Network: Comput. Neural Syst. **8** (1997) 441–452. Printed in the UK PII: S0954-898X(97)79986-X

Unsupervised discovery of invariances

Stephen Eglen†, Alistair Bray‡ and Jim Stone§
School of Cognitive and Computing Sciences, University of Sussex, Brighton BN1 9QH, UK

Received 2 December 1996, in final form 26 June 1997

Abstract. The grey level profiles of adjacent image regions tend to be different, whilst the 'hidden' physical parameters associated with these regions (e.g. surface depth, edge orientation) tend to have similar values. We demonstrate that a network in which adjacent units receive inputs from adjacent image regions learns to code for hidden parameters. The learning rule takes advantage of the spatial smoothness of physical parameters in general to discover particular parameters embedded in grey level profiles which vary rapidly across an input image. We provide examples in which networks discover stereo disparity and feature orientation as invariances underlying image data.

1. Introduction

A crucial requirement for an intelligent system operating in a complex environment is that it can 'see the wood for the trees', i.e. it can determine the significant 'hidden' parameters underlying large streams of confusing input data. This problem is confronted by a child learning the regularities underlying linguistic experience, or a perceptual system extracting invariant properties of the world from the changing phenomena of sensory experience. In this paper we describe how an unsupervised network of artificial neurons can solve this problem by using a learning rule which exploits fundamental constraints of the physical world. Essentially, this learning rule forces units to modify their connection weights so that their outputs vary in the same manner as useful descriptors of the world.

Using this approach, previous work (Stone and Bray 1995, Stone 1996a, b) has described a temporal learning rule based upon a general assumption; namely that inputs to sensory receptors tend to change rapidly over time, whereas the physical parameters underlying these changes vary more slowly, and therefore more smoothly. Hence, if a neuron codes for a physical parameter then its output should also change slowly and smoothly, despite its rapidly fluctuating inputs. Thus a neuron that adapts to make its output vary smoothly over time can learn to code the invariances underlying its input.

Such a rule captures the temporal smoothness constraint. However, another equally valid constraint is *spatial* smoothness, i.e. physical parameters tend to vary smoothly across space. When dealing with the temporal constraint one needs only to consider a single neuron and its output over time. However, in the spatial case one needs to consider a network of neurons arranged spatially such that neighbouring output neurons receive inputs

† Present address: Centre for Cognitive Science, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, UK. E-mail: stephen@cns.ed.ac.uk.
‡ E-mail: alib@cogs.susx.ac.uk.
§ Present address: Psychology Department, Sheffield University, Sheffield, S10 2TN, UK. E-mail: stone@shef.ac.uk.

0954-898X/97/040441+12$19.50

from neighbouring regions of space. We can then interpret the spatial smoothness constraint as follows: if a network of neurons codes for a physical parameter then its output should change slowly and smoothly across the network, despite the quite different inputs received by neighbouring neurons.

The temporal constraint concerns the output of a single unit over many time steps (Stone and Bray 1995, Stone 1996a, b), whereas the spatial constraint deals with the output of many units at a single time step. This paper demonstrates how this spatial constraint is sufficiently powerful for a network to 'discover' spatial disparity varying smoothly in random-dot stereograms, and smooth changes of oriented features in a textured image.

1.1. The approach

Much recent research into brain function has been influenced by the finding that the micro-architecture of neocortex is uniform across cortical areas (Douglas and Martin 1991). That is, neocortex consists of a large number of inter-connected modules which have essentially similar structure, but each module processes a different type of input. Such modules have been coined 'canonical microcircuits' (Douglas and Martin 1991). If we define such a microcircuit in terms of an architecture (which describes how a collection of excitatory and inhibitory cells may connect to one another) and a set of learning rules (which describe how these connections may change), it is apparent that simple microcircuits can compute highly complex functions of their inputs. Our work suggests a complementary strategy to investigating these microcircuits directly, i.e. to discover 'canonical microfunctions' (Stone 1996a). We propose that canonical microcircuits may have evolved to maximize relatively simple functions of their outputs (e.g. spatial smoothness). We suggest that through optimizing such simple functions a network can discover complex and varied representations for coding redundant data, and that determining the microfunction may lead more easily to a microcircuit for optimizing that function (rather than *vice versa*).

In the light of this we can consider other work for extracting invariances. Földiák (1991) demonstrated how a Hebbian learning rule that exploited short-term temporal correlations in the inputs of a unit could learn shift invariance. This learning rule was amongst the first to emphasize temporal correlation between units. However, the emphasis of the work lay heavily upon a specific learning rule for detecting oriented lines, where sets of lines at different orientations were represented by linearly separable clusters of input vectors (Oram and Földiák 1996, Földiák 1991). In a similar manner, Schraudolph and Sejnowski (1992) developed a learning rule for extracting disparity from stereo data (one of the tasks used in our work). In both of these above-mentioned models the emphasis was to provide a learning rule and architecture to extract one particular invariance, rather than to elucidate a general merit function which might have more general applicability.

The strategy of maximizing a general function has been adopted in two notable papers (Linsker 1988, Becker and Hinton 1992). Linsker (1988) demonstrated that, under certain conditions, a Hebbian learning rule which adjusts the weights of a unit to maximize its output variance over time will also maximize its information rate, subject to a normalization constraint. In our model, we ensure that our network learns 'interesting' physical parameters by maximizing the long-range variance (the long-term variance in our temporal model (Stone and Bray 1995)). However, whereas Linsker's work does not consider short-term, or spatially localized correlations, we also seek to minimize a measure of local variance. This has the critical effect of introducing a smoothness constraint.

The closest model to our own is the IMAX model (Becker and Hinton 1992) which maximizes the mutual information between units of two adjacent networks to learn stereo

disparity. They used a merit function which minimized the variance of the difference in activity of neighbouring output units. To prevent all output units producing zero activity, the merit function also enforced a constraint to maximize the variance of output units. However, using an architecture similar to that described here, pre-learning of the hidden layer was necessary for the system to learn successfully. In comparison, our model requires no pre-learning of the hidden layer and recasts the merit function in terms of minimizing short-range variances and maximizing long-range variances.

Following on with a similar approach to Becker and Hinton, Phillips *et al* (1995) emphasized how different sensory modalities, or streams within modalities, often signal aspects of the input which are correlated (e.g. the sound and sight of a word being spoken). The merit function from the IMAX model is extended to include the three-way mutual information between the receptive field, contextual field, and network output. When this merit function is maximized using gradient ascent, they demonstrate that the correlations between different streams of synthetic data can be used to discover the features which are related across streams, as well as discovering relations between them.

2. General method

Firstly, we define a multi-layer network of units in which neighbouring output units have neighbouring, non-overlapping 'receptive fields', i.e. they receive inputs from neighbouring regions of an input image. We then present an image to the network, where the image grey-level is a function of a 'hidden' parameter which varies smoothly across the image. Finally, we adjust the network weights so as to maximize the variance of the set of outputs units at a given time step whilst simultaneously minimizing the local variance of neighbouring units. This ensures that, at each time step, nearby units have similar outputs, and widely separated units have very different outputs. When the weights stabilize, the network has discovered some smooth function of the image which corresponds to the hidden parameter. We can confirm that the network has discovered the hidden parameter by presenting the network with a previously unseen image containing the hidden parameter, and checking that the network outputs correlate with the parameter values. The remainder of this section describes the precise merit function that is maximized, the method used for gradient ascent, and the general architecture of the network.

2.1. Function maximization

The temporal learning rule (Stone and Bray 1995, Stone 1996a, b) has been adapted slightly for the spatial model, although it remains essentially the same. In the temporal model there was a single output unit. We jointly maximized the long-term variance of the output V of the unit and minimized its short-term variance U by maximizing the merit function $F = \log(V/U)$ (the logarithm was used so that the derivative of F was easy to compute). The variances V and U were calculated using

$$V = \sum_{t=1}^{T} (\bar{z}_t - z_t)^2 \qquad \text{and} \qquad U = \sum_{t=1}^{T} (\tilde{z}_t - z_t)^2$$

where z_t was the output of the unit at a particular time, $\bar{z}_t$ was its long-term weighted average, and $\tilde{z}_t$ was its short-term weighted average (averaging over time). In the current spatial model we also maximize a function $F = \log(V/U)$; however, this time the long-

and short-term variances are measured as spatial variances:

$$V = \sum_{i,j} (\bar{z}_{i,j} - z_{i,j})^2 \qquad \text{and} \qquad U = \sum_{i,j} (\tilde{z}_{i,j} - z_{i,j})^2$$

where i, j are spatial indices spanning the whole output array. The quantities $\bar{z}_{i,j}$ and $\tilde{z}_{i,j}$ represent long- and short-range exponentially weighted spatial means of z, where the weighting is centred at position i, j, and the means are calculated by convolving the array of output activities with exponentially decaying masks $\bar{\Phi}$ and $\tilde{\Phi}$ of different half-lives:

$$\bar{z}_{i,j} = \sum_{x,y=-\infty}^{x,y=\infty} \bar{\Phi}_{x,y} z_{i+x,j+y} \qquad \text{and} \qquad \tilde{z}_{i,j} = \sum_{x,y=-\infty}^{x,y=\infty} \tilde{\Phi}_{x,y} z_{i+x,j+y}.$$

Therefore, the main difference between the spatial learning rule and the temporal learning rule is the source of contributions to $\tilde{z}$ and $\bar{z}$. Whereas the temporal version sums outputs of a single unit over time, the spatial version sums the output of units in a spatial array at a single time step. As before (Stone 1996a, b), we chose to maximize the function using the iterative method of conjugate gradient ascent. The error term $\delta_{i,j}$ for each output unit is defined as $\delta_{i,j} = \partial F / \partial a_{i,j}$, where $a_{i,j}$ is the total input to the output unit (i, j) at a given time. This can be reduced (see Eglen *et al* (1996) for full details of the derivation) to

$$\delta_{i,j} = \frac{1}{V} \sum_{x,y=-\infty}^{x,y=\infty} (\bar{z}_{i+x,j+y} - z_{i+x,j+y}) \bar{\Phi}'_{x,y} - \frac{1}{U} \sum_{x,y=-\infty}^{x,y=\infty} (\tilde{z}_{i+x,j+y} - z_{i+x,j+y}) \tilde{\Phi}'_{x,y}$$

where

$$\bar{\Phi}'_{x,y} = \begin{cases} \bar{\Phi}_{x,y} - 1 & \text{if } x, y = 0 \\ \bar{\Phi}_{x,y} & \text{otherwise} \end{cases} \qquad \tilde{\Phi}'_{x,y} = \begin{cases} \tilde{\Phi}_{x,y} - 1 & \text{if } x, y = 0 \\ \tilde{\Phi}_{x,y} & \text{otherwise} \end{cases}$$

so that $\bar{\Phi}'$ and $\tilde{\Phi}'$ are identical to the masks $\bar{\Phi}$ and $\tilde{\Phi}$, except that the central element of these masks is reduced by 1. These error terms are back propagated to hidden layer units using the chain rule (Rumelhart *et al* 1986). For each weight w_{uv}, from unit u to unit v, the weight change is $\Delta w_{uv} = \delta_v z_u$, where z_u is the activity of unit u.

2.2. Network architecture

The network exploits weight sharing. That is, there is only one small micro-network which is notionally replicated many times in an array to form a *virtual* network; this virtual network spans the entire image (see figure 1). Weights exist for the micro-network only, and all weight-changes in the virtual network therefore take effect in the micro-network. Hence the size of the micro-network determines the degrees of freedom of the virtual network. As an example, a micro-network might receive inputs from a 5×5 region of image, have a layer of nine hidden units (each connecting to all the inputs), and a single output unit connecting to all hidden units. If each micro-network 'sees' adjacent non-overlapping regions of image, we can construct a virtual network of 120×120 output units which completely covers a 600×600 input image.

Weight sharing is used to keep the network small for computational reasons. The equivalent network without weight sharing would need many more weights and input examples, resulting in much longer learning times and a need for more computational resources. The free parameters in this model therefore determine the architecture of the micro- and virtual network, and the extent of spatial averaging. Details of these parameters are given in the figure captions.

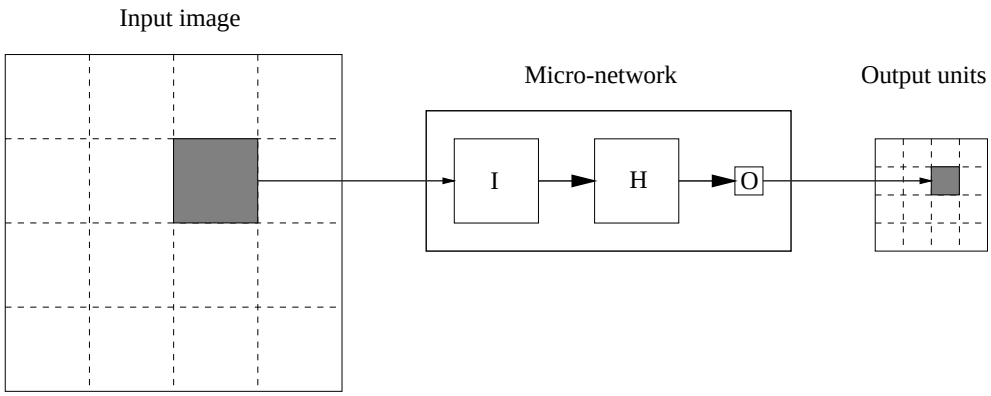


Figure 1. *General network definition.* The weight-sharing scheme is implemented using a micro-network. The input image is divided into patches, where each patch is the size of the input layer of the micro-network. The activation of each output unit in the virtual output array is defined to be the output from the micro-network when presented with the corresponding input patch. The micro-network is a standard feed-forward, fully connected network with an input (I), hidden (H) and output (O) layer. All units in the hidden layer have a bias weight connecting to a notional input which always has value 1. The activity of hidden units is calculated by taking the inner product of inputs and weights, and passing it through a tanh function, whereas the activity of output units is simply the inner product of inputs and weights. Weights are initialized at random to $\pm(1/\sqrt{2n})\log r$ where r is a random number chosen uniformly from $(0, 1)$ and n is the number of weights in the virtual network (Williams 1995). For conjugate gradient ascent we have used the parameters suggested in Williams (1991).

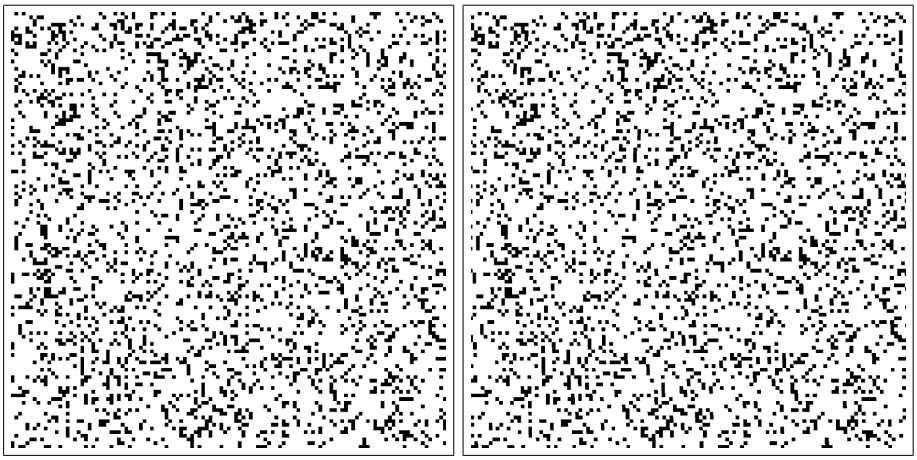
3. Discovering invariances

Here we demonstrate this general unsupervised algorithm discovering two quite different invariances, given two quite different images. The only significant difference between the two demonstrations is the inputs they receive.

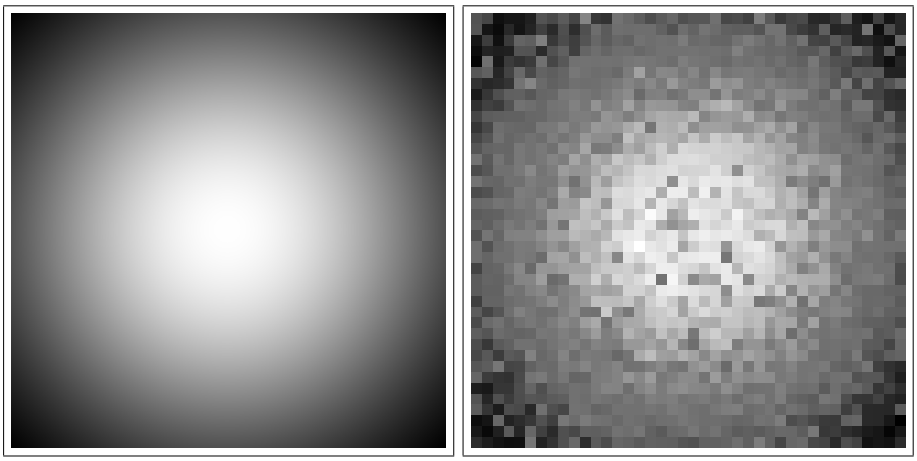
3.1. Demonstration 1: disparity

The spatial disparity between images to the left and right eye provides information concerning depth. Spatial disparity and depth both tend to vary smoothly in the world across slanted surfaces. Spatial smoothness has been used before (Becker and Hinton 1992) as a constraint for algorithms computing disparity estimates.

We created a pair of random-dot stereograms with sub-pixel disparities, for which the disparity was a Gaussian function of distance from the image centre (see figure 2(*a*)). Thus the disparity values varied smoothly across the image. We constructed a virtual network (as described above) such that each micro-network received two sets of inputs: one from a small region of the left image, and the other from the corresponding region of the right image (see the caption for details). We then maximized the merit function $\log(V/U)$ over the virtual network. When the weights had stabilized, the outputs of the units correlated highly ($|r| > 0.92$) with disparity (see figure 2(*b*)). We then tested the network on a new random-dot stereogram and found that the correlation between the network output and disparity was maintained ($|r| > 0.89$) (see figure 2(*c*)). The network had therefore 'discovered' that computing disparity from the random-dot inputs was the optimal way to make outputs vary smoothly across the network.

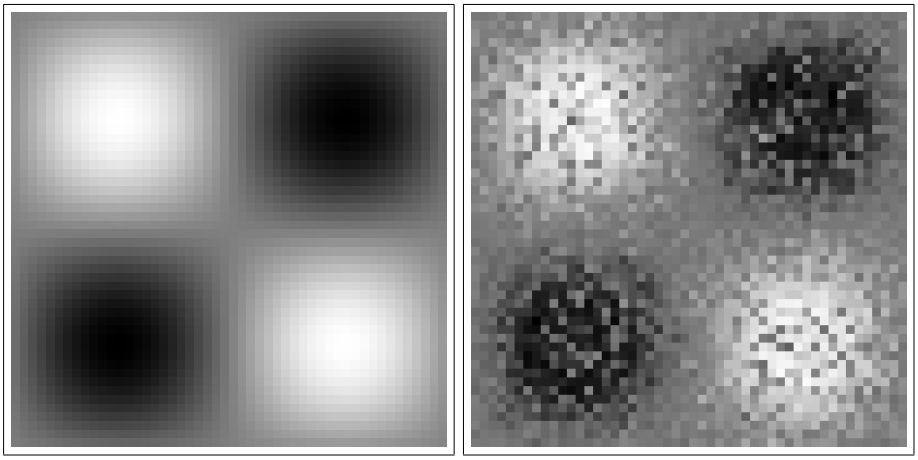


(*a*)



(*b*)

Figure 2. *Discovering disparity.* A stereo pair containing two 120 × 120 pixel images was generated in which 20% of pixels were set to black. Disparity varied between ±1 pixel as a Gaussian function ($\sigma = 30$ pixels) of distance from the centre of each image. Sub-pixel disparities were generated by sub-sampling from a single 1200 × 1200 image, so that each pixel in the stereo pair mapped to a 10 × 10 patch of pixels in this 1200 × 1200 image. Using this method, disparity varied in increments of 0.1 pixel. Each such image was then convolved with a Gaussian ($\sigma = 1$ pixel), and normalized to have zero mean and unit variance. The micro-network had three layers. It had two sets of 3 × 3 inputs, each set relating to corresponding regions of the two images. The hidden layer consisted of five units, and the output layer had one unit. A virtual network of 40 × 40 micro-networks therefore spanned the whole image space, with adjacent but non-overlapping sampling. The output space (*ij-space*) was therefore 40 × 40: we maximized $\log(V/U)$ in *ij-space*, treating it as toroidal (i.e. the edges wrapped around). The values $\bar{z}_{i,j}$ and $\tilde{z}_{i,j}$ are necessary for computing V and U at each iteration. The former was the mean of z across the whole space; the latter was an exponentially-weighted mean of z, centred upon i, j. The half-life of the exponential (in *ij-space*) was 2. (*a*) The input. The random-dot stereograms for the left and right eyes used in learning (before Gaussian smoothing). Pixel disparities vary smoothly between ±1 pixel. (*b*) The output. The left-hand image shows the Gaussian disparity values for the image-pair above. The right-hand image shows the network output after 1000 iterations of conjugate gradient ascent (at which point the value of the merit function was 1.8). The disparity values and network output are highly correlated ($|r| > 0.92$). (Continued oppposite.)



(*c*)

Figure 2. (Continued). (*c*) Generalization. The left-hand image shows the 'egg-box' disparity values for the test image. The right-hand image shows the network output when presented with these 'unseen' inputs. The high correlation is maintained ($|r| > 0.89$).

We repeated the simulation ten times using different initial network weights. In all cases, the network converged with a high value of the merit function (mean 1.644, s.d. 0.171). Furthermore, the absolute correlation between disparity and network was high for both the learning image (mean 0.905, s.d. 0.026) and the test image (mean 0.869, s.d. 0.029).

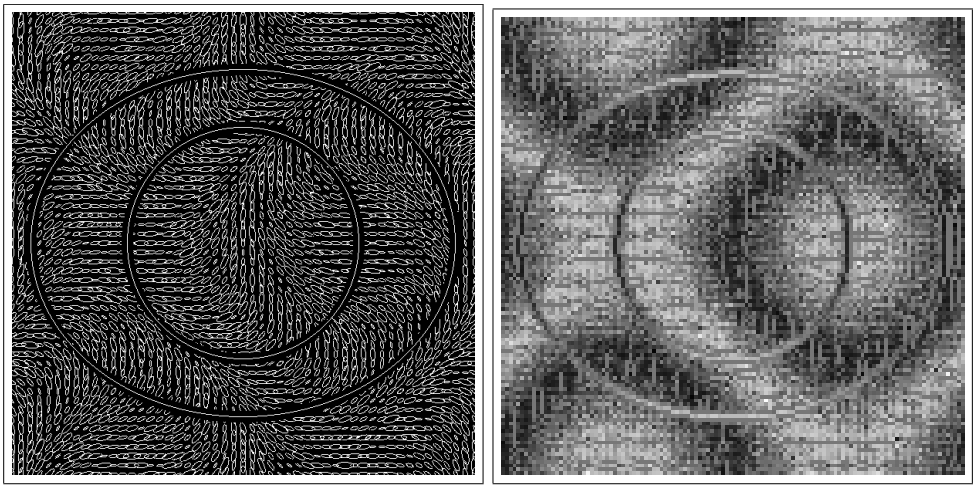
3.2. Demonstration 2: feature orientation

Many surfaces in the world have regular textures. The orientation of the texture elements on the image of a (curved or planar) surface tends to vary smoothly with image position†.

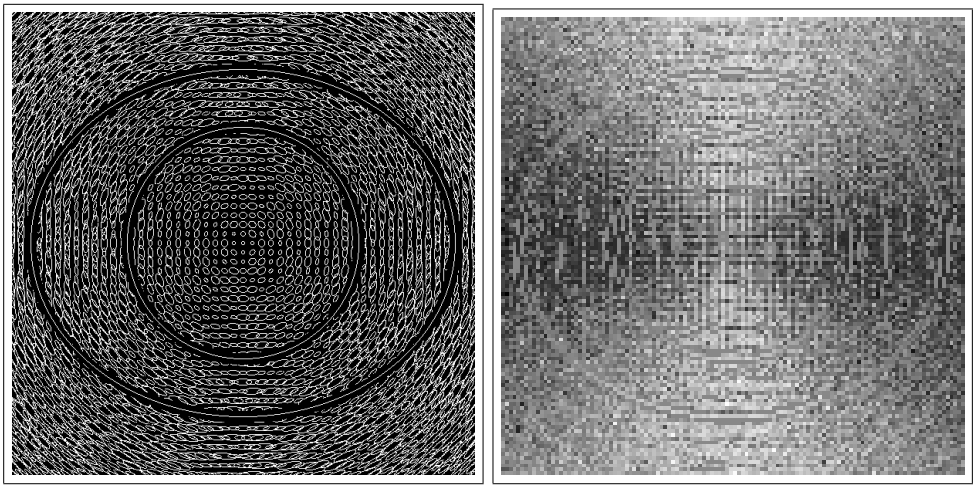
We created an image of oriented features in which orientation varied smoothly across the image. The features were small, possibly overlapping, ellipses whose size varied randomly (see figure 3(*a*), left). We constructed a virtual network such that each micro-network received inputs from a small region of the image, typically slightly smaller than the size of the smallest feature. As before, we maximized the merit function $\log(V/U)$ over the virtual network. In figure 3(*a*) (right) it can be seen that the output of a unit correlates with the orientation of ellipses at a corresponding position in the image. In order to ensure that the network had learnt the general property of orientation we generated another image along the same lines as the learning image (see figure 3(*b*), left). In figure 3(*b*) (right) it can be seen that when the trained network is presented with the new, unseen image the outputs are still correlated with feature orientation.

To check further that the network was extracting feature orientation, we computed the dominant orientation of each patch from the learning image. Figure 3(*c*) shows how network output varies in accordance with the computed orientation for the learning image. We see that output varies sinusoidally with the orientation (bearing in mind that orientations -90 and $+90$ are the same). This figure also shows the closest least-squares fit for the data to the curve $y = a\sin(2x + b) + c$ for the learning and test image. The two curves are

† Imagine a fronto-parallel plane covered in a regular texture of circles; as the eye scans left, right, up or down, the circles project to ellipses of varying orientation and aspect-ratio. The orientation of these elliptical projections will change smoothly with smooth eye motion.

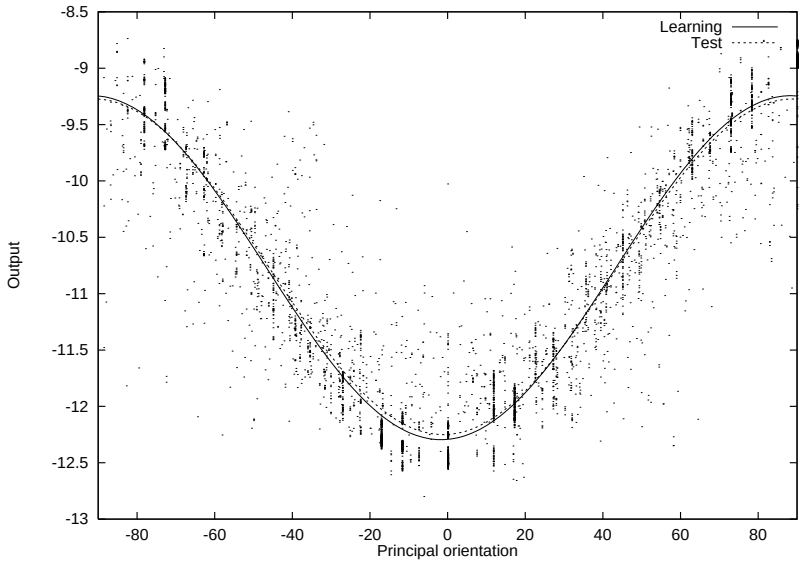


(*a*)

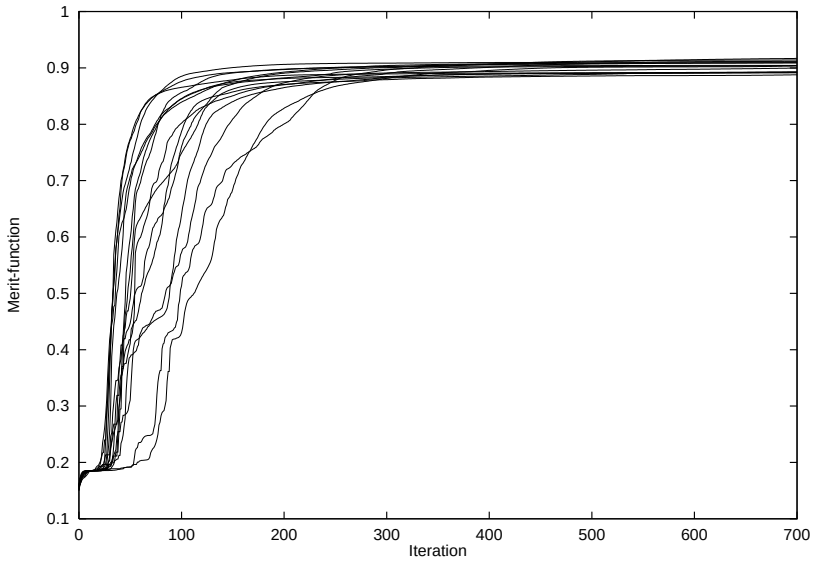


(*b*)

Figure 3. *Discovering feature orientation.* The images used were 600 × 600. Within each 12 × 12 patch of the image we drew a white, centred ellipse against the black background. The major axis had a random length between 4 and 11, and the appropriate orientation. The aspect ratio of the ellipse was 3. The micro-network had three layers. It had a set of 5 × 5 inputs; inputs were taken directly from the appropriate place in the image and then normalized to have zero mean and unit variance. It had nine units in the hidden layer and one unit in the output layer. A virtual network of 120 × 120 micro-networks therefore spanned the whole image space, with adjacent but non-overlapping sampling. The output space was therefore 120 × 120: we maximized $\log(V/U)$ in *ij-space*, treating it as toroidal. The value for $\bar{z}_{i,j}$ was taken as the mean across the whole space, whilst $\tilde{z}_{i,j}$ was computed using an exponential weighted mean of z centred upon i, j, using a half-life of 1 (in *ij-space*). (*a*) The learning image. The learning image is shown on the left, and the final outputs of the network (after maximizing $\log(V/U)$ for 1000 iterations of conjugate gradient ascent to a value of 0.9) on the right. Visual inspection shows that the outputs correlate well with feature orientation. (*b*) The test image. The image for testing generalization is shown on the left, and the network outputs, when presented with this image, on the right. Once more, visual inspection shows that the network generalizes to the new image and output correlates with feature orientation. (Continued oppposite.)



(c)



(d)

Figure 3. (Continued). (c) Network output as a function of the computed orientation. The dominant orientation of each learning image patch was computed by finding the axis about which the second moment was smallest (Horn 1986). Given the small patch size (25 pixels) we only included patches which had at least five non-zero pixel values. Furthermore, patches with a 'roundness' of greater than 0.2 were not considered, to reject weakly oriented patches. This reduced the number of patches from 14 400 down to around 5000. The solid line shows the fit of these data to the curve $y = a\sin(2x + b) + c$ for the learning image. The dashed line shows the corresponding fit for the data points from the test image (not shown). (d) Robustness of convergence. The value of the merit function during learning is plotted for each of the 15 simulations using different initial random weights. For each simulation, by around 300 iterations the network had already converged to a high value of $\log(V/U)$.

very similar (for the learning image $a = 1.526, b = -86.670, c = -10.770$; for the test image $a = 1.489, b = -87.797, c = -10.761$.) indicating that network performance is not particular to the learning image. (Note that not all data points were expected to fall directly on the theoretical sine curve, since each image patch could contain segments of ellipses with different orientations.)

Network robustness was tested by repeating the simulation with different initial weights. Again, in each case the network converged with a high value of the merit function (mean 0.903, s.d. 0.009, $n = 15$). Figure 3(d) shows the value of the merit function during network development, indicating the robustness of the convergence. Although in principle stimulus of any orientation could elicit the maximum response from the network, in practice we found that it was always either 0 or 90 degrees. This is most likely an artifact of the rectangular sampling of the input images.

4. Discussion

This work deals primarily with the discovery of representations through unsupervised learning. In common with other unsupervised learning methods (Becker and Hinton 1992, Linsker 1988), our method maximizes an information-theoretic function of the data (a nonlinear low-frequency function of the input (see appendix 2 of Stone 1996b)). It is our belief that maximizing a general function of this type can lead to the discovery of many varied representations, each depending solely on the nature of the input.

We have demonstrated that such an information-theoretic function can be derived from properties of the source generating the input; in the case of sensory input this is the physical world. In this paper we have considered a single constraint concerning the source of visual data; we have shown that this leads to a function which, if maximized, allows a network to discover properties of that visual world. We have demonstrated this for stereo disparity and feature orientation, but in neither case have we examined the representations discovered. Although we consider the nature of such representations interesting and informative, this omission reflects the focus of this work: namely, that representation is secondary in importance compared with the nature of the function that is maximized. Whereas any specific representation can only be understood in terms of the input the network receives, the form of the merit function is independent of any particular input.

However, this method is not without its drawbacks, both in terms of principles and implementation. Most importantly, our network has only been tested on input images with only one spatially smooth feature. In real-world images however, there is likely to be more than one spatially smooth feature present. This raises two issues. First, can the network detect both features? To do this, the virtual network must have more than one output unit, but the learning rule must also be changed because neighbouring output units will no longer be highly correlated. Second, if the network can only extract one feature, on what basis does it choose to extract one feature and ignore the others? If the underlying features vary on different scales, then the extent of short- and long-term averaging would favour one of the features. On a related issue, steps must be taken to prevent the network discovering trivial low-order features in the inputs. For example, in the feature-orientation experiments each image patch was normalized to prevent the network discovering average image intensity.

Our current model has several drawbacks as regards implementation. First, the size of the short-range half-life relies on the rate at which the underlying parameter varies: as the parameter changes more often, the short-range half-life must be reduced (Eglen *et al* 1996). However, if the parameter varies slowly enough the value of the short-range half-life is not crucial. For example, using the disparity images presented in figure 2, the network

converged with values of the short-range half-life varying between 1 and 10, although the networks with smaller half-lives required more time to converge. One solution to selecting the extent of spatial averaging would be to allow the network to adapt the extent over time during learning, in a similar fashion to Becker and Hinton's work on extracting disparity from slanted surfaces (Becker and Hinton 1992).

Second, we acknowledge that the task of the network has been simplified by using weight sharing since it reduces the number of free parameters in the network. However, we believe that conceptually the important factor is the ratio of the degrees of freedom in the network to the degrees of freedom in the data. As long as a network without weight sharing receives a sufficiently large number of image samples, this ratio will be maintained and the non-shared network will still converge.

Third, although the method we present is unsupervised, we use a back-propagation algorithm for function maximization. We regard the use of such a biologically improbable algorithm as just a convenient way of maximizing the merit function. Previous work on temporal invariance has shown how examination of the function derivative can lead to a synaptic learning rule which can be implemented as a biologically plausible mixture of Hebb and anti-Hebb learning (more acceptable in the domain of unsupervised learning (Stone and Bray 1995)). Performing a similar treatment on the learning rule presented here would be complicated by the use of a multi-layer network with nonlinear hidden units.

In this paper we have stressed the importance of the nature of the function that is maximized. Here we have considered a spatial constraint; further work could look at exploiting both spatial and temporal smoothness. However, other constraints derived from the nature of the data-source may be of equal use and affect the form of the function. We are looking for constraints which will allow 'canonical microfunctions' of cortex to be determined (Stone 1996a); such canonical microfunctions can provide insights into the structure of the canonical microcircuits (Douglas and Martin 1991) (i.e. the appropriate architecture and learning rule), rather than *vice versa*.

Acknowledgments

Thanks to Harry Barrow for help with adapting the learning rule to the spatial domain, and other useful comments.

References

Becker S and Hinton G E 1992 A self-organising neural network that discovers surfaces in random-dot stereograms *Nature* **355** 161–3

Douglas R J and Martin K A C 1991 A functional microcircuit for cat visual-cortex *J. Physiol.* **440** 735–69

Eglen S J, Stone J V and Barrow H 1996 Learning perceptual invariances: a spatial model *Technical Report* CSRP 404, School of Cognitive and Computing Sciences, University of Sussex

Földiák P 1991 Learning invariance from transformation sequences *Neural Comput.* **3** 194–200

Horn B K D 1986 *Robot Vision* (Cambridge, MA: MIT Press)

Linsker R 1988 Self-organisation in a perceptual network *IEEE Comput.* **21** 105–17

Oram M W and Földiák P 1996 Learning generalization and localization—competition for stimulus type and receptive-field *Neurocomputing* **11** 297–321

Phillips W A, Kay J and Smyth D 1995 The discovery of structure by multi-stream networks of local processors with contextual guidance *Network: Comput. Neural Syst.* **6** 225–46

Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors *Nature* **323** 533–6

Schraudolph N and Sejnowski T 1992 Competitive anti-hebbian learning of invariants *Advances in Neural Information Processing Systems* vol 4 (San Mateo, CA: Morgan Kaufmann) pp 1017–24

Stone J V 1996a A canonical micro-function for learning perceptual invariances *Perception* **25** 207–20
——1996b Learning perceptually salient visual parameters using spatiotemporal smoothness constraints *Neural Comput.* **8** 1463–92
Stone J V and Bray A J 1995 A learning rule for extracting spatio-temporal invariances *Network: Comput. Neural Syst.* **6** 1–8
Williams P M 1991 A Marquardt algorithm for choosing the step-size in backpropagation learning with conjugate gradients *Technical Report* CSRP 229, School of Cognitive and Computing Sciences, University of Sussex
——1995 Bayesian regularization and pruning using a Laplace prior *Neural Comput.* **7** 117–43