1996). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks,* forthcoming.
14. 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, Massachusetts: MIT Press, 282-317.
15. 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.
16. Jaakkola, T, Saul, LK & Jordan, MI (1996). Fast learning by bounding likelihoods in sigmoid type belief networks. *Advances in Neural Information Processing Systems, 8,* forthcoming.
17. Jacobs, RA, Jordan, MI, Nowlan, SJ & Hinton, GE (1991). Adaptive mixtures of local experts, *Neural Computation,* **3**, 79-87.
18. Jolliffe, IT (1986) *Principal Component Analysis,* New York: Springer-Verlag.
19. Kawato, M, Hayakama, H & Inui, T (1993). A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network,* **4**, 415-422.
20. Kinderman, R & Snell, JL (1980). *Markov Random Fields and their Applications.* American Mathematical Society.

21. Leopold, DA & Logothetis, NK (1996). Activity changes in early visual cortex reflect monkeys' percepts during binocular rivalry. *Nature,* **379**, 549-554.
22. Logothetis, NK, Leopold, DA & Sheinberg, DL (1996). What is rivalling during binocular rivalry. *Nature,* **380**, 621-624.
23. Luettgen, MR & Willsky, AS (1995). Likelihood calculation for a class of multiscale stochastic models, with application to texture discrimination. *IEEE Transactions on Image Processing,* **4**, 194-207.
24. Luttrell, SP (1995). A componential self-organizing neural network. Submitted to *Advances in Neural Information Processing Systems, 8.*
25. Marroquin, JL (1985). *Probabilistic Solution of Inverse Problems.* PhD Thesis, AI Lab, MIT, Cambridge, MA.
26. 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.
27. Neal, RM (1992). Connectionist learning of belief networks. *Artificial Intelligence,* **56**, 71-113.
28. Neal, RM (1993). *Probabilistic Inference using Markov Chain Monte Carlo Methods.* Technical Report CRG-TR-93-1. Department of Computer Science, University of Toronto.
29. Neal, RM & Dayan, P (1996). *Factor Analysis using Delta-Rule Wake-Sleep Learning.* TR-9607, Department of Statistics, University of Toronto.
30. Nowlan, SJ (1991). *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures.* CMU Technical Report CMU-CS-91-126, Carnegie-Mellon University, Pittsburgh PA.
31. Olshausen, BA & Field, DJ (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature,* **381**, 607-609.
32. Pearl, J (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, CA: Morgan Kaufmann.
33. Perrett, DI, Rolls, ET & Caan W (1982). Visual neurones responsive to faces in the monkey temporal cortex. *Experimental Brain Research,* **47**, 329-342.
34. Poggio, T & Torre, V (1984). Ill-posed problems and regularization analysis in early vision. *Proceedings of ARPA Image Understanding Workshop,* 257-263.
35. Rao, PNR & Ballard, DH (1995). *Dynamic Model of Visual Memory predicts Neural Response Properties in the Visual Cortex.* Technical report 95.4, Department of Computer Science, Rochester, NY.
36. Rissanen, J (1989). *Stochastic Complexity in Statistical Inquiry.* Singapore: World Scientific.
37. Rubin, DB & Thayer, DT (1982). EM algorithms for maximum likelihood factor analysis. *Psychometrika,* **47**, 69-76.
38. Saul, LK, Jaakkola, T & Jordan, MI (1996). Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research,* **4**, 61-76.
39. Sutton RS (1992). Gain adaptation beats least squares? In *Proceedings of the 7th Yale Workshop on Adaptive and Learning Systems.*
40. 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.
41. Zemel, RS (1994). *A Minimum Description Length Framework for Unsupervised Learning.* PhD Dissertation, Computer Science, University of Toronto, Canada.

This article was processed using the LaTeX macro package with LLNCS style