A Bayesian Unsupervised Learning Algorithm that Scales

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

Abstract

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

Introduction

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

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

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

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

Sparse Distributed Representations

Factor analysis and mixtures of Gaussians are at opposite ends of a spectrum of possible learning algorithms. In factor analysis, the representation is "componential" or "distributed" because it involves states in all of the hidden units. However, it is also linear and is therefore limited to capturing the informa-

tion in the pairwise covariances of the visible units. All the higher-order structure is invisible to it. At the other end of the spectrum, mixtures of Gaussians have localist representations because each data vector is assumed to be generated from a single hidden unit. This is an exponentially inefficient representation: each datapoint is represented by the identity of the winning hidden unit (*i.e.* the cluster it belongs to). So for the representation to contain, on average, n bits of information about the data, there must be at least 2^n hidden units. However, it is non-linear and with enough hidden units it can capture all of the higher-order structure in the data.

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

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

Scaling

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

A Brief History

The Boltzmann machine (Hinton and Sejnowski, 1986) was, perhaps, the first neural network learning algorithm to be based on an explicit generative model that used distributed, non-linear representations. Boltzmann machines use stochastic binary units and, with h hidden units, the number of possible representations of each datapoint is 2^h. It would take exponential time to compute the posterior distribution across all of these possible representations and most of them would typically have probabilities very close to 0, so the Boltzmann machine uses a Monte Carlo method known as Gibbs sampling (Hinton and Sejnowski, 1983; Geman and Geman, 1984) to stochastically pick representations according to their posterior probabilities. Unfortunately, the learning algorithm is extremely slow and, as a result, the unsupervised form of the algorithm never really succeeded in extracting interesting hierarchical representations.

Neal (1992) realised that learning is considerably more efficient if, instead of symmetric connections, the generative model uses directed connections that form an acyclic graph. This kind of generative model is called a belief network and it has an important property that is missing in models whose generative connections form cycles: It is straightforward to compute the joint probability of a datapoint and a configuration of states of all the hidden units. Neal investigated sigmoid belief nets (SBN) that consist of multiple layers of binary stochastic units. As with Boltzmann machines, it is exponentially expensive to compute the exact posterior distribution over the hidden units of an SBN when given a datapoint, so Neal used Gibbs sampling. With a particular datapoint clamped on the visible units, the hidden units are visited one at a time. Each time hidden unit j is visited, its state is stochastically selected to be 1 or 0 according to the conditional probabilities of those two states given the current state of all the other hidden and visible units in the network. Until very recently (Lewicki and Sejnowski, 1997), sigmoid belief nets were widely ignored as models of neural computation. This may be because the computation of the conditional probabilities needed for Gibbs sampling requires unit j to observe not only the states of units in the layer below but also their net top-down inputs from other units.

Becker and Hinton (1992) examined learning in models where different sub-networks attempt to maximize the mutual information between their outputs. One limitation of models which maximize information transfer is that they do not in general correspond to any generative model of the data. Thus, it is difficult, for example, to use the machinery of top-down and bottom-up Bayesian inference in such models. Also, if the inputs to the sub-networks are allowed to overlap, it is trivial to achieve very high mutual information without modeling the data.

Rectified Gaussian Belief Nets

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

We first describe the RGBN without considering neural plausibility. Then we show how lateral interactions within a layer can be used to perform explaining away correctly[1]. This

[1] "Explaining away" refers to the situation where causes that are *a priori* independent become dependent conditioned on some observed effect. For example, having the sprinkler on and whether it rains or not may be *a priori* independent. Now, assume that we observe that the ground is wet. Either cause can explain away the observation.

makes the RGBN far more plausible as a neural model and leads to a very natural explanation for the prevalence of topographic maps in cortex.

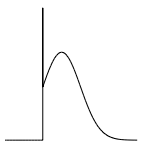

Figure 1: The rectified Gaussian.

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

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

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

Sampling from the posterior distribution

Consider a unit, j, in some intermediate layer of a multilayer RGBN. Suppose that we fix the unrectified states of all the other units in the net. To perform Gibbs sampling, we need to stochastically select a value for y_j according to its posterior distribution given the unrectified states of all the other units.

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

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

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 2b). The Gaussian posterior distributions corresponding to the two quadratics must agree at $y_j = 0$ (figure 2a). Because the posterior distribution is piecewise Gaussian it is possible to perform Gibbs sampling exactly and fairly efficiently.

One of them is probably true, but both of them together are unlikely, and therefore the two causes may be *a posteriori* anti-correlated.

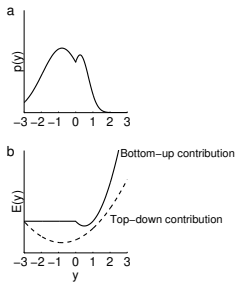

Figure 2: **a** Schematic of the posterior density of an unrectified state of a unit. **b** Bottom-up and top-down energy functions corresponding to **a**.

Learning the parameters of an RGBN

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

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

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

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

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.

The Role of Lateral Connections

Seung (personal communication, 1996) introduced a clever way of using lateral connections to handle explaining away effects. Consider the network shown in figure 3. One contribution, E, to the energy of the state of the network is the squared difference between the unrectified states of the units in the bottom layer, y_j, and the top-down expectations generated by the states of units in the layer above. Assuming the local noise models for the visible units all have unit variance, and ignoring biases and constant terms that are unaffected by the states of the units

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

This expression can be rearranged to give

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

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

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

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

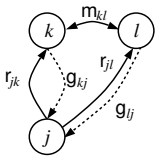

Figure 3: A network with lateral connections to handle explaining away effects.

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

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

There is one remaining difficulty that is a consequence of our decision to perform Gibbs sampling on the unrectified states. A unit needs to send its unrectified state to units in the layer above and its rectified state to units in the layer below.

A reason for topographic maps

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. Dayan, Hinton, Neal and Zemel (1995) show that when the wrong posterior distribution is used, the learning algorithm will alter the generative weights to make the true posterior distribution more like the distribution that is actually used. So the generative weight vectors of widely separated units will tend to become orthogonal. If the generative weights are constrained to be positive, the only way two vectors can be orthogonal is for each to have zeros whereever 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.

Discovering depth in simplified stereo pairs

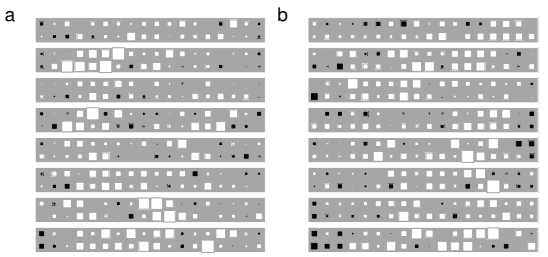

Figure 4: **a** Sample data from the simplified stereo disparity problem. The top and bottom row of each 2×18 image are the inputs to the left and right eye, respectively. **b** Sample outputs generated by the model after learning.

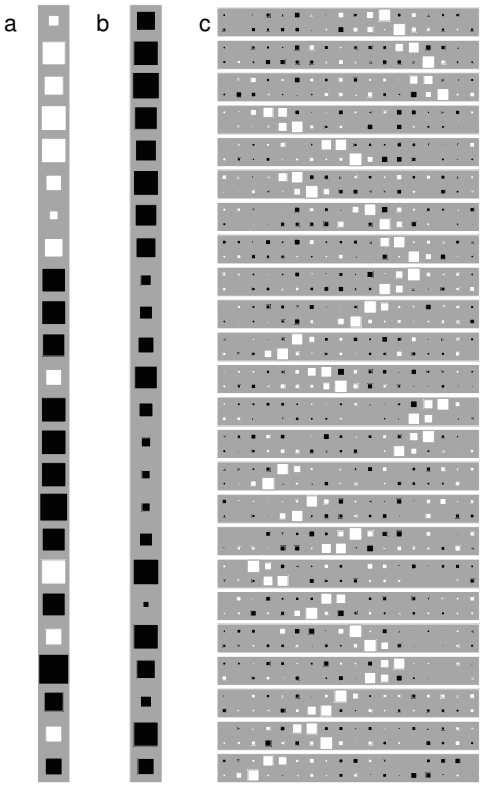

Figure 5: Generative weights of a three-layered RGBN after being trained on the stereo disparity problem. **a** Weights from the top layer hidden unit to the 24 middle-layer hidden units. **b** Biases of the middle layer hidden units, and **c** weights from the hidden units to the 2×18 visible array.

A problem in which discovering the higher order structure of a dataset has presented difficulties for some previous unsupervised learning algorithms is the one-dimensional stereo disparity problem (Becker and Hinton, 1992). Consider 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. Eight example images generated in this manner are shown in figure 4a.

We trained a three-layer RGBN consisting of 36 visible units, 24 units in the first hidden layer and 1 unit in the second hidden layer on the 18-pixel wide stereo disparity problem. Each of the 24 hidden units in the first hidden layer was connected to the entire array of 36 visible units, *i.e.* it had inputs from both eyes. The network was trained for 10 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 on every other iteration were used for learning, with a learning rate of 0.1 and a weight decay parameter of 0.01. Since the top level of the generative process makes a discrete decision between vertical and horizontal bars, we tried both the RGBN and a trivial extension of the RGBN in which the top level unit saturates both at 0 and 1. We show results for this extension, as it resulted in slightly cleaner representations at the top level. Results were relatively insensitive to other parametric changes.

Twelve of the hidden units learned to become local left-disparity detectors, while the other twelve became local right-disparity detectors (figure 5c). 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 5a). When presented with novel input images, the top unit was active for 87% of images with leftward disparity and inactive for 91% of images with rightward disparity. A random sample of images generated by the model after learning is shown in figure 4b.

Larger images improve learning

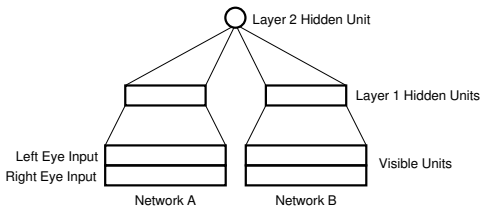

Figure 6: The double-network architecture, with two banks of 2×9 visible units, two banks of 12 first layer hidden units, and a single shared second layer hidden unit.

The goal of the first experiment was to determine the effect of scaling the number of visible and hidden units of a network of fixed depth. We compared learning and inference on the stereo disparity problem in two different network architectures. In the first architecture, each 18 pixel retinal image was divided into a left and a right half and the first layer hidden was similarly divided into a left half (Network A) and a right half (Network B), as shown in figure 6. The two networks shared the unit at second hidden layer, but otherwise had no interconnections. This double-network architecture was compared to a single-network architecture consisting of either Network A or B, including the unit at the top layer. Both the double- and single-networks were trained on exactly the same 1000 stereo images and learning parameters as in the previous example, where any particular single-network only saw one half of the image at a time.

Throughout learning, the two architectures were tested on all 1000 noisy images to determine whether the top unit had discovered a representation of depth. All four runs of the double-network architecture discovered this higher order structure in the data within the 10 000 time steps of learning, compared to only two out of six runs of the single-network (sample runs are shown in figure 7). While there is no reason to think that the single networks would not eventually discover this higher-order feature, this experiment suggests that, for this simple vision problem, increasing the network size to accommodate a larger image leads to faster rather than slower learning.

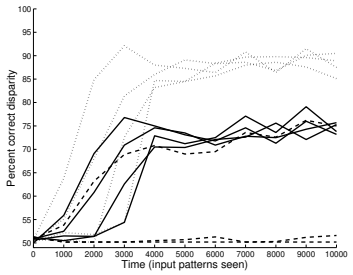

Figure 7: Percent correct disparity as a function of learning time for 4 runs of the double-network, when presented with both sides of the image (dotted lines), the double-network when presented with the left half of the image only, with units in the right side unclamped (solid lines) and 3 runs of the single-network trained on the left half and presented the left half (dashed lines).

We also examined the speed of perceptual inference, as measured by the convergence of the activity of the top layer unit, for these two network sizes. The top unit of the double-network, when presented with both sides of the image, converged reliably to two distinct values for images of the two disparities (figure 8a). The asymptotes were nearer to each other when only one half of the image was presented, and for the single-network architecture (figure 8b-c). It is difficult to say which architecture lead to faster perceptual inference.

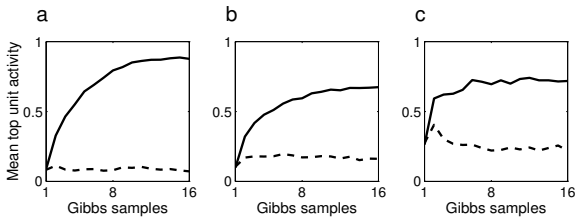

Figure 8: Activity of the top unit during Gibbs sampling when the clamped image has leftward (solid) and rightward (dashed) disparity. **a** a double-network when presented with both sides of the image; **b** a double-network when presented with the left half of the image only; **c** a single-network which successfully discovered depth, when presented with the left half of the image.

Deeper networks learn (slightly) better hidden representations

The goal of the second experiment was to investigate the effect of network depth on learning. We compared double-networks of the kind shown in figure 6 with a similar network

lacking the unit at the top layer. Under one hypothesis, the presence of the top unit should slow learning since at first it is simply introducing noise and spurious correlations in the layer below. On the other hand, the network may be able to use the top unit to clean up representations at the level below and therefore speed learning.

To compare the networks with and without a unit at the top layer, we calculated the correlation between the hidden layer representation and the true noise-free intensities of the image. This measure increased very rapidly at the beginning of learning for both types of networks. After this initial phase, the two hidden layer networks learned consistently better hidden representations that the one hidden layer network, although the effect was small (figure 9).

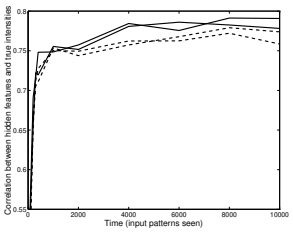

Figure 9: Two runs each of networks with a top layer hidden unit (solid lines) and without one (dashed lines).

One sub-network can teach another

The aim of the last experiment was to see if connecting a network that had already learned to represent disparity from one part of the image to a second network with random weights would result in faster learning in the second network. Using the double-network architecture in figure 6 we took a Network A that had learned disparity (the teacher) and attached a Network B with random weights (the student), and vice versa. In both cases, the results suggest that the student network greatly benefits from having the top unit in common with the teacher network (figure 10). Both the student and the combined network rapidly improve their top level representation of global disparity.

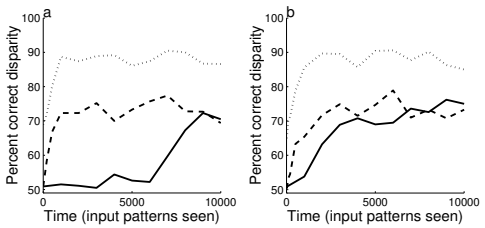

Figure 10: The percent correct disparity for the double-network when presented with both left and right images (dotted line), when presented only with the image to the student while leaving visible units to the teacher unclamped (dashed line), when no teacher was present (solid line). In **a** the left network was the teacher and the right network was the student and in **b** it is vice versa. Note that only 2 out of 6 of the student-alone networks ever discovered depth; we show the two that did.

Acknowledgements

We thank P. Dayan, B. Frey, D. MacKay, R. Neal and M. Revow for numerous insights. The research was funded by grants from the Canadian Natural Science and Engineering Research Council and the Ontario Information Technology Research Center. GEH is the Nesbitt-Burns fellow of the Canadian Institute for Advanced Research.

References

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

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

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

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The Helmholtz machine. *Neur. Comput.*, 7:889–904.

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

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

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pat. Anal. and Mach. Intell.*, 6:721–741.

Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161.

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

Hinton, G. E. and Sejnowski, T. J. (1983). Optimal perceptual Inference. In *Proc. of the IEEE Comp. Soc. Conf. on Comp. Vis. and Pat. Recogn.*, 448–453. Washington, DC.

Hinton, G. E. and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Vol. 1*. MIT Press, Cambridge, MA.

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

Lewicki, M. S. and Sejnowski, T. J. (1997). Bayesian unsupervised learning of higher order structure. In Mozer, M., Jordan, M., and Petsche, T., editors, *Adv. in Neur. Inf. Proc. Syst. 9*. MIT Press, Cambridge, MA.

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

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

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

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

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