In *Neural Computation* 5(2): 267-277, 1993.

Learning mixture models of spatial coherence

Suzanna Becker and Geoffrey E. Hinton
Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 1A4

August 24, 1995

Abstract

We have previously described an unsupervised learning procedure that discovers spatially coherent properties of the world by maximizing the information that parameters extracted from different parts of the sensory input convey about some common underlying cause. When given random dot stereograms of curved surfaces, this procedure learns to extract surface depth because that is the property that is coherent across space. It also learns how to interpolate the depth at one location from the depths at nearby locations (Becker and Hinton, 1992b). In this paper, we propose two new models which handle surfaces with discontinuities. The first model attempts to detect cases of discontinuities and reject them. The second model develops a mixture of expert interpolators. It learns to detect the locations of discontinuities and to invoke specialized, asymmetric interpolators that do not cross the discontinuities.

1 Introduction

Standard backpropagation is implausible as a model of perceptual learning because it requires an external teacher to specify the desired output of the network. We have shown (Becker and Hinton, 1992b) how the external teacher can be replaced by internally derived teaching signals. These signals are generated by using the assumption that different parts of the perceptual input have common causes in the external world. Small modules that look at separate but related parts of the perceptual input discover these common causes by striving to produce outputs that agree with each other (see Figure 1a). The modules may look at different modalities (e.g. vision and touch), or the same modality at different times (e.g. the consecutive 2-D views of a rotating 3-D object), or even spatially adjacent parts of the same image.

In previous work, we showed that when our learning procedure is applied to adjacent patches of images, it allows a neural network that has no prior knowledge of depth to discover stereo disparity in random dot stereograms of curved surfaces. A more general version of the method allows the network to discover the best way of interpolating the depth at one location from the depths at nearby locations. We first summarize this earlier work, and then introduce two new models which allow coherent predictions to be made in the presence of discontinuities. The first assumes a model of the world in which patterns are drawn from two possible classes: one which can be captured by a simple model of coherence, and one which is unpredictable. This allows the network to reject cases containing discontinuities. The second method allows the network to develop multiple models of coherence, by learning a mixture of depth interpolators for curved surfaces with discontinuities. Rather than rejecting cases containing discontinuities, the network develops a set of location-specific discontinuity detectors, and appropriate interpolators for each class of discontinuities. An alternative way of learning the same representation for this problem,

using an unsupervised version of the competing experts algorithm described by Jacobs, Jordan, Nowlan and Hinton (1991), is described in (Becker and Hinton, 1992a).

2 Learning spatially coherent features in images

Using a modular architecture as shown in Figure 1a), a network can learn to model a spatially coherent surface, by extracting mutually predictable features from neighboring image patches. The goal of the learning is to produce good agreement between the outputs of modules which receive input from neighboring patches. The simplest way to get the outputs of two modules to agree is to use the squared difference between the outputs as a cost function, and to adjust the weights in each module so as to minimize this cost. Unfortunately, this usually causes each module to produce the same constant output that is unaffected by the input to the module and therefore conveys no information about it. We would like the outputs of two modules to agree closely (i.e. to have a small expected squared difference) *relative* to how much they both vary as the input is varied. When this happens, the two modules must be responding to something that is common to their two inputs. In the special case when the outputs, d_a, d_b, of the two modules are scalars, a good measure of agreement is:

$$I = 0.5 \log \frac{V(d_a + d_b)}{V(d_a - d_b)} \tag{1}$$

where V is the variance over the training cases. Under the assumption that d_a and d_b are both versions of the same underlying Gaussian signal that have been corrupted by independent Gaussian noise, it can be shown that I is the mutual information (Shannon and Weaver, 1964) between the underlying signal and the average of d_a and d_b. By maximizing I we force the two modules to extract as pure a version as possible of the underlying common signal.

Insert figure 1 about here

2.1 The basic stereo net

We have shown how this principle can be applied to a multi-layer network that learns to extract depth from random dot stereograms (Becker and Hinton, 1992b). Each network module received input from a patch of a left image and a corresponding patch of a right image, as shown in Figure 1a). Adjacent modules received input from adjacent stereo image patches, and learned to extract depth by trying to maximize agreement between their outputs. The real-valued depth (relative to the plane of fixation) of each patch of the surface gives rise to a disparity between features in the left and right images; since that disparity is the only property that is coherent across each stereo image, the output units of modules were able to learn to accurately detect relative depth.

2.2 The interpolating net

The basic stereo net uses a very simple model of coherence in which an underlying parameter at one location is assumed to be approximately equal to the parameter at a neighboring location. This model is fine for the depth of fronto-parallel surfaces but it is far from the best model of slanted or curved surfaces. Fortunately, we can use a far more general model of coherence in which the parameter at one location is assumed to be an unknown linear function of the parameters at nearby locations. The particular linear function that is appropriate can be learned by the network.

We used a network of the type shown in Figure 1b). The depth computed locally by a module, d_c, was compared with the depth predicted by a linear combination $\hat{d}_c$ of the outputs of nearby modules, and the network tried to maximize the agreement between d_c and $\hat{d}_c$.

The contextual prediction, $\hat{d}_c$, was produced by computing a weighted sum of the outputs of *two* adjacent modules on either side. The interpolating weights used in this sum, and all other weights in the network, were adjusted so as to maximize agreement between locally computed and contextually predicted depths. To speed the learning, we first trained the lower layers of the network as before, so that agreement was maximized between neighboring locally computed outputs. This made it easier to learn good interpolating weights. When the network was trained on stereograms of cubic surfaces, it learned interpolating weights of $-0.147, 0.675, 0.656, -0.131$ (Becker and Hinton, 1992b). Given noise free estimates of local depth, the optimal linear interpolator for a cubic surface is $-0.167, 0.667, 0.667, -0.167$.

3 Mixture models of coherence

The models described above were based on the assumption of a single type of coherence in images. We assumed there was some parameter of the image which was either constant for nearby patches, or varied smoothly across space. In natural scenes, these simple models of coherence may not always hold. There may be widely varying amounts of curvature, from smooth surfaces, to highly curved spherical or cylindrical objects. There may be coherent structure at several spatial scales; for example, a rough surface like a brick wall is highly convoluted at a fine spatial scale, while at a coarser scale it is planar. And at boundaries between objects, or between different parts of the same object, there will be discontinuities in coherence. It would be better to have multiple models of coherence, which could account for a wider range of surfaces. One way to handle multiple models is to have a mixture of distributions (McLachlan and Basford, 1988). In this section, we introduce a new way of employing mixture models to account for a greater variety of situations. We extend the learning procedure described in the previous section based on these models.

3.1 Throwing out discontinuities

If the surface is continuous, the depth at one patch can be accurately predicted from the depths of two patches on either side. If, however, the training data contains cases in which there are depth discontinuities (see Figure 2) the interpolator will also try to model these cases and this will contribute considerable noise to the interpolating weights and to the depth estimates. One way of reducing this noise is to treat the discontinuity cases as outliers and to throw them out. Rather than making a hard decision about whether a case is an outlier, we make a soft decision by using a mixture model. For each training case, the network compares the locally extracted depth, d_c, with the depth predicted from the nearby context, $\hat{d}_c$. It assumes that $d_c - \hat{d}_c$ is drawn from a zero-mean Gaussian if it is a continuity case and from a uniform distribution if it is a discontinuity case, as shown in Figure 3. It can then estimate the probability of a continuity case:

$$p_{cont}(d_c - \hat{d}_c) = \frac{N(d_c - \hat{d}_c, 0, \hat{V}_{cont}(d_c - \hat{d}_c))}{N(d_c - \hat{d}_c, 0, \hat{V}_{cont}(d_c - \hat{d}_c)) + k_{discont}} \tag{2}$$

where N is a gaussian, and $k_{discont}$ is a constant representing a uniform density.[1]

[1] We empirically select a good (fixed) value of $k_{discont}$, and we choose a starting value of $\hat{V}_{cont}(d_c - \hat{d}_c)$ (some proportion of the initial variance of $d_c - \hat{d}_c$), and gradually shrink it during learning. The learning algorithm's

---

Insert figure 2 about here

Insert figure 3 about here

We can now optimize the *average* information d_c and $\hat{d}_c$ transmit about their common cause. We assume that no information is transmitted in discontinuity cases, so the average information depends on the probability of continuity and on the variance of $d_c + \hat{d}_c$ and $d_c - \hat{d}_c$ measured only in the continuity cases:

$$I^* = 0.5 \;\; P_{cont} \;\; \log \frac{V_{cont}(d_c + \hat{d}_c)}{V_{cont}(d_c - \hat{d}_c)} \tag{3}$$

where $P_{cont} = \left\langle p_{cont}(d_c - \hat{d}_c) \right\rangle$.

We tried several variations of this mixture approach. The network is quite good at rejecting the discontinuity cases, but this leads to only a modest improvement in the performance of the interpolator. In cases where there is a depth discontinuity between d_a and d_b or between d_d and d_e the interpolator works moderately well because the weights on d_a or d_e are small. Because of the term P_{cont} in equation 3 there is pressure to include these cases as continuity cases, so they probably contribute noise to the interpolating weights. In the next section we show how to avoid making a forced choice between rejecting these cases or treating them just like all the other continuity cases.

3.2 Learning a mixture of interpolators

The presence of a depth discontinuity somewhere within a strip of five adjacent patches does not necessarily destroy the predictability of depth across these patches. It may just restrict the range over which a prediction can be made. So instead of throwing out cases that contain a discontinuity, the network could try to develop a number of different, specialized models of spatial coherence across several image patches. If, for example, there is a depth discontinuity between d_c and d_e in Figure 1 b), an extrapolator with weights of $-1.0, +2.0, 0, 0$ would be an appropriate predictor of d_c. The network could also try to detect the locations of discontinuities, and use this information as the basis for deciding which model to apply on a given case. This information is useful not only in making clean decisions about which coherence model to apply, but it also provides valuable cues for interpreting the scene by indicating the locations of object boundaries in the image. Thus, we can use the both the interpolated depth map, as well as the locations of depth discontinuities, in subsequent stages of scene interpretation.

A network can learn to discover multiple coherence models using a set of competing interpolators. Each interpolator tries, as before, to achieve high agreement between its output and the depth extracted locally by a module. Additionally, each interpolator tries to account for as

performance is fairly robust with respect to variations in the choice of $k_{discont}$; the main effect of changing this parameter is to sharpen or flatten the network's probabilitstic decision function for labelling cases as continuous or discontinuous (equation 2). The choice of $\hat{V}_{cont}(d_c - \hat{d}_c)$, on the other hand, turns out to affect the learning algorithm more critically; if this variance is too small, many cases will be treated as discontinuous, and the network may converge to very large weights which overfit only a small subset of the training cases. There is no problem, however, if this variance is too large initially; in this case, all patterns are treated as continuous, and as the variance is shrunk during learning, some discontinuous cases are eventually detected.

many cases as possible by maximizing the probability that its model holds. The objective function maximized by the network is the sum over models, i, of the agreement between the output of the ith model, $\hat{d}_{ic}$, and the predicted depth, d_c, weighted by the probability of the ith model:

$$I^{**} = \sum_i \langle p_i \rangle \log \frac{V^i(\hat{d}_{ic} + d_c)}{V^i(\hat{d}_{ic} - d_c)} \tag{4}$$

where the V^is represent variances given that the ith model holds. The probability that the ith model is applicable on each case α, p_i^α, can be computed independently of how well the interpolators are doing;[2] this can be done by adding extra "controller" units to the network, as shown in Figure 4, whose sole purpose is to compute the probability, p_i, that each interpolator's model holds. The weights of both the controllers and the interpolating experts can be learned simultaneously, so as to maximize I^{**}. By assigning a controller to each expert interpolator, each controller should learn to detect a discontinuity at a particular location (or the absence of a discontinuity in the case of the interpolator for pure continuity cases). And each interpolating unit should learn to capture the particular type of coherence that remains in the presence of a discontinuity at a particular location.

Insert figure 4 about here

The outputs of the controllers are normalized, so that they represent a probability distribution over the interpolating experts' models. We can think of these normalized outputs as the probability with which the system selects a particular expert. Each controller's output is a normalized exponential function of its *squared* total input, x_i:

$$p_i = \frac{e^{{x_i}^2/T\ \hat{\sigma}(x_i)^2}}{\sum_j e^{{x_j}^2/T\ \hat{\sigma}(x_j)^2}} \tag{5}$$

Squaring the total input makes it possible for each unit to detect a depth edge at a particular location, independently of the direction of contrast change. We normalize the squared total input in the exponential by an estimate of its variance, $\hat{\sigma}(x_j)^2 = k \sum_{ji} {w_{ji}}^2$. (This estimate of the variance of the total weighted input is exact if the unweighted individual inputs are independent, Gaussian, and have equal variances of size k.) This discourages any one unit from trying to model all of the cases simply by having huge weights. The controllers get to see all five local depth estimates, $d_a \ldots d_e$. As before, each interpolating expert computes a linear function of four contextually extracted depths, $\hat{d}_{ic} = w_{ia} d_a + w_{ib} d_b + w_{id} d_d + w_{ie} d_e$, in order to try to predict the centrally extracted depth d_c.

We first trained the network using the original continuous model, as described in Section 2, on a training set of 1000 images with discontinuities, until the lower layers of the network became well tuned to depth. So the interpolators were initially pretrained using the continuity model, and all the interpolators learned similar weights. We then froze the weights in the lower layers, added a small amount of noise to the interpolators' weights (uniform in $[-0.1, 0.1]$), and applied the mixture model to improve the interpolators and train the controller units. We ran the learning

[2] More precisely, this computed probability is *conditionally independent* of the interpolators' performance on a particular case, with independence being conditioned upon a fixed set of weights. As the reviewer has pointed out, when the weights change over the course of learning, there is an interdependence between the probabilities and interpolated quantities via the shared objective function.

procedure for ten runs, each run starting from different random initial weights and proceeding for 10 conjugate gradient learning iterations. The network learned similar solutions in each case.

A typical set of weights on one run is shown in Figure 5. The graph on the right in this figure shows that four of the controller units are tuned to discontinuities at different locations. The weights for the first interpolator (shown in the top left) are nearly symmetrical, and the corresponding controller's weights (shown immediately to the right) are very small; the graph on the right shows that this controller (shown as a solid line plot) mainly responds in cases when there is no discontinuity. The second interpolator (shown in the left column, second from the top) predominantly uses the leftmost three depths; the corresponding controller for this interpolator (immediately right of the top left interpolator's weights) detects discontinuities between the rightmost two depths, d_c and d_d. Similarly, the remaining controllers detect discontinuities to the right or left of d_c; each controller's corresponding interpolator uses the depths on the opposite side of the discontinuity to predict d_c.

Insert figure 5 about here

---

4 Discussion

We have described two ways of modelling spatially coherent features in images of scenes with discontinuities. The first approach was to simply try to discriminate between patterns with and without discontinuities, and throw away the former. In theory, this approach is promising, as it provides a way of making the algorithm more robust against outlying data points. We then applied the idea of multiple models of coherence to a set of interpolating units, again using images of curved surfaces with discontinuities. The competing controllers in Figure 4 learned to explicitly represent which regularity applies in a particular region. The output of the controllers was used to compute a probability distribution over the various competing models of coherence.

The representation learned by this network has a number of advantages. We now have a measure of the probability that there is a discontinuity which is independent of the prediction error of the interpolator. So we can tell how much to trust each interpolator's estimate on each case. It should be possible to distinguish clear cases of discontinuities from cases which are simply noisy, by the entropy of the controllers' outputs. Furthermore, the controller outputs tell us not only that a discontinuity is present, but exactly where it lies. This information is important for segmenting scenes, and should be a useful representation for later stages of unsupervised learning. Like the raw depth estimates, the location of depth edges should exhibit coherence across space, at larger spatial scales. It should therefore be possible to apply the same algorithm recursively to the the outputs of the controllers, to find object boundaries in two-dimensional stereo images.

The approach presented here should be applicable to other domains which contain a mixture of alternative local regularities across space or time. For example, a rigid shape causes a linear constraint between the locations of its parts in an image, so if there are many possible shapes, there are many alternative local regularities (Zemel and Hinton, 1991).

5 Acknowledgments

This research was funded by grants from NSERC and the Ontario Information Technology Research Centre. Hinton is Noranda fellow of the Canadian Institute for Advanced Research. Thanks

to John Bridle and Steve Nowlan for helpful discussions.

References

Becker, S. and Hinton, G. E. (1992a). Learning to make coherent predictions in domains with discontinuities. In *Advances In Neural Information Processing Systems 4*. Morgan Kaufmann Publishers.

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

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1).

McLachlan, G. J. and Basford, K. E. (1988). *Mixture models: inference and applications to clustering*. Marcel Dekker, Inc.

Shannon, C. E. and Weaver, W. (1964). *The Mathematical Theory Of Communication*. The Univeristy Of Illinois Press.

Zemel, R. S. and Hinton, G. E. (1991). Discovering viewpoint-invariant relationships that characterize objects. In *Advances In Neural Information Processing Systems 3*, pages 299–305. Morgan Kaufmann Publishers.

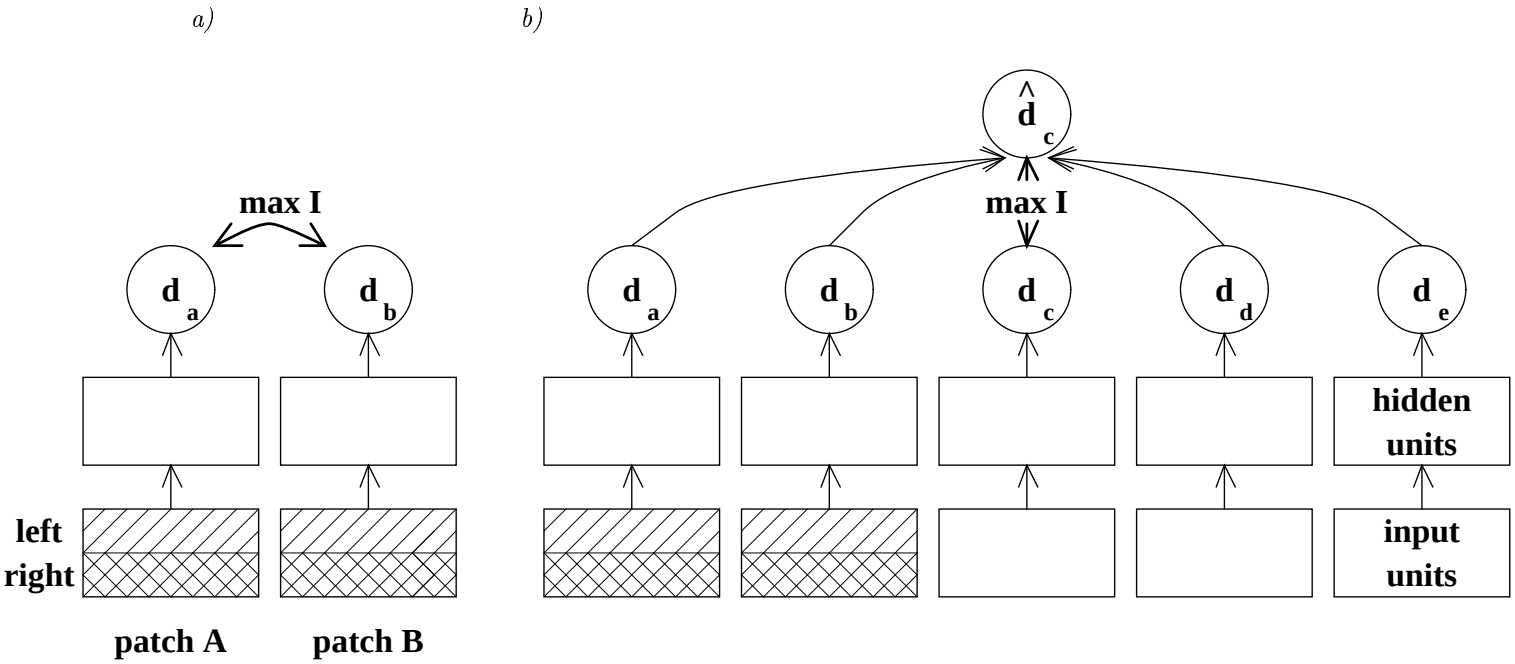

Figure 1: *a) Two modules that receive input from corresponding parts of stereo images. The first module receives input from stereo patch A, consisting of a horizontal strip from the left image (striped) and a corresponding strip from the right image (hatched). The second module receives input from an adjacent stereo patch B. The modules try to make their outputs, d_a and d_b, convey as much information as possible about some underlying signal (i.e., the depth) which is common to both patches. b) The architecture of the interpolating network, consisting of multiple copies of modules like those in a) plus a layer of interpolating units. The network tries to maximize the information that the locally extracted parameter d_c and the contextually predicted parameter $\hat{d}_c$ convey about some common underlying signal. We actually used 10 modules and the central 6 modules tried to maximize agreement between their outputs and contextually predicted values. We used weight averaging to constrain the interpolating function to be identical for all modules.*

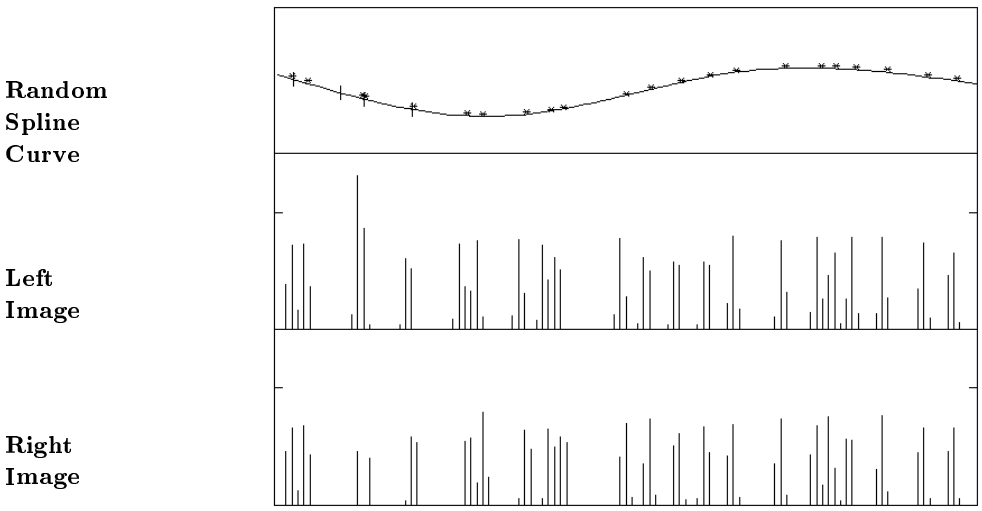

Figure 2: **Top:** *A curved surface strip with a discontinuity created by fitting 2 cubic splines through randomly chosen control points, 25 pixels apart, separated by a depth discontinuity. Feature points are randomly scattered on each spline with an average of* 0.22 *features per pixel.* **Bottom:** *A stereo pair of "intensity" images of the surface strip formed by taking two different projections of the feature points, filtering them through a gaussian, and sampling the filtered projections at evenly spaced sample points. The sample values in corresponding patches of the two images are used as the inputs to a module. The depth of the surface for a particular image region is directly related to the disparity between corresponding features in the left and right patch. Disparity ranges continuously from* -1 *to* $+1$ *image pixels. Each stereo image was 120 pixels wide and divided into 10 receptive fields 10 pixels wide and separated by 2 pixel gaps, as input for the networks shown in Figure 1. The receptive field of an interpolating unit spanned 58 image pixels, and discontinuities were randomly located a minimum of 40 pixels apart, so only rarely would more than one discontinuity lie within an interpolator's receptive field.*

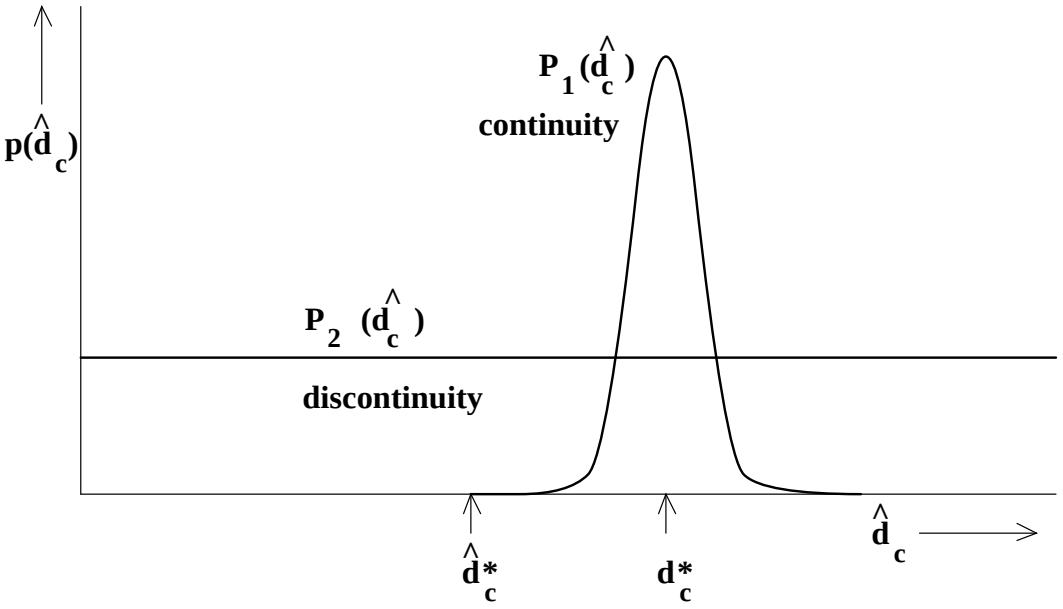

Figure 3: *The probability distribution of $\hat{d}_c$, $P_1(\hat{d}_c)$, is modeled as a mixture of two distributions: a Gaussian with mean = d_c^* and small variance, and $P_2(\hat{d}_c)$, a uniform distribution. Sample points for $\hat{d}_c$ and d_c, $\hat{d}_c^*$ and d_c^* are shown. In this case, $\hat{d}_c^*$ and d_c^* are far apart so $\hat{d}_c^*$ is more likely to have been drawn from P_2.*

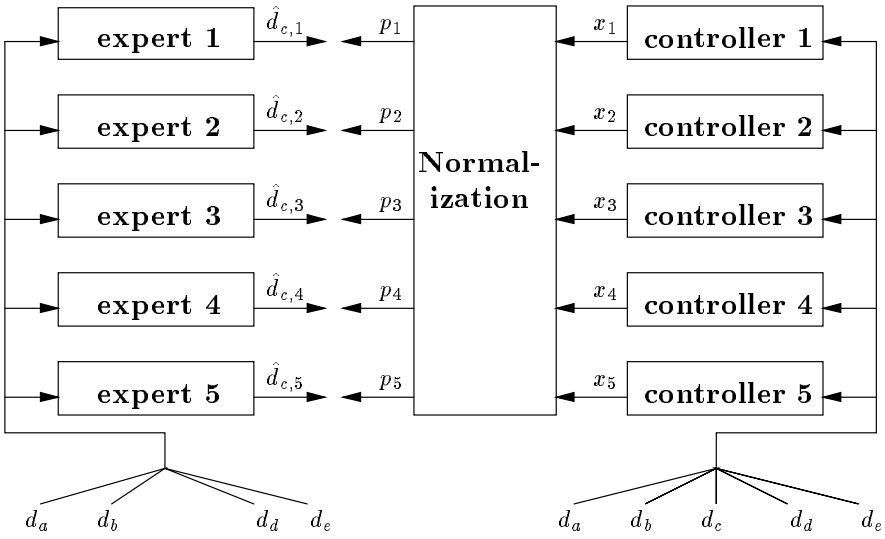

Figure 4: *An architecture for learning a mixture model of curved surfaces with discontinuities, consisting of a set of interpolators and discontinuity detectors. We actually used a larger modular network and equality constraints between the weights of corresponding units in different modules, with 6 copies of the architecture shown here. Each copy received input from different but overlapping parts of the input.*

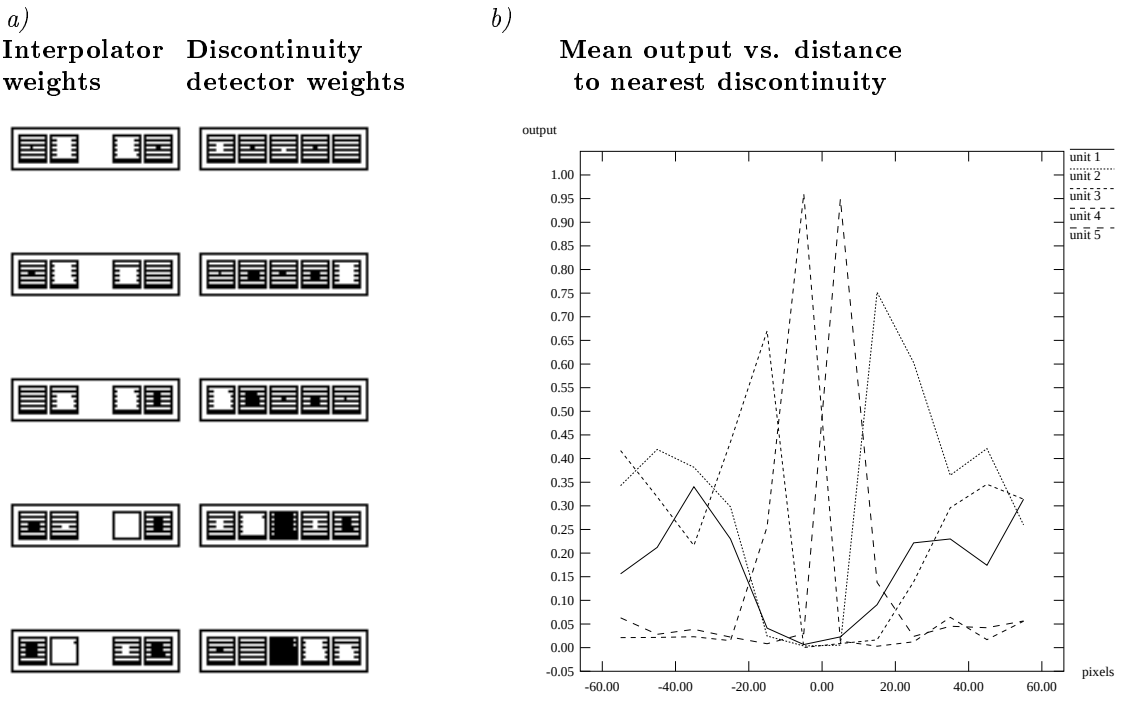

Figure 5: *a) Typical weights learned by the five competing interpolators and corresponding five discontinuity detectors. Positive weights are shown in white, and negative weights in black. b)The mean probabilities computed by each discontinuity detector are plotted against the the distance from the center of the units' receptive field to the nearest discontinuity. The probabilistic outputs are averaged over an ensemble of 1000 test cases. If the nearest discontinuity is beyond ± thirty pixels, it is outside the units' receptive field and the case is therefore a continuity example.*