Communicated by Ning Qian

A Nonlinear Hebbian Network that Learns to Detect Disparity in Random-Dot Stereograms

Christopher W. Lee
Washington University School of Medicine, St. Louis, MO 63110 USA
Bruno A. Olshausen
Washington University School of Medicine, St. Louis, MO 63110 USA and California Institute of Technology, Pasadena, CA 91125 USA

An intrinsic limitation of linear, Hebbian networks is that they are capable of learning only from the linear pairwise correlations within an input stream. To explore what higher forms of structure could be learned with a nonlinear Hebbian network, we constructed a model network containing a simple form of nonlinearity and we applied it to the problem of learning to detect the disparities present in random-dot stereograms. The network consists of three layers, with nonlinear sigmoidal activation functions in the second-layer units. The nonlinearities allow the second layer to transform the pixel-based representation in the input layer into a new representation based on coupled pairs of left–right inputs. The third layer of the network then clusters patterns occurring on the second-layer outputs according to their disparity via a standard competitive learning rule. Analysis of the network dynamics shows that the second-layer units' nonlinearities interact with the Hebbian learning rule to expand the region over which pairs of left-right inputs are stable. The learning rule is neurobiologically inspired and plausible, and the model may shed light on how the nervous system learns to use coincidence detection in general.

1 Introduction

In recent years, linear Hebbian learning rules have been used to model the development of receptive field properties in the central nervous system (e.g., Linsker 1988; Miller *et al.* 1989; Sereno and Sereno 1991; Berns *et al.* 1993). These networks have many attractive features: they discover structure in input data, reduce redundancy, and perform principal component analysis (Hertz *et al.* 1991, pp. 197–215). Significantly, these models have for the most part ignored the nonlinearities inherent in real neurons (for an exception see Miller 1990). It is important to develop sound theories for the roles nonlinearities might play in such unsupervised neural networks. However, "good theories rarely develop outside the context of a background of well-understood real problems and special cases"

(Minsky and Papert 1988, p. 3). Thus, we have chosen to study in detail the problem of disparity detection for the extraction of surface depth from stereoscopic images. This problem is particularly appropriate because (1) disparity has known behavioral and neurobiological relevance, (2) psychophysicists have shown that random-dot stereograms provide simple, mathematically well-defined stimuli that capture the essence of the disparity problem (Julesz 1971), and (3) disparity processing has been proven to require nonlinearity for its implementation (Minsky and Papert 1988, pp. 48–54). For these reasons we created a network containing simple, nonlinear units that can learn to detect disparity in random-dot stereograms under biologically inspired, Hebbian learning rules. In this paper, we present a description of this network followed by an in-depth characterization of its learning and performance.

2 Inspiration from Neurobiology

Several basic characteristics of neural signaling shape our approach to modeling a nonlinear network. A first, simple form of nonlinearity is inherent in the neurobiology of synaptic transmission: a real synapse is either excitatory or inhibitory, whereas a model linear synapse may change from one to the other.[1] Another nonlinear aspect of synaptic transmission is long-term potentiation (LTP). LTP is the increase in synaptic efficacy that occurs between active pre- and postsynaptic neurons. The phenomenon expresses three basic properties in relating presynaptic activity to changes in synaptic strength: input specificity, associativity, and cooperativity. LTP is input specific in that nonactive synapses are not potentiated during induction of LTP. Associativity refers to the cell's ability to potentiate a weak input if it is paired with a simultaneously active strong input. Finally, "cooperativity describes the existence of an intensity threshold for induction" (Bliss and Collingridge 1993); in other words, a critical number of afferents must be active for induction of LTP (McNaughton *et al.* 1978).

By contrast, *linear* Hebbian rules are not cooperative in this sense. For a linear neuron, Hebb's rule states only that the change in synaptic strength is proportional to the postsynaptic activity times the presynaptic activity. This rule is associative and input specific, but has no threshold for induction: a single weak input repeated twice achieves the same level of synaptic enhancement that coactive inputs would achieve in one step (cf. Holmes and Levy 1990). Thus, LTP is Hebbian, but not linear.[2]

[1] Here we consider only fast synaptic transmission.

[2] Previously, Miller (1990) addressed some of these sources of nonlinearity by studying a system with two populations of on-off cells, developing an analytical framework based upon linearizing the difference between these two input sources. We have chosen a different approach by incorporating these properties of synaptic rectification and cooperativity directly into a neural unit.

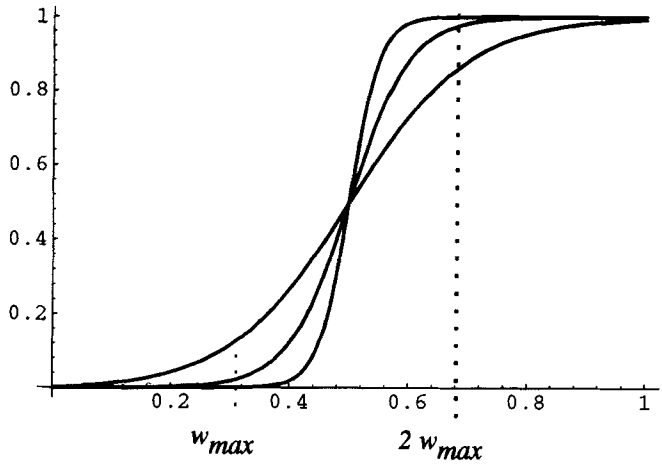

Figure 1: Form of the nonlinear function σ. Shown are graphs of $\sigma(x)$ for typical values used in our simulation, $\beta = 5$, 10, and 20. Dotted lines indicate the activation levels given by one or two synapses with maximal synaptic weights.

3 Mathematical Formalism

We incorporate the above properties into a neural unit based upon one of the standard model units commonly used in neural network models: the summing unit with sigmoid-shaped activation function. We define this sigmoid unit by its input–output relation and learning rule. Let y_j denote the unit output and let x_i denote the set of inputs to the unit. Then, if each connection strength is represented by a weight, w_{ji}, the output of the unit is given by

$$y_j = \sigma\left(\sum w_{ji} x_i\right) \tag{3.1}$$

where $\sigma(x)$ is a sigmoid-shaped function with a bias of $1/2$ so that for zero input the output is zero. Specifically,

$$\sigma(x) = \begin{cases} \frac{1}{1+e^{-2\beta(x-\frac{1}{2})}} & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{3.2}$$

where β determines the steepness of the slope at $x = 1/2$ (see Fig. 1). We also assume that the inputs are normalized to take on values from zero to one.

For our learning rule we use a standard form of Hebbian learning (Linsker 1988) with synaptic changes, Δw_{ji}, given by

$$\Delta w_{ji} \propto (\text{output}) \times (\text{input}) \tag{3.3}$$

$$\Delta w_{ji} \propto \sigma\left(\sum_a w_{ja} x_a\right)(x_i - o) \tag{3.4}$$

where ϕ is a parameter describing the amount of heterosynaptic competition among synapses ($0 < o < 1$).

To ensure consistency with the one-sided nature of (excitatory) synaptic transmission, the weights are allowed to take on values only greater than or equal to zero. Cooperativity in the learning rule comes from the sigmoidal nonlinearity. To maintain a "threshold" for weight increases, a single active connection should not be able to affect the unit strongly. Therefore, we must set an upper limit, $w_{\max}$, on the strength of any one connection. We choose $w_{\max}$ so that two strong inputs can have a strong effect on the unit [$\sigma(2w_{\max}) \approx 1$], but a single input can only weakly affect the unit [$\sigma(w_{\max}) \ll 1$]. Specifically, we set $w_{\max} = 1/3$. (The responses resulting from one or two inputs are superimposed on the graph of Figure 1.) In effect, we set our threshold to discriminate between states with one active input and states with two or more active inputs.

To get a feeling for how the weights will evolve, one can qualitatively describe the behavior of a single unit as follows: As a series of inputs is presented to the unit, the synapses will "compete" among themselves to maximize their strengths due to the heterosynaptic depression term, o. Over time, some synapses will begin to win out and others will be suppressed. However, unlike the linear case, a single synapse will be much less likely to dominate all the others because a single input, acting alone, is not able to induce a substantial change in the unit's response, and, hence, it is also unable to make a change in the unit's synapses. Strong synaptic modulation requires at least two inputs, and two inputs that are active at the same time will, on average, strengthen their weights more than the competition between the two inputs weakens the weights. In other words, it "pays" to cooperate, and so synaptic competition becomes a competition between pairs of inputs. On average, the pairs whose inputs are statistically correlated will have an advantage in that competition.

How much does it pay to cooperate? First, let us define the ratio, $R = \sigma(2w_{\max})/\sigma(w_{\max})$. Now, assuming that a pair of inputs with synaptic strengths $(w_{\max}, w_{\max})$ has already evolved, we can ask what it would take to destabilize the pair. On average, for each simultaneous, paired firing event, one synapse would have to fire without the other approximately R times to destabilize the pair. Note that the linear Hebb rule has $R = 2$, and thus shows a relatively small preference for pairings.

4 Simulation: "Learning Disparity"

We define our version of the problem of "learning disparity" as follows: Given a set of one-dimensional, random-dot stereograms, create a set of neural units that learns to become tuned selectively to the disparities present in the input. Random-dot stereograms have often been used as tests for disparity algorithms (e.g., Marr and Poggio 1975; Becker and Hinton 1992). This is, in part, because the lack of higher-level visual cues forces the algorithm to deal with the problem of false matches between left and right image elements. However, this lack of structure—specifically, the lack of correlations between pixel elements within each eye field—can simplify the learning process because it leaves only disparity-based structure in the input.

We make use of the nonlinear units defined above in a three-stage network that learns to solve this problem (illustrated in Fig. 2). At the first layer, the inputs are assumed to be one-dimensional, binary images from the left and right eyes, which we denote $\mathbf{x}^{\mathbf{L}}$ and $\mathbf{x}^{\mathbf{R}}$, respectively. The second stage consists of the sigmoid units, with outputs y_j and connection weights w_{ji}^e to the inputs x_i^e, where e takes on values **L** and **R**. Following equation 3.1, the outputs, y_j, are given by

$$y_j = \sigma\left(\sum_{e,i} w_{ji}^e x_i^e\right) \tag{4.1}$$

Each unit has connections to an equal number of inputs from the left and right eyes corresponding to the same region in the visual field. For ease of analysis, the inputs to each unit are chosen so they do not overlap.[3] In the third layer, we use a variant of a standard clustering network whose properties have been well characterized (Rumelhart and Zipser 1985; Hertz *et al.* 1991, pp. 217–219). Each unit, z_k, receives inputs from all the y_j weighted by synaptic efficacies V_{kj}; then a winner-take-all competition takes place among the third layer units to determine their final output. Only the winning unit changes its synaptic weights via a Hebb rule while the other units' outputs are set to zero. Thus, the third layer follows the equations:

$$z_k' = \sum_j V_{kj} y_j \tag{4.2}$$

$$z_k = \begin{cases} z_k' & z_k' > z_l' \ \forall l \neq k \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

$$\Delta V_{kj} \propto z_k(y_j - \psi) \tag{4.4}$$

After the winner adjusts its weight vector in the direction of the current input vector, the weights are renormalized so that $\sum_j V_{kj} = 1$. The sub-

[3] We have also shown that overlapping starting receptive fields can be used in conjunction with lateral inhibition to achieve a similar result (unpublished data).

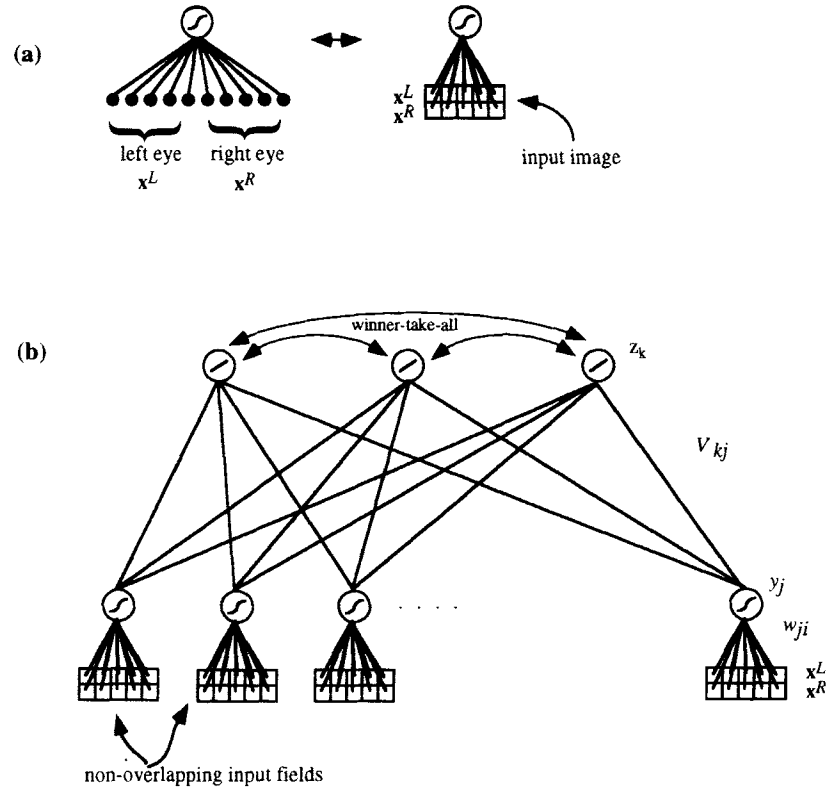

Figure 2: Model network. (a) A sigmoid unit y_j, receives a small number of inputs from the left and right eye first-layer units ($\mathbf{x^L}$ and $\mathbf{x^R}$, respectively). The input units are arranged with the left eye units stacked on top of the right eye units so that the two images are in register. (b) The architecture of the model network. The sigmoid units in the second layer have nonoverlapping input fields. A third layer of units, z_k, is connected to the sigmoid units, y_j though weights V_{kj}. The third layer units compete in a winner-take-all manner. The weights from the input to second layer and from the second to third layer evolve according to Hebbian equations 3.4 and 4.4.

tractive term, ψ, is added to help sharpen the competition within the weight vector (see also Goodhill and Barrow 1994).

We train the network on a sequence of random-dot stereograms at three different disparities. On each trial, the bits in the left eye image are

set randomly with bit probability p. The right eye image is then copied from the left eye image, shifted by an amount d:

$$x_i^R = x_{i+d}^L \tag{4.5}$$

where the disparity, d, is a randomly chosen integer from the set $\{-1, 0, 1\}$. We simulated the model using a second layer composed of 18 sigmoid units, with each unit connecting to five inputs from each eye. The inputs to each of these units corresponded to a separate field within the input array, and a one-pixel border was used to separate each input field from the next. The third layer of the network consisted of three units, each having connections to all of the first-layer units. The weights to the second and third layer were set to random values, and both sets of connections were allowed to evolve simultaneously according to equations 3.4 and 4.4.

5 Results

After training, the output-layer units had each learned to respond selectively to stereograms with different disparities. Results were similar for parameter values in the ranges: $5 \leq \beta \leq 20$, $0.65 \leq \phi \leq 0.85$, and $0.0 \leq \psi \leq 0.25$, and the results appeared insensitive to the type of initial conditions. Figure 3a shows a snapshot of the initial state of the network before learning. Figure 4a and b shows snapshots of the network at progressive stages of learning. In Figure 4a, the sigmoid units are beginning to become tuned to specific coincidences between left and right image pixels. In Figure 4b, this process has completed and the third layer neurons have become tuned to specific disparities over the entire input space. These examples represent typical results for these parameters. Because each second-layer unit has an approximately equal chance of becoming tuned to a disparity of either $+1$, 0, or -1, the number of units in the second layer tuned to a specific disparity varies from one simulation run to the next.

To illustrate the overall performance of the converged state of the network, we arranged the y_js so that their 18 first-layer receptive fields for each eye were stacked in two adjacent columns of nine. Thus, for each eye, the network's kernel has a rectangular domain of size 10×9 [$(5 \times 2) \times 9$] pixels that represents the 18 component receptive fields of 5 pixels apiece. We then convolved the network, thus arranged, with a random-dot stereogram. The result is shown in Figure 5. The output of the third layer is represented by plotting a different color pixel depending on which third-layer unit won. The network segregates a 0 disparity square (green) from a -1 disparity background (blue).

6 Analysis

Here we examine how each part of the network contributes toward the ability to extract disparities. The network relies upon the second layer units' ability to develop weights corresponding to disparity pairs. Each sigmoid unit acts as a coincidence detector; its output essentially computes a logical AND, or pseudo-multiplication, of two inputs $(x_i x_j)$ when β is large. (In the analysis that follows, we will always assume that β is large enough to be in this regime.) The second layer therefore consists of an array of primitive, location-specific disparity detectors. The third

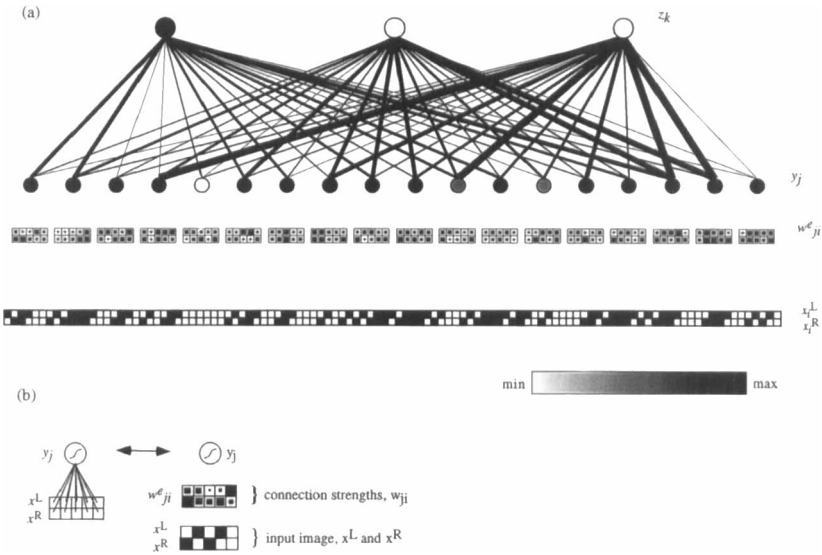

Figure 3: Initial state of the simulation. (a) A snapshot of the initial state of the network is shown with random connection strengths, $w'_{ji} \in [0.0.1]$ and $V_{kj} \in [0.01.1]$. The architecture is the same as in Figure 2. Connection strengths V_{kj} between the second and third layer units are indicated by line thickness. Connection strengths w'_{ji} are indicated by the size of the filled rectangle in the small boxes beneath each second layer unit, with a completely filled box indicating a connection of maximum strength. Activities of the units are indicated by shades of gray. The gradient bar shows the scale from zero (white) to one (black) of the unit activities. The input layer, labeled by $\mathbf{x^L}$ and $\mathbf{x^R}$, shows a $+1$ disparity image. (b) A detailed illustration of how the connection strengths w_{ji} are represented showing the correspondence between the depiction of weights in this and the previous figure.

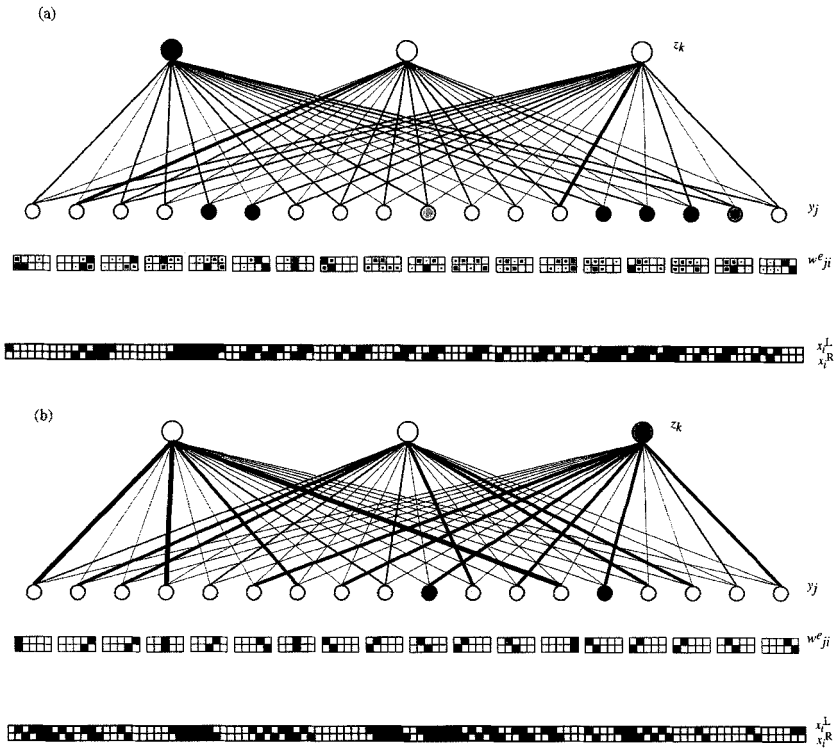

Figure 4: Simulation evolution. (a) The state of the network after a few hundred iterations. The nonlinear units in the second layer begin dropping their inputs. Some have already settled on two inputs, while others are still converging. (b) The final state of the network after several thousand iterations. All the sigmoid units in the second layer have eliminated all but two inputs—one from each image—making the unit crudely selective for disparity. The units in the third layer have successfully learned to group together the first-layer units signaling the same disparity. Weights values, w^e_{ji}, and activities are indicated by filled boxes and grayscale as in Figure 3: white = 0.0, black = 1.0. The strengths of the weights, V_{kj}, are denoted by the line thickness. The parameters used were $\beta = 10$, $\phi = 0.70$, and $\psi = 0.25$.

layer integrates across this array, effectively performing a cluster analysis on the variables $y_a = (x_{i_a} x_{j_a})$, where the y_as are the result of a selective sampling of the space of all multiplicative pairs of inputs. The network as a whole performs a generalized form of clustering, analogous to techniques used in statistics in which a set of variables is transformed before

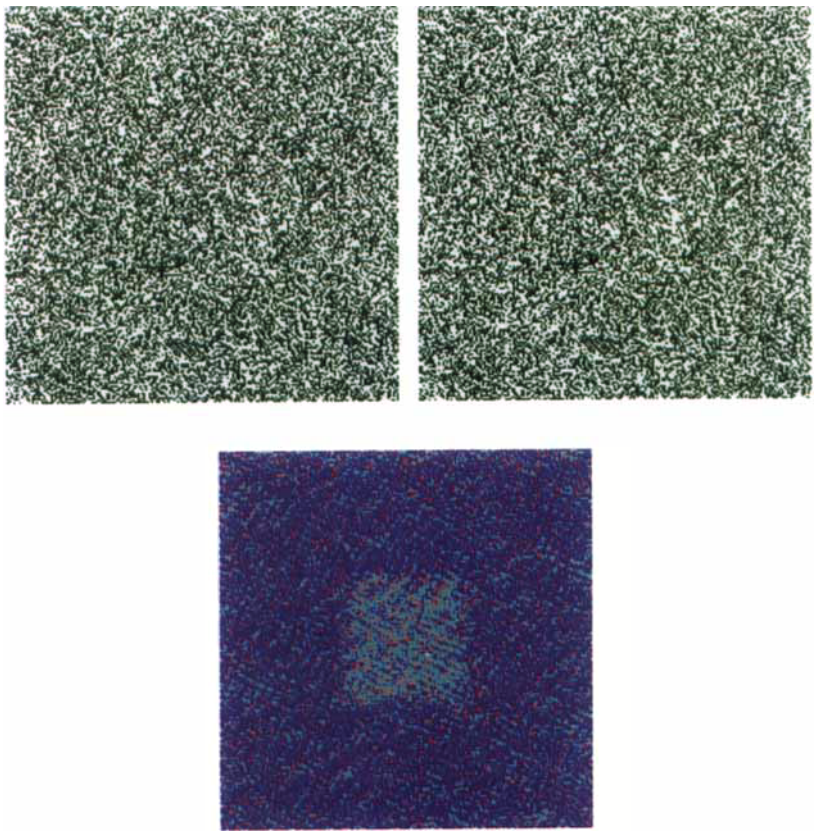

Figure 5: Performance of the network on a random-dot stereogram. The network's input fields were arranged as described in the text, then convolved with a random-dot stereogram (top) containing a square of 0 disparity upon a -1 disparity background. (below) The output of the network at each position is shown coded by color. Legend: blue, green, and red indicate -1, 0, and $+1$ disparities, respectively.

applying a standard procedure such as principal component or cluster analysis.

6.1 Evolution of the Second-Layer Units. Coincidence detection in the second-layer units depends upon the evolution of a weight vector with two nonzero components that are matched to one of the disparities

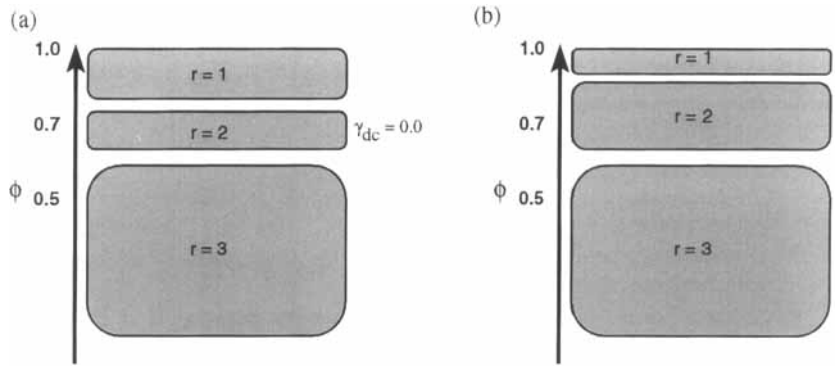

Figure 6: The effect of ϕ on the stable fixed points of the system. The number of nonzero components, r, in the stable points of (a) a three-dimensional linear system with $y = \beta \mathbf{w} \cdot \mathbf{x}$, and (b) a three-dimensional nonlinear system for $\beta = 10$. Note that the nonlinearity has enlarged the $r = 2$ region. These values are based upon computer simulations with random initial conditions and presentations as described previously, with the exception of the value of γ^{DC} in the three-dimensional linear system that was solved for directly (see Appendix).

in the input, i.e., a vector with $w_i^L = w_{i+d}^R = w_{max}$, for $d = -1, 0$, or 1, with all other $w_i = 0$. (We will refer to this weight configuration as a "disparity pair.") Strictly speaking, a nonlinear activation function is not required for development of this weight configuration. A modified linear system can develop disparity pairs when equipped with synaptic positivity ($0 \leq w_i \leq w_{max}$) and subtractive constraints (the ϕ term in our nonlinear system, see Miller and MacKay 1993 for an analysis of this form of constraint). For example, if instead of equation 3.1, we substitute for the unit output $y = \beta \mathbf{w} \cdot \mathbf{x} + b$, then under the same Hebb rule as equation 3.4, the system will evolve disparity pairs for $b = 0$, $\beta = 5$, and $\phi = 0.7$. Other combinations of β, b, and ϕ will also suffice.

We observed that a nonlinear activation function offers one advantage over a linear function in the process of developing disparity pairs: the nonlinear system converges to paired weights for a larger range of the parameters β and ϕ than the linear system. This occurs because the linear system must balance these parameters carefully. ϕ must be large enough to eliminate synapses while not so large that it eliminates one member of a disparity pair. By contrast, the cooperativity inherent in the sigmoid function selectively stabilizes pairs. Therefore ϕ may be larger than for a similar linear system while still converging to a pair. An illustration of this enlargement is shown in Figure 6. In the limit as $\beta \to \infty$, single

synaptic inputs produce no activation whatsoever, so that any weight vector that begins with greater than one component can never have less than two components. It is possible to estimate the critical value ϕ_c above which ϕ must be set to develop pairs in our system. For a system, with probability, p, of an input pixel being active, this is given by

$$\phi_c \approx \frac{2+3p}{5} \tag{6.1}$$

(see Appendix for a derivation). Note that the potential for enlargement of the ϕ parameter regime as compared to the linear system increases as the inputs become sparser.

An upper bound for ϕ is harder to define for reasons that bring us to the last issue concerning weight evolution: why the sigmoid units develop disparity pairs preferentially over nondisparity pairs. With the proper parameters, the sigmoid units pick out disparity pairs exclusively; that is, they become sensitive to shifts of $+1, 0$, and -1, but not $+2$ or -3, for example, since these disparities do not appear in the input. Members of a disparity pair fire together more often than the members of nondisparity pairs, but only slightly: for $p = 1/2$, the probability of a disparity pair of inputs firing together is $1/3$ while the probability for a nondisparity pair is $1/4$.[4] From our simulations we see that this is enough to allow the sigmoid units to select disparity pairs exclusively for values of ϕ near to ϕ_c; but as ϕ is increased the units begin to pick out nondisparity pairs occasionally. For example, in simulations with $\phi = 0.7$ the sigmoid units always pick out disparity pairs, while for $\phi = 0.85$, 9% of the units (eight out of ninety) converged to nondisparity pairs. (Other parameters were kept equal to those used in Fig. 3.) This may occur because with the greater rate of weight reduction for higher ϕ, the system does not have as much time to sample the input ensemble—allowing fluctuations in the input or in the initial weight configuration to have a greater effect in picking out an otherwise unfavored pair.

6.2 Clustering. Following the analysis of Rumelhart and Zipser (1985) for the $\psi = 0$ case, the third layer acts as a clustering mechanism that partitions the outputs from the second layer into compact regions. This relationship can be expressed graphically by projecting the input to the clustering layer onto the surface of a sphere,[5] as shown in Figure 7. Clusters are formed based upon the distance between points on the sphere, and the weight vectors $\mathbf{V}_k$ describe the cluster midpoints. The axes for this three-dimensional subspace are defined so that each

[4] In general, the ratio of these quantities, P(disparity pair)/P(nondisparity pair), equals $(1 + 2p)/3p$, indicating that learning is facilitated by sparse inputs (cf. equation 6.1; see also Field 1994).

[5] The constraint $\sum_j V_{kj} = 1$ actually defines a plane, but we project to a sphere for consistency with the geometric analogy used in Rumelhart and Zipser (1985). Either projection leads to the same conclusions in what follows.

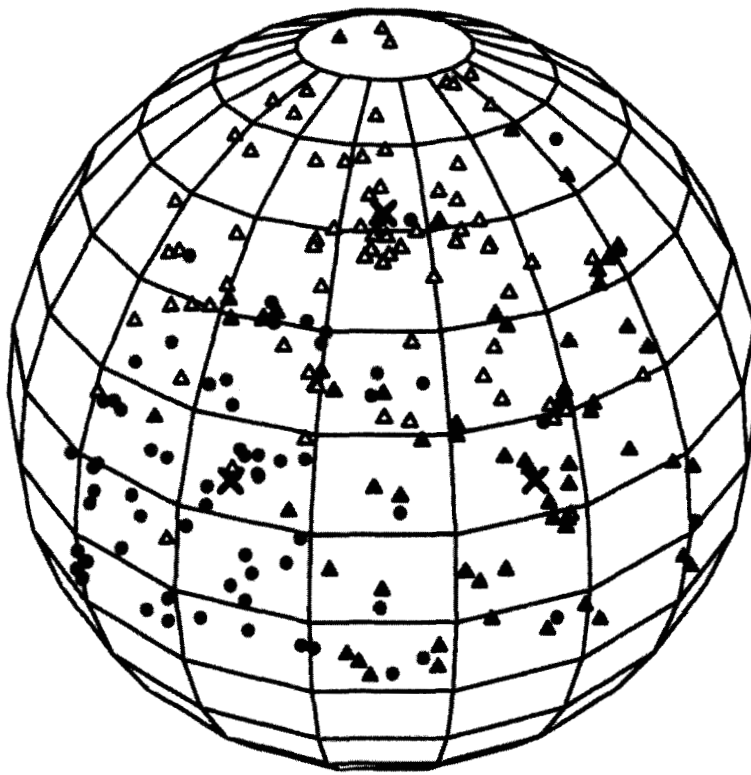

Figure 7: Distribution of first-layer outputs projected onto the sphere. Each symbol on the sphere represents the output of the second layer in response to one of the 200 stereograms presented to the network. The symbol shape corresponds to the actual disparity of the stereogram used to generate that point: open triangles, filled triangles, and gray circles represent disparities of $+1$, 0, and -1, respectively. The vector $\mathbf{y}$ was projected onto a sphere by reordering the basis used for the y_j into three groups. The first group corresponded to y_js that tune for $+1$ disparity, the second group corresponded to those that tuned for 0 disparity, while the last group corresponded to -1 disparity. A three-dimensional subspace was defined by axis directions $(1,1,\ldots,1;0,\ldots,0;0,\ldots,0)$, $(0,\ldots,0;1,1,\ldots,1;0,\ldots,0)$, and $(0,\ldots,0;0,\ldots,0;1,1,\ldots,1)$ where semicolons indicate the division between groups. The sphere corresponds to the unit sphere embedded into this subspace. Small xs mark the predicted center of mass for each symbol.

corresponds to a different disparity, and the unit sphere is embedded into this subspace as explained in the caption to Figure 7. If we project the output of the second layer of our network over 200 presentations onto the surface of this sphere, the points distribute to form a triangle (Fig. 7). The natural clusters for data distributed evenly over a triangle are the three corner regions. (To see this, create the Voronoi diagram that divides up the points equally for the triangular region using three polyhedrons.) Thus, for $\psi = 0$ the V_{kj} will stably align with the three corner regions of the triangle (in the vicinity of the Xs in Fig. 7) because these directions minimize cluster size. The z_k become disparity selective because the corners of this triangle also correspond to the different disparities. This structure can be seen by looking once again at Figure 7 where each point is labeled according to the disparity of the stereogram that generated it with either a filled triangle, open triangle, or gray circle. The center of mass of each symbol is strongly biased toward one of the corners of the output distribution, reflecting the different expectations in the y_j given the disparity of the input. That is, for $p = 1/2$:

$$\langle y_j \rangle = \begin{cases} \frac{1}{2} & \text{given } D = d_{y_j} \\ \frac{1}{4} & \text{given } D \neq d_{y_j} \end{cases} \tag{6.2}$$

where d_{y_j} is the disparity for which y_j is best tuned. In this way, position-independent disparity tuning results from the geometry of the second-layer responses to stereograms.[6]

Allowing $\psi > 0$ has little effect on the clustering actually performed by the network, though it makes it easier to see and analyze the clusters by "extremizing" the weight vectors—i.e., the weight vectors are moved away from the interior of the triangle and toward the corner vertices.

These spherical plots also illustrate why a linear second layer is inadequate for producing disparity selectivity. Figure 8 shows the output of a second layer using the same 200 input patterns and identical weights as in Figure 7, but with a linear activation function. Note that there is no structured shape to the distribution and that the different disparities are completely intermixed. Correspondingly, for the linear case, the average activities of the y_j do not differentiate between disparities, that is, for $y_j = \beta \mathbf{w} \cdot \mathbf{x}$,

$$\langle y_j \rangle = \begin{cases} \beta w_{\max} & \text{given } D = d_{y_j} \\ \beta w_{\max} & \text{given } D \neq d_{y_j} \end{cases} \tag{6.3}$$

Thus in this case, the V_{kj} can never reach a stable equilibrium. Instead, as might be predicted from Figure 8, the V_{kj} cycle continuously.

[6]Note, this geometry is contingent upon having spatially separate receptive fields. The clusters that confer position-independent disparity would be disrupted by the strong correlation in firing between the two units with overlapping fields. Thus, for this form of model to be effective, some mechanism—such as lateral inhibition or, as in the model described in this paper, direct subdivision of the input array—must exist to keep the units spatially decorrelated.

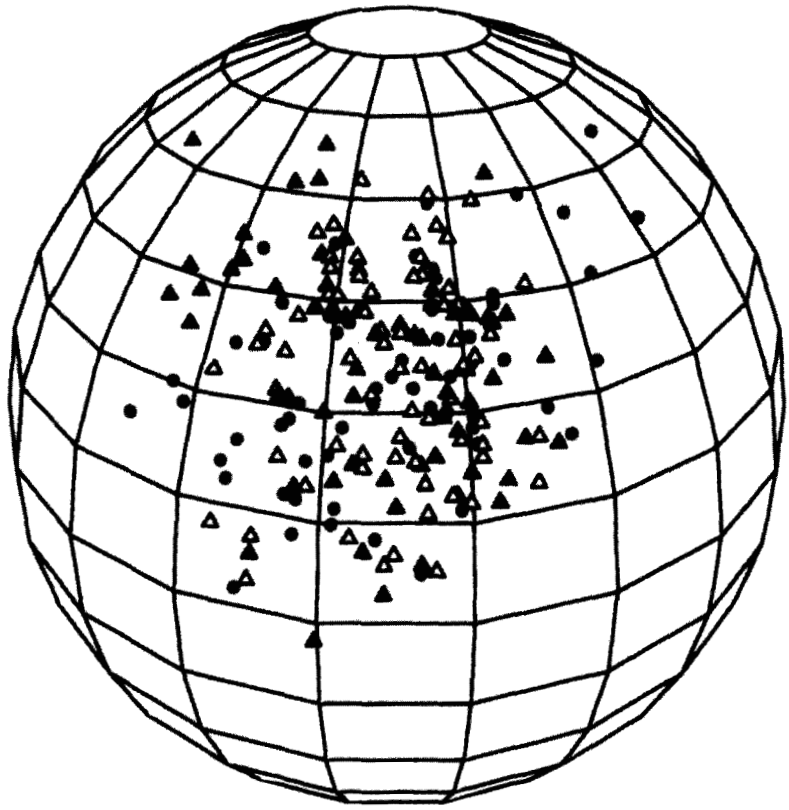

Figure 8: Distribution of linear first-layer outputs. Spherical plot generated in a manner identical to Figure 7, using the same 200 stereograms, except that the output of the second layer was based upon a linear activation function rather than a sigmoidal one.

6.3 Network Performance. We can calculate the accuracy of the network in classifying stereograms. Consider the y_j as binary random variables, with, for example, $p = 1/2$, $P(y_j = 1 \mid D = d_{y_j}) = 1/2$, and $P(y_j = 1 \mid D \neq d_{y_j}) = 1/4$. Let $\{j_{z_k}\}$ denote the set of indices into the second layer units to which z_k connects strongly, and let N_k equal the number of elements in this set. At equilibrium with $\psi > 0$, we observe that $z'_k = \sum_j V_{kj} y_j \approx 1/N_k \sum_{j \in \{j_{z_k}\}} y_j$. For convenience, define the integer random variable $S_k = (N_k z'_k)$. These S_k are conditionally independent, binomially distributed random variables, and their values on a given trial

determine the winner among the third-layer units. When an image with disparity D is presented, the performance of a z_k in signaling the correct disparity is given by

$$P(z_k = \text{winner} \mid D = d_{z_k}) = \sum_{l=1}^{N_k} P(z_k = l/N_k \text{ and } z_m < l/N_k,\ \forall m \neq k \mid D = d_{z_k}) \tag{6.4}$$

For symmetrically distributed y_j (i.e., $N_i = N_j,\ \forall i, j$) this simplifies to

$$P(z_k = \text{winner} \mid D = d_{z_k}) = \sum_l P(S_k = l \mid D = d_{z_k}) \times \prod_{m \neq k} P(S_m < l \mid D = d_{z_k}) \tag{6.5}$$

$$P(S_k = l \mid D = d_{z_m}) = B_{N_k,l} q^l (1-q)^{N_k - l} \tag{6.6}$$

where d_{z_k} is the disparity that z_k is tuned to, $B_{a,b}$ denotes the binomial coefficient, and $q = P(y_{j_{z_k}} = 1 \mid D = d_{z_m})$. This formula indicates that this type of network can classify stereograms with arbitrarily good accuracy by adding enough units. For $p = 1/2$ and a network with 18 first-layer units divided symmetrically among the z_k, $P(\text{correct}) = P(z_k = \text{winner} \mid D = d_{z_k}) = 0.61$ (versus 0.33 for chance). Accuracy improves gradually as the number of second-layer units increases, with the number of nodes scaling exponentially with increasing $P(\text{correct})$ for accuracies from 50 to 95%. (Ninety units are needed for a 95% accuracy level in our example network.) Likewise, to maintain fixed accuracy with an increasing number of disparities, $|D|$, the number of units must scale (approximately) as $O(|D| \log |D|)$.

7 Discussion

Our model demonstrates that a simple Hebbian network can learn to detect disparity. This is in contrast to all previous unsupervised models that have employed more complex and less biologically plausible learning schemes. In particular, it is instructive to compare our work with the work of Sanger (1989) and Becker and Hinton (1992). Sanger's network learns under a nonlinear extension of a "Generalized Hebbian Algorithm," which maximizes an observer's ability to reconstruct the network's input from its output. When presented with images derived from random-dot stereograms containing two disparities, the network learns to discriminate between these two disparities. Like our model, Sanger's network has a three-layer structure with simple (rectifying) nonlinearities in the second layer to generate the initial disparity sensitivity. However, learning occurs quite differently from our model in that once the second layer has converged, the disparity-sensitive units are identified and the

remaining units are discarded. The third layer is then trained solely on the outputs of these hand-picked units. In addition, Sanger's network cannot be implemented to simultaneously allow for a simple feedforward network architecture and a local learning rule. Finally, though learning can be made local, Generalized Hebbian Learning requires that synapses on different neurons be constrained to maintain the same synaptic strengths (Hertz *et al.* 1991, pp. 206–209). A biological mechanism capable of enforcing this constraint has yet to be demonstrated.

Becker and Hinton (1992) address a more complex version of the disparity problem in that their network learns to discriminate a continuous range of disparities and is also capable of representing spatial variations in disparity (i.e., curved surfaces). Their network is correspondingly more elaborate than ours. Learning is based upon the principle of maximizing the mutual information between groups of units viewing adjacent regions of visual space. In contrast with the simple Hebbian mechanisms used in our model, the weight update rule involves nonlocal backpropagation of the information signal.

Both Sanger's and Becker and Hinton's approaches involve powerful learning algorithms that can be more easily generalized to other problem domains than our own because they are derived from well-defined optimality principles. Our network embodies a complementary approach, emphasizing the ease of analysis and simplicity of the learning rule in the context of a specific task.

7.1 Computational Issues and Neurobiological Relevance. A criticism of much work involving "toy problem" networks is that by failing to characterize their scaling properties, they tend to leave the impression that a problem has been "solved" once and for all, leading investigators to neglect other approaches. In light of this, we point out that our network is computationally expensive to scale up to detect more disparities because network size scales as $O(|D| \log |D|)$. For example, a network that detects 20 disparity levels at 95% accuracy would require approximately 1000 second-layer units, a large number to require for even the most massively parallel devices. One of the reasons for this scaling inefficiency is that the steep sigmoid activation function on the subunits in the intermediate layer and the winner-take-all at the output layer effectively "binarize" the unit's responses, throwing out the information to be gained from a unit's continuous output. In addition, the input layer's pixel-based representation is far from optimal for the task. In the primate visual system, for example, the first stage of binocular image processing uses a monocular, spatially distributed input representation akin to a difference-of-gaussian (DOG), or possibly even orientation-tuned, set of filters. Disparity algorithms based upon this sort of representation both perform more robustly and scale more nicely for increasing disparity ranges than algorithms that use pixel representations alone (Qian 1994; Fleet *et al.* 1991).

While our emphasis in this paper has been to keep the network simple to highlight the essential aspects of nonlinear Hebbian learning, we are currently working to extend the model defined here to use continuous-valued units and a spatially distributed representation. Our preliminary results indicate that it is difficult for a single layer of a Hebbian network to develop a DOG-like or Gabor-like representation while simultaneously developing binocular disparity sensitivity (see also Erwin *et al.* 1995). One way around this difficulty appears to be the use of multiple network layers, with each layer learning one stage of the transformation. The brain uses such a multistage architecture to produce its representation, even though one layer would theoretically be adequate to implement the transformation. These observations, coupled with the scaling results set forth above, suggest that a multistage architecture may be a way of dealing with one of the general problems in designing a neural computer: that, for a given task, there is a tradeoff between network fan-in, depth, and number of nodes that can affect both the network's ability to represent a given function and its ability to learn that function (cf. Minsky and Papert 1988). From this point of view, the brain's hierarchy of visual processing (Felleman and Van Essen 1991) embodies a workable compromise that balances the costs of adding more connections, more neurons, or more stages, so as to effectively mix parallel and serial modes of computation.

Besides the lack of preprocessing mentioned previously, our network neglects the complexities of neural processing in the visual system in a number of respects. Real neurons in visual cortex receive input from thousands of synapses, have extensive and overlapping receptive fields,[7] and are regulated by sophisticated adaptive gain control mechanisms (Carandini and Heeger 1994) when presented with realistic images. In addition, while our learning rule captures the early aspects of synaptic plasticity (Levy *et al.* 1990), more sophisticated learning rules (e.g., Bienenstock *et al.* 1982) are required to better approximate the neurobiology. Nevertheless, we believe that our model may capture some of the essential, nonlinear aspects of synaptic learning, and it serves to illustrate a general strategy available to the nervous system: how a layer of units may be used to nonlinearly transform an input so that a downstream layer can learn to discriminate higher-order features in the environment. Evidence for this computational strategy can be seen across species and across sensory modalities. In the monkey visual system (Poggio *et al.* 1988), the bat echolocation system (Olsen and Suga 1991), and the sound localization system of the barn owl (Carr and Konishi 1990; Konishi 1992), nonlinear units act to transform sensory information into a format that explicitly represents coincidences within the input stream. Our results

[7] For an interesting model that deals with some of these issues and that is more closely related to the developmental neurobiology, see Berns *et al.* (1993). Though their model does not possess the requisite nonlinearities to perform disparity detection, they generate the primitives of disparity tuning via a Hebbian-type mechanism that relies upon the use of critical periods for development.

indicate that in addition to their function in processing signals, such coincidence detectors may also play a role in learning.

Appendix: Calculation of ϕ_c

To estimate ϕ_c, we rely upon a heuristic argument based upon a combination of observation, assumption, and approximation. The argument, while not completely rigorous, has given us useful rules of thumb for understanding the system, and it appears to be supported by our simulations. We begin by stating a few features of the system of equations defining the weights for a second layer unit.

Characteristics of the System. Perhaps the most important aspect we observe of the dynamical system, $\mathcal{S}$, as defined by equation 3.4, is that, like related linear systems (Linsker 1988; MacKay and Miller 1990; Miller and MacKay 1994), it has no equilibrium points within the interior of hypercube domain of w_j, as defined by $0 \leq w_j \leq w_{\max}$. All the stable points lie at the corners of the hypercube, which means that ultimately a w_j becomes either 0 or $w_{\max}$. A corollary to this fact is that the dimensionality of $\mathcal{S}$ effectively decreases as various w_j become zero. For the values of ϕ that we are interested in, we can view the evolution of $\mathcal{S}$ as a passage through a family of dynamic systems, $\mathcal{S}_i$, where the subscript denotes the effective dimension of the system. Over time, the dimension is steadily reduced until a final stable point is reached, that is, $\mathcal{S}_{\text{initial}} \to \mathcal{S}_{\text{initial}-1} \to \cdots \to \mathcal{S}_{\text{final}}$. In this framework, our problem is reduced to finding that value of ϕ which makes $\mathcal{S}_{\text{final}} = \mathcal{S}_2$.

Stability Analysis. Let r denote the effective dimension of a given system, i.e., for $\mathcal{S}_i$, $r = i$. Let $\mathbf{w}^{(r)}(t)$ denote the vector of weights w_j that are not zero, and let $\mathbf{n}^{(r)}$ denote the r-dimensional vector, $\mathbf{n}_i^{(r)} = 1$. An $\mathcal{S}_r$ can be the final system in the chain only when the point $\mathbf{w}^{(r)}(t) = w_{\max}\mathbf{n}^{(r)}$ is a stable point of the system $\mathcal{S}_r$.

To analyze the stability of this point, we resort to a piecewise linear approximation of σ.

$$\sigma(x) \approx \begin{cases} 0 & x < c_1 \\ m(x - c_1) & c_1 \leq x \leq c_2 \\ 1 & x > c_2 \end{cases} \tag{A.1}$$

where $c_1 = (m - 1)/2m$ and $c_2 = c_1 + m^{-1}$, and m is the slope of the line chosen to match the sigmoid. We can motivate this approximation by observing that the sigmoid has an approximately linear region around input levels of one half, and by noting that, in simulations using this piecewise

function instead of σ, we saw similar behavior to those simulations using a smooth sigmoid.[8]

If we substitute this approximation for the linear regime into equation 3.4 and set the constant of proportionality to one, we get

$$\frac{dw_j}{dt} = m(\Sigma w_i x_i - c_1)(x_j - \phi) \tag{A.2}$$

Following MacKay and Miller (1990), we write the equations for the linear region of the activation function, average over the input, and write the result in matrix form [while suppressing the superscript (r)]:

$$\langle \dot{\mathbf{w}} \rangle = (mQ + k_2 J)\mathbf{w} + k_1 \mathbf{n} \tag{A.3}$$

$$k_1 = \frac{(1-m)(p-\phi)}{2} \tag{A.4}$$

$$k_2 = mp(p-\phi) \tag{A.5}$$

Q is the covariance matrix $\langle (x_i - \langle x \rangle)(x_j - \langle x \rangle) \rangle$, J is the matrix $J_{ij} = 1$, and $\mathbf{n}$ is defined by $n_i = 1$ in the synaptic basis. Numerical solution for the fixed points of these systems, $\mathbf{w}^{\mathrm{FP}}$, shows that it generally lies outside the hypercube $[0, w_{\max}]^r$ close to the axis defined by $\mathbf{n}$. Given this, the stability of the hypercube vertex $w_{\max}\mathbf{n}$ is largely determined by the eigenvector (and its associated eigenvalue) of $mQ + k_2 J$, which has the largest component along the direction $\mathbf{n}$. Call this eigenvector, v^{DC} and denote its eigenvalue γ^{DC}. When γ^{DC} is close to zero, the point $w_{\max}\mathbf{n}$ is unstable because the other orthogonal eigenvectors with larger eigenvalues dominate the trajectory, carrying $\mathbf{w}(t)$ away from that vertex.[9]

We can solve directly for the value of ϕ that makes $\gamma^{\mathrm{DC}} = 0$ for $r = 3$. For inputs consisting of a disparity pair and one non-correlated input, Q is given by

$$\mathrm{Q} = \begin{pmatrix} p - p^2 & \frac{p}{3} - \frac{p^2}{3} & 0 \\ \frac{p}{3} - \frac{p^2}{3} & p - p^2 & 0 \\ 0 & 0 & p - p^2 \end{pmatrix} \tag{A.6}$$

Let $\phi_0^{(r)}$ denote the value of ϕ where $\gamma^{\mathrm{DC}} = 0$. Then $\phi_0^{(3)}$ is given by

$$\phi_0^{(3)} = \frac{2 + 3p}{5} \tag{A.7}$$

[8] An exception to this remark occurs for those units whose activation cannot rise above zero because of the absolute threshold of our approximation. These units become "dead units." In the nonlinear case, the smooth lower leg of the sigmoid allows such unit's weights to increase.

[9] Note, these other, non-v^{DC}, eigenvectors are orthogonal because the matrix is symmetric. MacKay and Miller (1990) show these eigenvectors and eigenvalues are less affected by changes in ϕ. The stability of vertices comes from the shape of the hypercube: at the boundary trajectories tend to be projected toward the corners.

We use this value as our estimate for ϕ_c. One might worry that while this value for ϕ_c might destabilize $\mathcal{S}_3$ it might not destabilize other, higher dimensioned systems. Direct calculation shows that $\phi_0^{(r)} > \phi_0^{(r+1)}$, for $r = 2, 3, 4, \ldots, 9$. In addition, for $r > 4$ or so, $\mathcal{S}_r$ is usually in the saturated region of the activation function, for which any $\phi > p$ implies instability.

References

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

Berns, G. S., Dayan, P., and Sejnowski, T. J. 1993. A correlational model for the development of disparity selectivity in visual cortex that depends on prenatal and postnatal phases. *Proc. Natl. Acad. Sci. U.S.A.* **90** (17), 8277–8281.

Bienenstock, E. L., Cooper, L. N., and Munro, P. W. 1982. Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* **2**, 32–48.

Bliss, T. V. P., and Collingridge, G. L. 1993. A synaptic model of memory: Long-term potentiation in the hippocampus. *Nature (London)* **361**, 31–39.

Carandini, M. C., and Heeger, D. J. 1994. Summation and division by neurons in primate visual cortex. *Science* **264**, 1333–1336.

Carr, C. E., and Konishi, M. 1990. A circuit for detection of interaural time differences in the brain stem of the barn owl. *J. Neurosci.* **10**(10), 3227–3246.

Erwin, E., Obermayer, K., and Schulten, K. 1995. Models of orientation and ocular dominance columns in the visual cortex: A critical comparison. *Neural Comp.* **7**, 425–468.

Felleman, D. J., and Van Essen, D. C. 1991. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex* **1**(1), 1–47.

Field, D. J. 1994. What is the goal of sensory coding? *Neural Comp.* **6**, 559–601.

Fleet, D. J., Jepson, A. D., and Jenkin, M. R. M. 1991. Phase-based disparity measurement. *CVGIP* **53**, 198–210.

Goodhill, G. J., and Barrow, H. G. 1994. The role of weight normalization in competitive learning. *Neural Comp.* **6**, 255–269.

Hertz, J., Krogh, A., and Palmer, R. G., 1991. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, CA.

Holmes, W. R., and Levy, W. B. 1990. Insights into associative long-term potentiation from computational models of NMDA receptor-mediated calcium influx and intracellular calcium concentration changes. *J. Neurophysiol.* **63**, 1148–1168.

Julesz, B. 1971. *Foundations of Cyclopean Perception.* University of Chicago Press. Chicago (as cited in Marr 1982, pp. 111–159).

Konishi, M. 1992. The neural algorithm for sound localization in the owl. *Harvey Lect.* **86**, 47–64.

Levy, W. B., Colbert, C. M., and Desmond, N. L. 1990. Elemental adaptive processes of neurons and synapses: A statistical/computational perspective.

In *Neuroscience and Connectionist Theory*, M. A. Gluck and D. E. Rumelhart, eds., pp. 187–235. Lawrence Erlbaum, Hillsdale, NJ.

Linsker, R. P. 1988. Self-organization in a perceptual network. *Computer* **March** 105–117.

MacKay, D. J. C., and Miller, K. D. 1990. Analysis of Linsker's application of Hebbian rules to linear networks. *Network* **1**, 257–298.

Marr, D. 1982. *Vision*. W. H. Freeman, New York.

Marr, D., and Poggio, T. 1975. Cooperative computation of stereo disparity. *Science* **194**, 283–287.

McNaughton, B. L., Douglas, R. M., and Goddard, G. V. 1978. Synaptic enhancement in fascia dentata: Cooperativity among co-active afferents. *Brain Research* **157**, 277–293.

Miller, K. D. 1990. Derivation of linear Hebbian equations from a nonlinear Hebbian model of synaptic plasticity. *Neural Comp.* **2**, 319–331.

Miller, K. D., and MacKay, D. J. C. 1994. The role of constraints in Hebbian learning. *Neural Comp.* **6**, 100–126.

Miller, K. D., Keller, J. B., and Stryker, M. P. 1989. Ocular dominance column development: Analysis and simulation. *Science* **245**, 605–615.

Minsky, M. L., and Papert, S. A. 1988. *Perceptrons: An Introduction to Computational Geometry* (expanded edition). MIT Press, Cambridge, MA.

Olshausen, B. A., and Field, D. J. 1995. Sparse coding of natural images produces localized, oriented, bandpass receptive fields. Tech. Rep. CNN–100–95, Dept. of Psychology, Cornell University. (Submitted for publication.)

Olsen, J. F., and Suga, N. 1991. Combination-sensitive neurons in the medial geniculate body of the mustached bat: Encoding of target range information. *J. Neurophysiol.* **65**(6), 1275–1296.

Poggio, G. F., Gonzalez, F., and Krause, F. 1988. Stereoscopic mechanisms in monkey visual cortex: Binocular correlation and disparity selectivity. *J. Neurosci.* **8**(12), 4531–4550.

Qian, N. 1994. Computing stereo disparity and motion with known binocular cell properties. *Neural Comp.* **6**, 390–404.

Rumelhart, D. E., and Zipser, D. 1985. Feature discovery by competitive learning. *Cog. Sci.* **9**, 75–112. Reprinted in Rumelhart et al. (1986, vol. 1, chap. 5).

Rumelhart, D. E., McClelland, J. L., and the PDP Research Group. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. MIT Press, Cambridge, MA.

Sanger, T. D. 1989. An optimality principle for unsupervised learning. In *Advances in Neural Information Processing Systems I*, D. Touretzky, ed., pp. 11–19. Morgan Kaufmann, San Mateo, CA.

Sereno, M. I., and Sereno, M. E. 1991. Learning to see rotation and dilation with a Hebb rule. In *Advances in Neural Information Processing Systems 3*, R. P. Lippmann, J. Moody, and D. S. Touretzky, eds. Morgan Kaufmann, San Mateo, CA.

Received January 3, 1994; accepted July 18, 1995.