NIPS*97 Submission
Category: Algorithms and Architectures
Preference: Oral

Hierarchical Non-linear Factor Analysis and Topographic Maps

Zoubin Ghahramani and Geoffrey E. Hinton
Department of Computer Science
University of Toronto
Toronto, Ontario, M5S 3H5, Canada
{zoubin,hinton}@cs.toronto.edu

Abstract

We first describe a hierarchical, generative model that can be viewed as a non-linear generalization of factor analysis and can be implemented in a neural network. The model performs perceptual inference in a probabilistically consistent manner by using top-down, bottom-up and lateral connections. These connections can be learned using simple rules that require only locally available information. We then show how to incorporate non-adaptive lateral connections into the generative model. The model extracts a sparse, distributed, hierarchical representation of depth from simplified random-dot stereograms and the localized disparity detectors in the first hidden layer form a topographic map.

1 Introduction

The generative model underlying factor analysis assumes that each observed data vector is caused by linearly combining the real-valued activation states of all the factors and then adding independent Gaussian noise to each component of the data vector. A major advantage of this generative model is that, given a data vector, the posterior probability distribution in the space of factor states is a Gaussian whose mean is a linear function of the data. It is therefore tractable to compute the posterior distribution exactly and to use this distribution when learning the parameters of the model (the factor loadings and noise variances). A major disadvantage is that factor analysis is a linear model that is insensitive to the higher order statistical structure of the observed data vectors.

One way to make factor analysis non-linear is to use a mixture of factor analyser modules, each of which captures a different linear regime in the data. We can view the factors of *all* of the modules as a large set of basis functions for describing

the data and the process of selecting one module then corresponds to selecting an appropriate subset of the basis functions. Since the number of subsets under consideration is only linear in the number of modules, it is still tractable to compute the full posterior distribution when given a data point. Unfortunately, this mixture model is often inadequate. Consider, for example, a typical image that contains multiple objects. To represent the pose and deformation of each object we want a componential representation of the object's parameters which could be obtained from an appropriate factor analyser. But to represent the multiple objects we need several of these componential representations at once so the pure mixture idea is not tenable. A more powerful non-linear generalization of factor analysis is to have a large set of factors and to allow *any* subset of the factors to be selected. This can be achieved by using a generative model in which there is a high probability of generating factor activations of exactly 0.

2 Rectified Gaussian Belief Nets

The Rectified Gaussian Belief Net (RGBN) uses multiple layers of units with states that are either positive real values or zero. Its main disadvantage is that the computation of the posterior distribution over the factor states when given a data vector involves Gibbs sampling. In general, Gibbs sampling can be very time consuming, but in practice 10 to 20 samples per unit have proved adequate and there are theoretical reasons for believing that learning can work well even when the Gibbs sampling fails to reach equilibrium (Neal and Hinton, 1993).

We first describe the RGBN without considering neural plausibility. Then we show how lateral interactions within a layer can be used to perform probabilistic inference correctly using locally available information. This makes the RGBN far more plausible as a neural model than a sigmoid belief net (Neal, 1992; Lewicki and Sejnowski, 1997) because it means that Gibbs sampling can be performed without requiring units in one layer to see the total top-down input to units in the layer below.

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

$$\hat{y}_j = g_{0j} + \sum_k g_{kj}[y_k]^+ \tag{1}$$

The rectified state $[y_j]^+$ therefore has a Gaussian distribution above zero, but all of the mass of the Gaussian that falls below zero is concentrated in an infinitely dense spike at zero as shown in figure 1a. This infinite density creates problems if we attempt to use Gibbs sampling over the rectified states, so, following a suggestion by Radford Neal, we perform Gibbs sampling on the unrectified states.

Consider a unit, j, in some intermediate layer of a multilayer RGBN. Suppose that we fix the unrectified states of all the other units in the net. To perform Gibbs sampling, we need to stochastically select a value for y_j according to its posterior distribution given the unrectified states of all the other units. If we think in terms of energy functions, which are equal to negative log probabilities (up to a constant), the rectified states of the units in the layer above contribute a quadratic energy term by determining $\hat{y}_j$. The unrectified states of units, i, in the layer below contribute a constant if $[y_j]^+$ is 0, and if $[y_j]^+$ is positive they each contribute a quadratic term

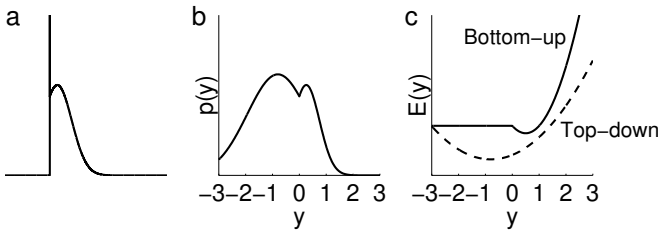

Figure 1: **a**) A probability distribution in which all the mass of the Gaussian below zero has been replaced by an infinitely dense spike at zero. **b**) Schematic of the posterior density of an unrectified state of a unit. **c**) Bottom-up and top-down energy functions corresponding to **b**.

because of the effect of $[y_j]^+$ on $\hat{y}_i$.

$$E(y_j) = \frac{(y_j - \hat{y}_j)^2}{2\sigma_j^2} + \sum_i \frac{(y_i - \sum_h g_{hi}[y_h]^+)^2}{2\sigma_i^2} \tag{2}$$

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

Given samples from the posterior, the generative weights of a RGBN can be learned by using the online delta rule to maximize the log probability of the data.[1]

$$\Delta g_{ji} = \epsilon \, [y_j]^+ \, (y_i - \hat{y}_i) \tag{3}$$

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

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

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

3 The Role of Lateral Connections in Perceptual Inference

In RGBNs and other layered belief networks, fixing the value of a unit in one layer causes correlations between the parents of that unit in the layer above. One of the main reasons why purely bottom-up approaches to perceptual inference have proven inadequate for learning in layered belief networks is that they fail to take into account this phenomenon, which is known as "explaining away" (Pearl, 1988).

[1] If Gibbs sampling has not been run long enough to reach equilibrium, the delta rule follows the gradient of the *penalized* log probability of the data (Neal and Hinton, 1993). The penalty term is the Kullback-Liebler divergence between the equilibrium distribution and the distribution produced by Gibbs sampling. Other things being equal, the delta rule therefore adjusts the parameters that determine the equilibrium distribution to reduce this penalty, thus selecting models for which Gibbs sampling works quickly. However, although the delta rule takes into account the effect of the parameter changes on the true posterior distribution, it ignores their effect on the distribution reached by the Gibbs sampling.

Lee and Seung (1997) introduced a clever way of using lateral connections to handle explaining away effects during perceptual inference. Consider the network shown in figure 3. One contribution, E_{below}, to the energy of the state of the network is the squared difference between the unrectified states of the units in the bottom layer, y_j, and the top-down expectations generated by the states of units in the layer above. Assuming the local noise models for the bottom layer units all have unit variance, and ignoring biases and constant terms that are unaffected by the states of the units

$$E_{\text{below}} = \frac{1}{2}\sum_j (y_j - \hat{y}_j)^2 = \frac{1}{2}\sum_j (y_j - \textstyle\sum_k [y_k]^+ g_{kj})^2. \tag{5}$$

Rearranging this expression and setting $r_{jk} = g_{kj}$ and $m_{kl} = -\sum_j g_{kj} g_{lj}$ we get

$$E_{\text{below}} = \frac{1}{2}\sum_j y_j^2 - \sum_k [y_k]^+ \sum_j y_j r_{jk} - \frac{1}{2}\sum_k [y_k]^+ \sum_l [y_l]^+ m_{kl}. \tag{6}$$

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

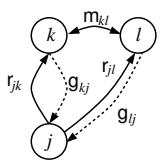

Figure 2: A network with lateral connections to handle explaining away effects.

Seung's trick can be used in an RGBN and it eliminates the most neurally implausible aspect of this model which is that a unit in one layer appears to need to send both its state y and the top-down prediction of its state $\hat{y}$ to units in the layer above. Using the lateral connections, the units in the layer above can, in effect, compute all they need to know about the top-down predictions. In computer simulations, we can simply set each lateral connection m_{kl} to be $-\mathbf{g}_k \cdot \mathbf{g}_l$. It is also possible to learn these lateral connections in a more biologically plausible way by driving units in the layer below with unit-variance independent Gaussian noise and using a simple anti-Hebbian learning rule. Similarly, a purely local learning rule can learn recognition weights equal to the generative weights. If units at one layer are driven by unit-variance, independent Gaussian noise, and these in turn drive units in the layer below using the generative weights, then Hebbian learning between the two layers will learn the correct recognition weights (Hinton and Ghahramani, in press).

4 Lateral Connections in the Generative Model

When the generative model contains only top-down connections, lateral connections are useful during perceptual inference. But it is also interesting to explore the use of symmetrical lateral connections to create Markov Random Fields (MRF's) within each layer of the generative model. Symmetric lateral interactions between the unrectified states of units within a layer correspond to adding a quadratic term to the energy function,

$$E_{\text{MRF}} = \frac{1}{2}\sum_k \sum_l M_{kl}\, y_k y_l. \tag{7}$$

This MRF can, for example, cause nearby units in a layer to have *a priori* correlated activities in the generative model. During Gibbs sampling, each unit adds this input to its top-down energy contribution and samples from its unrectified state as before. Learning is considerably more difficult. The difficulty stems from the need to know the derivatives of the partition function of the MRF for each data vector. This partition function depends on the top-down inputs to a layer so it varies from one data vector to the next, even if the lateral connections themselves are non-adaptive. Fortunately, since both the MRF and the top-down prediction define Gaussian probability distributions over the states of the units in a layer, these derivatives can be easily calculated. Assuming unit variances and using matrix notation,

$$\frac{\partial \log P}{\partial G} = \left(\mathbf{y}_b - G\left[\mathbf{y}_a\right]^+\right)\left[\mathbf{y}_a\right]^{+T} + M(I+M)^{-1}G\left[\mathbf{y}_a\right]^+\left[\mathbf{y}_a\right]^{+T} \tag{8}$$

where $\mathbf{y}_a$ and $\mathbf{y}_b$ are the vectors of activities of units in the layer above and below, respectively. G is the generative weight matrix from a to b, M is the MRF matrix acting on layer b and T denotes matrix transpose. The first term is the delta rule in matrix form (equation 3); the second term is the derivative of the partition function which unfortunately involves a matrix inversion.

We will assume a fixed structure for the MRF throughout the paper. However, since the partition function for a multivariate Gaussian is analytical it is also possible to learning the lateral connections in the MRF (Dayan, 1997).

5 Discovering depth in simplified stereo pairs

Consider the following multi-stage generative process for stereo pairs. Random dots of uniformly distributed intensities are scattered sparsely on a one-dimensional surface, and the image is blurred with a Gaussian filter. This surface is then randomly placed at one of two different depths, giving rise to two possible left-to-right disparities between the images seen by each eye. Separate Gaussian noise is then added to the image seen by each eye. Twelve example images generated in this manner are shown in figure 3a.

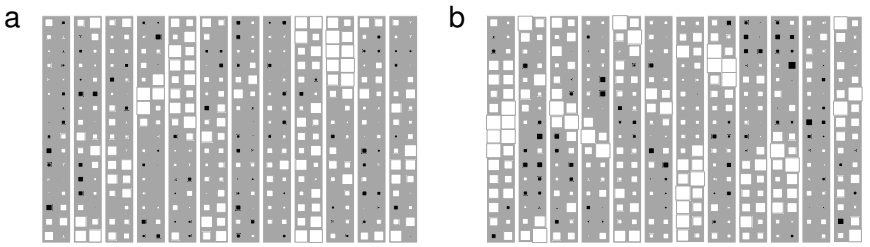

Figure 3: **a**) Sample data from the simplified stereo disparity problem. The left and right column of each 2×16 image are the inputs to the left and right eye, respectively. The value of a pixel is represented by the size of the square, with white being positive and black being negative, and periodic boundary conditions were used. Notice that the pixel noise makes it difficult to infer the disparity, *i.e.* the vertical shift between the left and right columns, in some images. **b**) Sample outputs generated by the model after learning.

We trained a three-layer RGBN consisting of 32 visible units, 32 units in the first hidden layer and 1 unit in the second hidden layer on the 16-pixel wide stereo disparity problem. Each of the 32 hidden units in the first hidden layer was connected to the entire array of 32 visible units, *i.e.* it had inputs from both eyes. The hidden units in this layer were also laterally connected in an MRF so that nearby units

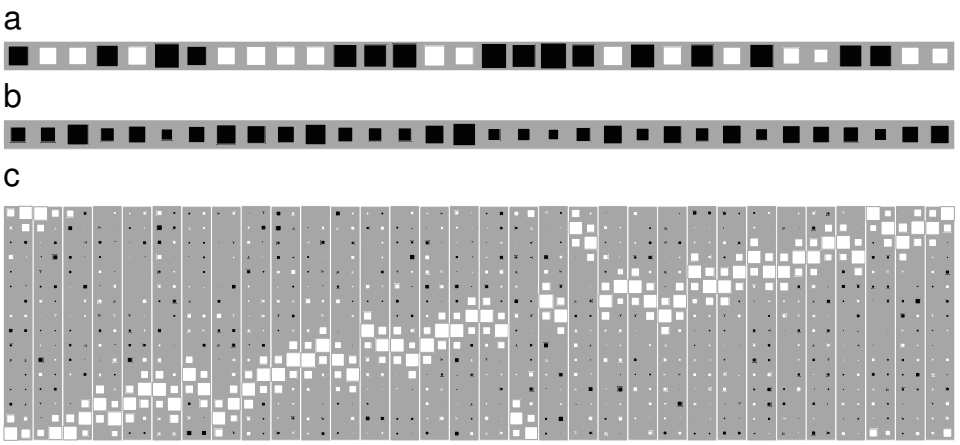

Figure 4: Generative weights of a three-layered RGBN after being trained on the stereo disparity problem. **a**) Weights from the top layer hidden unit to the 32 middle-layer hidden units. **b**) Biases of the middle layer hidden units, and **c**) weights from the hidden units to the 2×16 visible array.

excited each other and more distant units inhibited each other, with the net pattern of excitation being a difference of two Gaussians. This MRF was initialized with large weights which decayed exponentially to zero over the course of training. The network was trained for 30 passes through a data set of 1000 images using a different, random order for each pass. For each image we used 16 iterations of Gibbs sampling to approximate the posterior distribution over hidden states. Each iteration consisted of sampling every hidden unit once in a fixed order. The states after the fourth iteration of Gibbs sampling were used for learning, with a learning rate of 0.1 and a weight decay parameter of 0.001. Since the top level of the generative process makes a discrete decision between left and right global disparity we used a trivial extension of the RGBN in which the top level unit saturates both at 0 and 1.

Sixteen of the hidden units learned to become local left-disparity detectors, while the other sixteen became local right-disparity detectors (figure 4c). The unit in the second hidden layer has positive weights connecting it to leftward disparity detecting hidden units in the layer below, and negative weights for the rightward units (figure 4a). When presented with novel input images, the top unit was active for 89% of images with leftward disparity and inactive for 85% of images with rightward disparity[2]. A random sample of images generated by the model after learning is shown in figure 3b.

In addition to forming a hierarchical distributed representation of depth, units in the hidden layer self-organised into a topographic map. The MRF caused high correlations between nearby units early in learning, which in turn resulted in nearby units learning similar weight vectors. The emergence of topography depended on the strength of the MRF and on the speed with which it decayed. Results were relatively insensitive to other parametric changes. Similar global topographic maps resulted if we ignored the derivative of the partition function and applied the local delta rule (equation 3).

[2]The 16^{th} Gibbs sample was used to compute the activity and a threshold of 0.25 was chosen to distinguish active from inactive.

6 Discussion

Classical models of topography formation such as Kohonen's self-organising map (Kohonen, 1982) and the elastic network (Durbin and Willshaw, 1987; Goodhill and Willshaw, 1990) can be thought of as variations on mixture models where additional constraints have been placed to encourage neighboring hidden units to have similar generative weights. The problem with a mixture model is that it cannot handle images in which there are several things going on at once. In contrast, we have shown that topography can arise in a much richer hierarchical and componential generative model by inducing correlations between neighboring units.

There is a sense in which topography is a necessary consequence of the lateral connection trick used for perceptual inference. It is infeasible to interconnect all pairs of units in a cortical area. If we assume that direct lateral interactions (or interactions mediated by interneurons) are primarily local, then widely separated units will not have the apparatus required for explaining away. Consequently the computation of the posterior distribution will be incorrect unless the generative weight vectors of widely separated units are orthogonal. If the generative weights are constrained to be positive, the only way two vectors can be orthogonal is for each to have zeros wherever the other has non-zeros. Since the redundancies that the hidden units are trying to model are typically spatially localised, it follows that widely separated units must attend to different parts of the image and units can only attend to overlapping patches if they are laterally interconnected. The lateral connections in the generative model can be used to assist in the formation of the topography required for correct perceptual inference.

Acknowledgements

We thank P. Dayan, B. Frey, G. Goodhill, D. MacKay, R. Neal and M. Revow. The research was funded by NSERC and ITRC. GEH is the Nesbitt-Burns fellow of CIAR.

References

Dayan, P. (1997). Recurrent sampling models. *Technical Report*.

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

Goodhill, G. J. and Willshaw, D. J. (1990). Application of the elatic net algorithm to the formation of ocular dominance stripes. *Network: Computation in Neural Systems*, 1:41–59.

Hinton, G. E. and Ghahramani, Z. (In Press). Generative models for discovering sparse distributed representations. *Philos. Trans. Roy. Soc. London* **B***: Biol. Sci.*

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

Lee, D. D. and Seung, H. S. (1997). Unsupervised learning by convex and conic coding. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.

Lewicki, M. S. and Sejnowski, T. J. (1997). Bayesian unsupervised learning of higher order structure. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.

Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113.

Neal, R. M. and Hinton, G. E. (1993). A new view of the EM algorithm that justifies incremental and other variants. *Unpublished Manuscript*.

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