[February 1993; revised April 1994; accepted October 1994.
In press, *Neural Networks* 8(3), 1995.]

Adaptive Perceptual Pattern Recognition by Self-Organizing Neural Networks: Context, Uncertainty, Multiplicity, and Scale

Jonathan A. Marshall

Department of Computer Science
University of North Carolina at Chapel Hill

Abstract— *A new context-sensitive neural network, called an "EXIN" (excitatory+inhibitory) network, is described. EXIN networks self-organize in complex perceptual environments, in the presence of multiple superimposed patterns, multiple scales, and uncertainty. The networks use a new inhibitory learning rule, in addition to an excitatory learning rule, to allow superposition of multiple simultaneous neural activations (multiple winners), under strictly regulated circumstances, instead of forcing winner-take-all pattern classifications. The multiple activations represent uncertainty or multiplicity in perception and pattern recognition. Perceptual scission (breaking of linkages) between independent category groupings thus arises and allows effective global context-sensitive segmentation constraint satisfaction, and exclusive credit attribution. A Weber Law neuron-growth rule lets the network learn and classify input patterns despite variations in their spatial scale. Applications of the new techniques include segmentation of superimposed auditory or biosonar signals, segmentation of visual regions, and representation of visual transparency.*

Acknowledgements: Supported in part by a UNC-CH Junior Faculty Development Award, an ORAU Junior Faculty Enhancement Award from Oak Ridge Associated Universities, the Office of Naval Research (Cognitive and Neural Sciences, N00014-93-1-0208), the National Eye Institute (EY09669), the University of Minnesota Center for Research in Learning, Perception, and Cognition, the National Institute of Child Health and Human Development (HD-07151), and the Minnesota Supercomputer Institute (Visiting Research Scholar award and Supercomputer Resource Grant). The author thanks Albert Nigrin, Christina Burbeck, John Hummel, Stephen Aylward, Robert Hubbard, William Gnadt, Michael Cohen, Sharon Chen, Vinay Gupta, George Kalarickal, and Charles Schmitt for their helpful comments on the paper, and R. Eric Fredericksen for help with the software.

Requests for reprints should be sent to the author at Department of Computer Science, CB 3175, Sitterson Hall, University of North Carolina, Chapel Hill, NC 27599-3175, U.S.A. E-mail marshall@cs.unc.edu. Telephone 919-962-1887.

Running title: **Context, Uncertainty, Multiplicity, and Scale**

Keywords—Masking fields, Anti-Hebbian learning, Distributed coding, Adaptive constraint satisfaction, Decorrelators, Excitatory+inhibitory (EXIN) learning, Transparency, Segmentation.

1. Introduction: Interactions Between Perceptual Groupings

Human perceptual systems do a remarkably good job of absorbing and organizing the complex data with which they are continuously barraged. They exhibit some powerful capabilities:

- **Context.** They are sensitive to contextual nuances, which can strongly bias their perceptual interpretations, and to contextual constraints, which limit the allowable interpretations.
- **Uncertainty.** They detect ambiguities in possible interpretations of the data and resolve them when warranted.
- **Multiplicity.** They separate multiple patterns occurring simultaneously in the data.
- **Scale.** They organize and interpret fine details as well as broad general trends.

This paper introduces a simple self-organizing neural network (SONN) model, called an **EXIN** (excitatory+inhibitory) network, that *learns* to perform aspects of each of these tasks. Because an EXIN network can learn, its design does not need to anticipate, or pre-wire, specific knowledge about the perceptual data to be organized. Rather, exposure to a perceptual environment during a developmental period configures the network so that it can appropriately organize subsequent data. Furthermore, because the SONN model is unsupervised, no external teacher is needed in establishing specific organizational decisions. A key feature of EXIN networks is their ability to segment, or parse, perceptual data in a globally context-sensitive manner into multiple coherent component groupings, subject to learned environmental constraints.

SONNs have been useful in modeling many aspects of the biological development of low-level perceptual processing capabilities. Such models have appeared most prevalently in the vision literature (Bienenstock, Cooper, & Munro, 1982; Coolen & Kuijk, 1989; Földiák, 1991; Grossberg, 1976abc; Hinton & Becker, 1990; Linsker, 1986ab; Marshall, 1989ab, 1990abef, 1991, 1992b; Martin & Marshall, 1993; M.E. Sereno, 1986, 1987; M.E. Sereno, Kersten, & Anderson, 1988; M.I. Sereno, 1989; M.I. Sereno & M.E. Sereno, 1990; Sun, Chen, & Lee, 1987; von der Malsburg, 1973). Unlike supervised learning techniques, such as backpropagation (Rumelhart, Hinton, & Williams, 1986; Werbos, 1974), unsupervised methods have the advantage of not requiring that an external "teacher" be available to program the system; in the lower levels of animal perceptual systems no external teacher is available (Reeke, Finkel, & Edelman, 1990). The ability of unsupervised SONNs to adapt or learn illustrates how certain perceptual response properties might be acquired in animal brains, without requiring detailed genetic specification of the full cortical wiring plan (Marshall, 1990a; Price & Zumbroich, 1989).

The job of the SONN researcher is like that of the physicist who creates a theory of the small (e.g., quantum mechanics) to account indirectly for many behaviors in the large (shapes of galaxies, formation of supernovae, etc.). In the same spirit, the SONN researcher creates a theory of the small (activation rules, learning rules, developmental rules, etc.) to account indirectly for many behaviors in the large (self-organized development of neural mechanisms that model various perceptual or cognitive phenomena). Rather than figuring out how to design neural networks to perform specific tasks, the SONN researcher tries to understand how they can configure *themselves* adaptively, through exposure to a pattern environment during a developmental period, without external supervision.

A deep challenge facing the researcher is how to choose a minimal set of learning rules that maximizes the breadth of perceptual or cognitive phenomena that the SONN can model. The researcher is highly constrained by considerations of environmental structure, physical locality, availability of top-down supervision information, and neurophysiological plausibility. Fortunately, certain simple rules go a long way toward producing such networks.

This paper describes one simple set of learning rules and its implications for two main questions in adaptive pattern recognition: (1) What constitutes a *pattern*? and (2) What is the structure of *interactions* between patterns? Issues of context, uncertainty, multiplicity, and scale all center around these questions and are therefore closely related to one another.

2. Scale Sensitivity and Adaptive Sequence Masking

The following example by Cohen and Grossberg (1986) illustrates certain scale-sensitive interactions between patterns, in the domain of speech perception:

> A word such as Myself is used by a fluent speaker as a unitized verbal chunk. In different verbal contexts, however, the components My, Self, and Elf or Myself are all words in their own right. Moreover, although an utterance which ended at My would generate one grouping of the speech flow, an utterance which went on to include the entire word Myself could supplant this encoding with one appropriate to the longer word (p. 1).

How exactly do the relative sizes, or *scales*, of patterns (Figure 1) affect the network's choice of an active pattern representation? Other things equal, according to Cohen and Grossberg, the *largest complete* representation (grouping) into which a component utterance fits is usually the appropriate one. When the complete word "self" is uttered, the representation of "self" should become active, and the representations of "my," "elf," and "myself" should be suppressed. This must occur even though the representations of "elf" (a smaller, subset word) and "myself" (a larger, superset word) are also receiving excitation at the same time. Grossberg (1978, 1986) and Cohen and Grossberg (1986, 1987) call such behavior the *sequence masking* property.

How can the sequence masking property be implemented? Some interactions between representations is required. Consider the representation of "elf." The representation of "elf" cannot merely receive input from its constituent parts (the written letters e, l, f or the corresponding auditory phonemes), for if it did not receive other inputs, there would be no way for it to distinguish between the presence of "elf" (and become activated) and the presence of "self" (and become inactivated). Thus, a given representation must in general incorporate information not only about the pattern(s) it represents, but also about patterns that it does *not* represent. In this sense, the representations of different patterns must interact.

The "myself" example by Cohen and Grossberg thus reveals key issues of *context*, as well as scaling. One's perceptual interpretation of e, l, f depends on context: the presence or absence of s. Because patterns "elf" and "self" overlap substantially, a relatively small contextual cue (or nuance) – the single input item s – can strongly bias one's interpretation of an input pattern containing e, l, f.

Issues of scale and context are crucial to understanding other perceptual domains as well, including aspects of vision. In vision, for example, one can view and recognize in isolation an object that could also be a component of a larger object. For example, when one views the letter d, one is not fooled into seeing the letter c, even though the d completely contains the c shape.

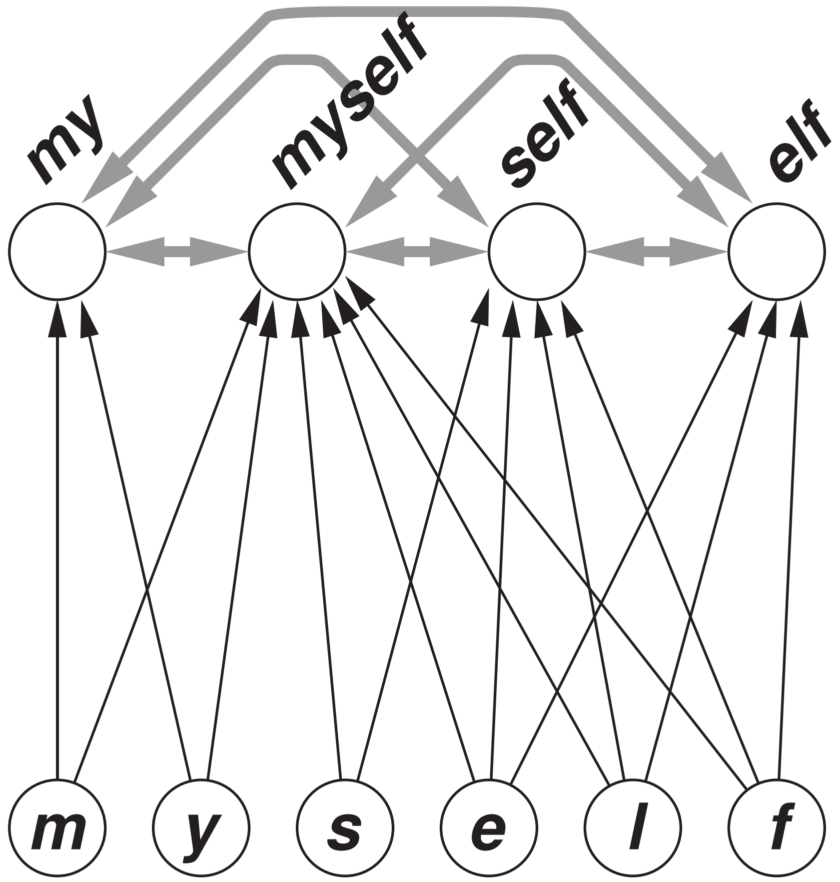

Figure 1: Sequence Masking Principle. How can the connection weights in the network be chosen so that the words "my," "myself," "self," and "elf" can each activate the corresponding Layer 2 neuron?

2.1. Balancing Excitatory Connection Weights

How can the connection weights in a network be chosen to implement the desired scaling properties? Figure 2A–D illustrates some of the specific issues, using a simple (five neuron) neural network. Two layers of neurons are schematically depicted: an input layer (Layer 1), for which neurons are labeled $a, b, c, \ldots$, and a processing layer (Layer 2). In general, such layers can be cascaded, forming a hierarchy of processing stages. Activation of a particular Layer 1 neuron can represent the presence of a certain feature in the input environment. In vision, for instance, activation of Layer 1 neuron a might represent the presence of a vertically-oriented red stripe centered at coordinates $(123, 45)$ of a visual image, or it might represent the presence of a bicycle wheel centered at coordinates $(67, 89)$. Other Layer 1 neurons could represent other features, or even the same features at different coordinates. What a given Layer 1 neuron actually represents is determined by its *own* inputs, from other neurons or from sensory transducers.

The networks developed in this paper work both with *binary* (0 or 1) inputs and with *analog* (continuously graded) inputs. For convenience, a binary input *pattern*, or a conjunction of input features, will be designated by the Layer 1 neurons that it activates. For example, the statement "input pattern bc is presented to the network" means that Layer 1 neurons b and c become active. A different notation to designate analog input patterns is also needed; these are indicated graphically in the Figures of this paper.

The active Layer 1 neurons feed excitatory signals to Layer 2 neurons via a network of excitatory connections. Layer 2 neurons are similarly designated by listing the Layer 1 neurons from which they receive strong excitatory connections; hence, the two Layer 2 neurons shown in Figure 2A will be labeled *ab* and *abc*. The two neurons *code* the patterns *ab* and *abc*, respectively. The Layer 2 neurons are interconnected by a network of lateral inhibitory connections, so that activation of one Layer 2 neuron inhibits the activation of other Layer 2 neurons.

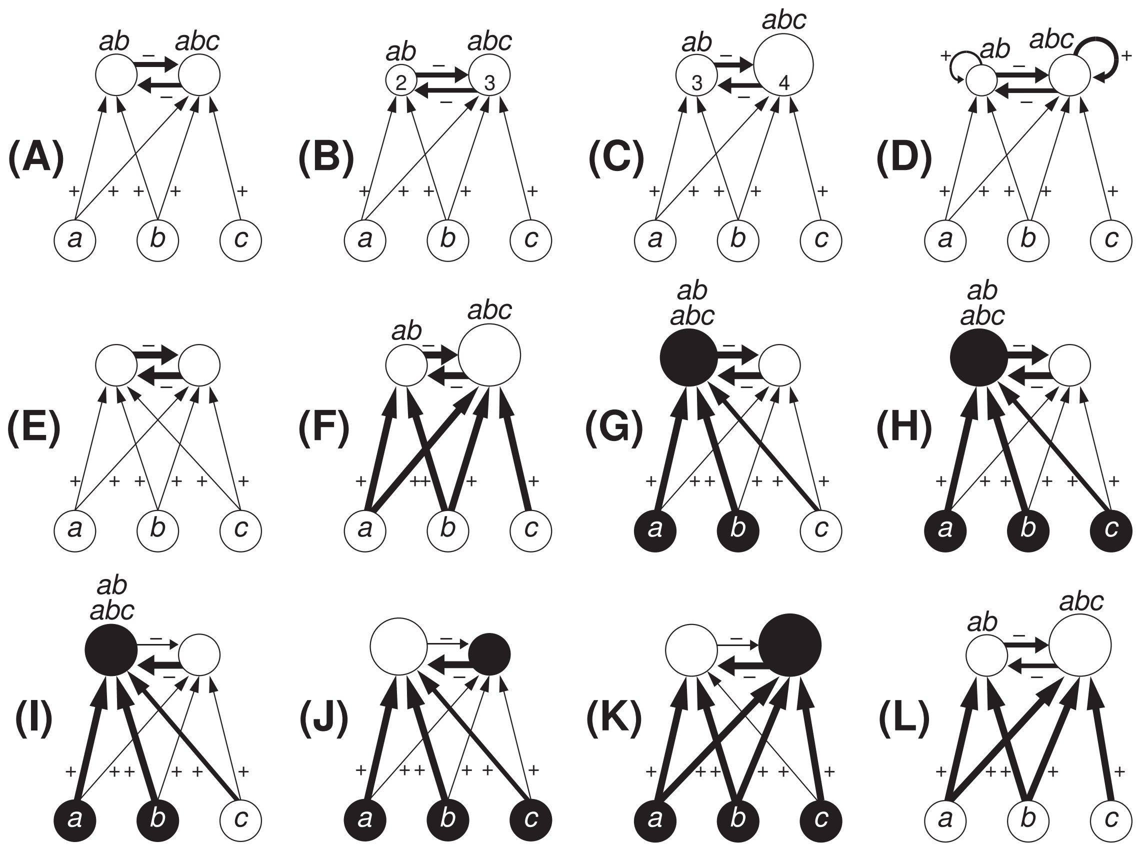

Figure 2: Key scaling issues. Layer 1 (bottom row in each figure) contains neurons *a*, *b*, and *c*, and Layer 2 (top row) contains neurons *ab* and *abc*. Feedforward excitatory connections (+) transmit excitatory signals from active Layer 1 neurons to Layer 2 neurons. Lateral inhibitory connections (–) transmit inhibitory signals from active Layer 2 neurons to other Layer 2 neurons. (A) Excitatory connections are all chosen to be equally strong (Balancing Rule 1). (B) Excitatory connection weights are chosen to be normalized by total sum of excitatory weights into each Layer 2 neuron (Balancing Rule 2). Excitatory connections into neurons *ab* and *abc* all have equal weights, but neurons *ab* and *abc* have size (sensitivity) normalization factors of 2 and 3, respectively, indicated by circle size. (C) In the case of Weber Law normalization (Balancing Rule 3) with $\alpha = 1$, neurons *ab* and *abc* have size (sensitivity) normalization factors of 3 and 4, respectively. (D) The masking field networks of Cohen and Grossberg use an additional feedback excitatory term from each Layer 2 neuron back to itself; the strength of each neuron's self-feedback depends on its total number of bottom-up excitatory inputs. (E) Initially, all neurons are uniformly connected, up to a small random factor. Thus, all Layer 2 neurons can potentially respond to all patterns, and no particular selectivity is exhibited. Network is then exposed repeatedly to patterns *ab* and *abc*. (F) It is desirable for connections to develop so that one neuron codes *ab* and the other codes *abc*. But it is possible that instead, one Layer 2 neuron will develop to respond (shading) to *both ab* (G) and *abc* (H), while the other Layer 2 neuron responds to *neither* pattern. (I) Inhibitory input weights to unused neuron gradually weaken, until (J) unused neuron begins to respond. (K) Formerly unused neuron now becomes selective for a pattern, via excitatory learning rule. (L) Inhibitory weights move toward restored symmetry. (Reprinted with permission from Marshall, 1990c.)

For the network to choose the largest complete matching representation for an input pattern, the input pattern sensitivities of all the network's Layer 2 neurons must be properly balanced.

Incorrectly balanced connection weights could cause the network to choose a representation at too large or too small a scale. Two incorrect ways and one correct way to balance the weights are discussed below.

Balancing Rule 1. Suppose all the network's excitatory connection weights were incorrectly initialized to uniform (equally strong) values (Figure 2A). Then when pattern abc is presented, Layer 2 neuron ab receives 2 units of excitation (write $E_{ab} = 2$), and neuron abc receives 3 units of excitation ($E_{abc} = 3$). So, neuron abc becomes more active than neuron ab and correctly suppresses ab's activity via lateral inhibition – the network recognizes abc. But when pattern ab is presented, both neurons receive equal amounts of excitation ($E_{ab} = E_{abc} = 2$), and an undesirable race condition occurs. The network does not guarantee that neuron ab will remain active and suppress neuron abc's activity. Thus, it is incorrect to specify uniform excitatory connection weights for this rudimentary network (i.e., Balancing Rule 1 does not work). This example illustrates a limitation (even neglecting learning) of some simple neural networks – they do not directly handle the scaling issues that arise when patterns of different sizes are encoded.

Balancing Rule 2. Alternatively, each neuron's excitatory inputs could be normalized by some factor, to compensate for the tendency of larger input patterns to dominate. In biological neurons, this normalization factor could realistically correspond to an overall *sensitivity* parameter within each neuron. In somewhat less biologically realistic terms, this normalization factor could perhaps more conveniently be conceptualized as the scale or "size" (Cohen & Grossberg, 1986, 1987; Grossberg, 1978, 1986) of each neuron. A given amount of signal entering a "large" neuron would diffuse into a large cell volume and would thus produce a small effect on the neuron's activity level. The same amount of signal entering a "small" neuron would diffuse into a small cell volume and would thus produce a larger effect on the neuron's activity level.

One possible mechanism would set the normalization (or size) factor equal to the sum of all the excitatory input connection weights to the neuron (Figure 2B). Then when ab is presented, $E_{ab} = \frac{2}{2} = 1$ and $E_{abc} = \frac{2}{3}$, so neuron ab wins the inhibitory competition and remain active. But when abc is presented, $E_{ab} = \frac{2}{2} = 1$ and $E_{abc} = \frac{3}{3} = 1$, so an undesirable race condition again would occur, this time for the larger pattern abc instead. Thus, this simple normalization scheme also would fail (i.e., Balancing Rule 2 does not work either).

Balancing Rule 3. A compromise between the two schemes is suggested: use a Weber Law rule, whereby a new constant, α, is added to the denominator of the normalization fraction (Figure 2C). More formally, the total excitatory input signal into the ith neuron is equal to

$$E_i = \frac{\beta \sum_j \left([x_j] z_{ji}^{+} \right)}{\alpha + \sum_j z_{ji}^{+}}, \tag{1}$$

where the quantity x_j represents the activity level of the jth neuron, $[x_j] \equiv \max(x_j, 0)$, and z_{ji}^{+} represents the weight of the excitatory connection from the jth neuron to the ith neuron. Then (supposing $\alpha = 1$) when input pattern ab is presented, $E_{ab} = \frac{2}{1+2} = \frac{2}{3}$, and $E_{abc} = \frac{2}{1+3} = \frac{1}{2}$, so neuron ab correctly wins. And when abc is presented, $E_{ab} = \frac{2}{1+2} = \frac{2}{3}$, and $E_{abc} = \frac{3}{1+3} = \frac{3}{4}$, so abc correctly wins. Thus, this scaling issue is properly solved. However, one possible disadvantage of the Weber Law scheme is that for larger input patterns, the differences between the amounts of excitation received by similar Layer 2 neurons can be quite small. The results in the present paper were produced using the Weber Law method.

Grossberg (1978, 1986) and Cohen and Grossberg (1986, 1987) have proposed a neural network architecture called a *masking field*, which uses a similar compromise. Instead of a Weber Law, they use an additional feedback excitation signal, from each Layer 2 neuron to itself, for which the magnitude depends on the sum of the excitatory input connection weights to the neuron (Figure 2D). Their method can be compared to the Weber Law method by writing

$$E_i \;=\; \frac{\sum_j\left([x_j]z_{ji}^+\right)}{\sum_j z_{ji}^+} \;+\; f(x_i)\sum_j z_{ji}^+. \tag{2}$$

Similar scaling behavior is obtained. However, EXIN networks overcome certain disadvantages of masking fields; these are described below in the Discussion section.

2.2. Adaptive Sensitivity Scaling

In a more dynamic framework, where learning occurs, *changes* in a neuron's scaling/size/sensitivity can occur in response to changes in the neuron's input connection weights. If a neuron's individual excitatory input connection weights become stronger or weaker, then the "size" or sensitivity of the neuron itself will grow or shrink correspondingly. In this way, a neuron's size normalization factor will always be maintained adaptively at the correct value for the pattern that the neuron codes.

If the normalization parameter is (unrealistically) viewed as a neuron's physical size, then each neuron can also be imagined as having a finite dendritic surface area available for implantation of synaptic receptor sites (Cohen & Grossberg, 1986). If a neuron's receptor area is already fully allocated, then the neuron must physically grow to increase that area when additional input connections are required. Then a given input signal would produce a smaller effect. Conversely, if certain input connections weaken, then the neuron may shrink to absorb the disused receptor sites, and a given input signal would then have a larger effect. Neuron "growth" is not the only process that might be hypothesized to subserve Weber Law normalization; instead, other processes such as long-term changes in membrane excitability could accomplish the same effects.

If S_i^+ represents the ith neuron's normalization parameter, then equation (1) can be rewritten as

$$E_i \;=\; \frac{\beta\sum_j\left([x_j]z_{ji}^+\right)}{S_i^+}. \tag{3}$$

Neuron growth (or changes in sensitivity) can then be described by the equation

$$\frac{d}{dt}S_i^+ \;=\; \phi\, l(x_i)(-S_i^+ + \alpha + \sum_j z_{ji}^+), \tag{4}$$

where the constant ϕ governs the growth rate, and the function $l(x_i)$ models the dependence (if any) of the growth on the neuron's activation. At equilibrium,

$$S_i^+ \;=\; \alpha + \sum_j z_{ji}^+. \tag{5}$$

That is, a neuron's "size" is equal to the sum of its excitatory input connection *weights*, plus a constant.

2.3. Excitatory Learning in Winner-Take-All Networks

Typically, adaptive changes in connection weights in SONNs are governed by a variant of a Hebbian (Hebb, 1949) learning rule, such as the following: *Whenever a neuron is active, its input excitatory connections from <u>active</u> neurons become gradually stronger, while its input excitatory* *connections from* <u>*inactive*</u> *neurons become gradually weaker.* The excitatory learning rule can be expressed mathematically as a differential equation (Grossberg, 1982b). Let z^+_{ji} represent the weight of the excitatory connection from neuron j to neuron i. Then

$$\frac{d}{dt} z^+_{ji} = \epsilon \, f(x_i)\Big(-z^+_{ji} + h(x_j)\Big), \tag{6}$$

where x_j represents the activity level of the jth neuron, $\epsilon > 0$ is a small learning rate constant, and f and h are half-rectified increasing functions, for example, $h(x_j) = \max(0, x_j)$.

Many SONNs have been designed to operate in a winner-take-all (WTA) fashion. That is, only one Layer 2 neuron or a small, tightly clustered population of Layer 2 neurons is allowed to be active at any moment. Often, this WTA rule is implemented by heavy lateral inhibition. Because only one functional output unit can be active at a time, any input to such networks is represented as a single, unitary, lumped pattern.

Figure 2E depicts the initial architecture of the simplest such network. A set of initially nonspecific feedforward excitatory connections projects from Layer 1 to Layer 2; that is, all Layer 2 neurons receive connections with roughly the same pattern of weights from the neurons in Layer 1.

Within Layer 2, a nonspecific set of lateral inhibitory connections interconnects the neurons. If the lateral inhibitory connections are strong enough, then this network operates as a simple WTA adaptive classifier. When an input pattern is presented (by activating or partially activating some of the Layer 1 neurons), it most strongly excites the "nearest" (in some pattern space) Layer 2 neuron, which then suppresses the activation of its neighbors in a WTA fashion via lateral inhibition. The feedforward excitatory connection weights to the active Layer 2 neuron are then modified to rotate its pattern preferences slightly closer to the input pattern (Amari, 1977; Amari & Takeuchi, 1978; Grossberg, 1976b; Kohonen, 1982, 1984; Singer, 1983, 1985).

Although such WTA networks are useful for a variety of simple category learning and recognition tasks, their capabilities are limited. Below are described some problems that simple WTA networks cannot handle, some simple extensions that give rise to networks with greater power, and the ways in which EXIN networks handle these problems.

2.4. The Pattern Overlap Problem in Learning

Networks for which learning is governed by a simple excitatory Hebbian rule such as the one described by equation (6) face a difficulty when exposed to a set of overlapping input patterns. For instance, suppose the initially nonspecific network of Figure 2E is exposed repeatedly to the patterns *ab* and *abc*. The abstract patterns *ab* and *abc* might represent similarly spelled but semantically distinct words like "my" and "myself" or like "to" and "top." One of the Layer 2 neurons should learn to respond to *ab* and the other to *abc* (Figure 2F). But because patterns *ab* and *abc* are quite similar, one of the Layer 2 neurons might begin to respond to *both* patterns (Figure 2G–H). The learning rule causes the connection to that neuron from neurons *a* and *b* to remain strong and causes the connection from neuron *c* alternately to strengthen and weaken. There is no guarantee that the second Layer 2 neuron would ever become activated enough to learn either of the input patterns. One neuron would eventually develop an ambiguous pattern sensitivity, and the other neuron would remain unused. Thus the network might be unable to attain its full representational capacity, and it might thereby be unable to distinguish between patterns *ab* and *abc*. Such a network could therefore be considered *inefficient*.

2.5. Inhibitory Learning

How can a neural network avoid getting stuck in such inefficient structural states? A parsimonious method (Marshall, 1989ab, 1990acdef, 1991, 1992ab) achieves the desired result, using only strictly local self-organization processes. The technique involves imposing an anti-Hebbian (Amari & Takeuchi, 1978; Carlson, 1990; Easton & Gordon, 1984; Földiák, 1989, 1990, 1992; Kohonen, 1984; Marshall, 1989ab, 1990acdef, 1991, 1992ab; Nigrin, 1990abc, 1992, 1993; Rubner & Schulten, 1990; Soodak, 1991; Wilson, 1988) *inhibitory* learning rule, to govern changes in the weights of the lateral inhibitory connections, in addition to the excitatory learning rule. For reasons described below, both the excitatory and inhibitory learning rules must not allow connection weights to "change sign" from excitatory to inhibitory, or vice versa. Both the excitatory and the inhibitory learning rules are variants of Hebb's (1949) learning rule; they are simple, physically local, and neurophysiologically plausible. EXIN networks use both excitatory *and* inhibitory learning rules (Easton & Gordon, 1984; Marshall, 1989ab, 1990acdef, 1991, 1992ab). The combination of the two rules allows EXIN networks to adaptively acquire more complex forms of behavior than WTA networks.

One inhibitory learning rule specifies that *whenever a neuron is active, its output inhibitory connections to other active neurons become gradually stronger (i.e., more inhibitory), while its output inhibitory connections to inactive neurons become gradually weaker.* The inhibitory rule can also be expressed mathematically as a differential equation (Marshall, 1989ab, 1990acdf, 1991, 1992ab). Let z_{ji}^- represent the weight of the inhibitory connection from neuron j to neuron i. Then

$$\frac{d}{dt} z_{ji}^- \;=\; \delta\, g(x_j)\Big(-z_{ji}^- + V q(x_i)\Big), \tag{7}$$

where x_j represents the activity level of the jth neuron, $0 < \delta < \epsilon$, and g and q are half-rectified increasing functions. Parameter δ is chosen very small, so that inhibitory weights become established very gradually, based on the long-term coactivation frequencies of each pair of neurons. Parameter V governs the overall amount of coactivation permitted in the SONN.

Thus, if two neurons, i and j, are frequently coactivated, perhaps because they represent identical or highly similar patterns, then the inhibitory learning rule tends to cause the weights of their reciprocal inhibitory connections to increase (become more inhibitory). This decreases the likelihood that they can become coactivated in the future. One of the two neurons thus becomes freed to learn a new, more dissimilar pattern. In this manner, the efficiency of the network in representing patterns according to their spatiotemporal input distribution (Barlow, 1980; Field, 1987; Kersten, 1987; Watson, 1987) is enhanced.

On the other hand, if neurons i and j are rarely coactivated, then the weights of their reciprocal connections tend to decrease (become less inhibitory), thereby permitting them to become coactivated on relatively rare occasions.

Adding this inhibitory learning rule lets the network solve the pattern overlap problem in learning, reducing the inefficiency, as described in Figure 2I–L. Suppose one neuron responds to both patterns *ab* and *abc* and another neuron does not respond to either pattern (Figure 2G–H), and suppose that *ab* and *abc* arise equally often in the input stream. As input patterns are presented, the first neuron is repeatedly activated, and the second neuron is not. According to the inhibitory learning rule, the weight of the inhibitory connection from the first neuron to the second becomes gradually weaker and weaker (Figure 2I), until eventually the second neuron begins to respond to some pattern (Figure 2J). The reciprocal inhibition weights between the two neurons are thus temporarily asymmetric.

Once the second neuron begins to respond, its excitatory connections begin to learn one of the patterns, say *abc*, and its size begins to correspond to that of its new input pattern (Figure 2K). Now that the second neuron is drawn into the fray, it gradually becomes selective for one of the input patterns, according to the usual excitatory learning rule. Even though the first neuron may still at first receive more excitation, the second neuron can compete effectively against it because the second neuron receives less inhibition. The learning rules ensure that both neurons tend to respond to different patterns (otherwise inhibition weights would keep rising). The reciprocal inhibition weights between the two neurons therefore change again, heading back toward restored symmetry (Figure 2L). Now, as desired, one neuron responds to *ab*, and the other to *abc*. The network thereby becomes able to distinguish between *ab* and *abc*, and both neurons are efficiently used. Thus, the network exploits a temporary instability in the balance of inhibition to equalize the response frequency of all the Layer 2 neurons. This equalization effect is similar to that achieved by "conscience" rules (DeSieno, 1988; Van den Bout & Miller, 1989) and by adaptive threshold rules (Földiák, 1990, 1992).

Nigrin (1990abc, 1992, 1993) has proposed a network, called SONNET, that also uses a form of adaptive "size" normalization, combined with excitatory and inhibitory learning. Both EXIN networks and SONNET networks self-organize to have similar connectivity patterns and neuron sizes. SONNET networks, however, require substantially more complex structure, involving top-down feeback, reset mechanisms, and confidence measures. By using the extra complexity in combination with a form of weight "freezing," SONNET networks are intended to code learned patterns more stably (Carpenter & Grossberg, 1987a; Grossberg, 1980) than EXIN networks and other competitive learning algorithms do, particularly in nonstationary environments (where the statistical distribution of input patterns changes unpredictably over time). Such extra stability is sometimes desirable; however, it may not always be necessary, as discussed below in Section 8.7.

2.6. Symmetric Inhibition From an Asymmetric Rule

It is often desirable during learning to keep inhibitory connection weights symmetric between pairs of neurons; that is, $z_{ji}^{-} = z_{ij}^{-}$. Otherwise, one neuron could in principle become able to suppress the activity of all other neurons. Thus, whenever z_{ji}^{-} changes, so must z_{ij}^{-}. In addition, some mathematical analyses of the stability of learning require the assumption of symmetric inhibition (Cohen & Grossberg, 1983; Grossberg, 1982b, Hopfield, 1982). The inhibitory learning rule of equation (7), which operates in a strictly local manner, tends dynamically to make asymmetric inhibitory connections symmetric. Suppose $z_{ji}^{-} > z_{ij}^{-}$. Then neuron j is more likely to become activated than i, because j can suppress i's activity. However, when j does become activated, the rule causes z_{ji}^{-} to weaken, heading toward restored symmetry.

The inhibitory learning rule of equation (7) is an outstar (Grossberg, 1982b) rule, unlike the excitatory learning rule of equation (6), which is an instar rule. This means that the *rate* of learning depends on the activity level of the *pre*synaptic neuron (x_j), and the target connection weight depends on the activity level of the *post*synaptic neuron (x_i). The reason that it is an outstar depends on a worst-case analysis. Suppose that all the bottom-up connections to one neuron, n_0, are strong, and that all the bottom-up connections to other neurons are weak. This is the worst case because n_0 will respond to all patterns unselectively, and none of the other neurons will respond to any input pattern.

If the lateral inhibitory connection weights were governed by an outstar rule, then a connection weight would change when its source neuron is active. All the inhibitory connections from n_0 to other neurons would weaken progressively, until some other neuron(s) began to respond to some input patterns. Then the other active neurons would begin learning these patterns; the excitatory connections to them would strengthen, and some excitatory connections to n_0 would weaken. The network is thus able eventually to climb out of the worst-case state.

But if the lateral inhibitory connection weights were governed by an instar rule, then a connection weight could change only when its target neuron is active. Because n_0 is the only active neuron, its inhibitory connections to all the other neurons would remain strong. Therefore, it would be unlikely that any of the other neurons could ever become active. Thus, the network would tend to remain in this worst-case state. For this reason, an outstar form, not an instar form, is favored for the inhibitory learning equation (7).

It also turns out to be an advantage for the inhibitory learning rule to be asymmetric. This allows infrequently-activated neurons to become less inhibited yet maintain strong inhibition of their more-active competitors. The infrequently-activated neurons can then become active more often. The network thus tends to keep all neurons active with roughly the same frequency. This property tends to maximize the information content of each neuron's activation (Linsker, 1988). Although the inhibitory learning rule automatically maintains a rough symmetry of reciprocal inhibitory weights, it also allows reciprocal inhibitory weights to differ temporarily until roughly equal frequency of activation is achieved.

3. Wholes vs. Parts: Scale Sensitivity in an EXIN Neural Network

The adaptive network behavior governed by the inhibitory learning rule provides several additional benefits for pattern recognition. These are illustrated by Simulation I, described below. Complete implementation details for all the simulations are supplied in the Appendix.

The network of Simulation I has six Layer 1 neurons and six Layer 2 neurons, initially connected nonspecifically (Figure 3A). The excitatory connections from Layer 1 neurons to Layer 2 neurons were initially all strong (weight values all near 1) so that any input pattern could strongly excite every Layer 2 neuron. (If they were initially weak, then some patterns might be unable to activate a Layer 2 neuron enough for learning to occur.) The lateral inhibitory connections between Layer 2 neurons were also initially strong (weights near 0.250) so that the Layer 2 neurons would be selective (WTA) despite the initial similarity of their codes.

The activity level x_i of each Layer 2 neuron changes according to a shunting equation (Grossberg, 1972, 1982b):

$$\frac{d}{dt}x_i \;=\; -Ax_i \;+\; (B - x_i)E_i \;-\; (C + x_i)I_i, \tag{8}$$

where A, B, and C are constants, and where E_i and I_i represent the neuron's total excitatory and inhibitory input signals, respectively. Because equation (8) is a shunting equation, neuron activation levels are forced to remain within a bounded range. The bounding of the activations causes the learning rules to be bounded as well: all connection weights remain within a specified range. E_i is defined as in equation (1), and I_i is defined as

$$I_i \;=\; \gamma \sum_j \Big([x_j] z_{ji}^- \Big), \tag{9}$$

where parameter γ describes the overall influence of the summed inhibitory input.

Changes in the excitatory connection weights z_{ji}^+ and the inhibitory connection weights z_{ji}^- are governed by the learning equations (6) and (7), respectively. A simplifying assumption is made: neuron "growth" occurs fast relative to the frequency with which each Layer 2 neuron is activated; that is, the growth rate parameter ϕ in the neuron "growth" equation (4) is taken to be relatively

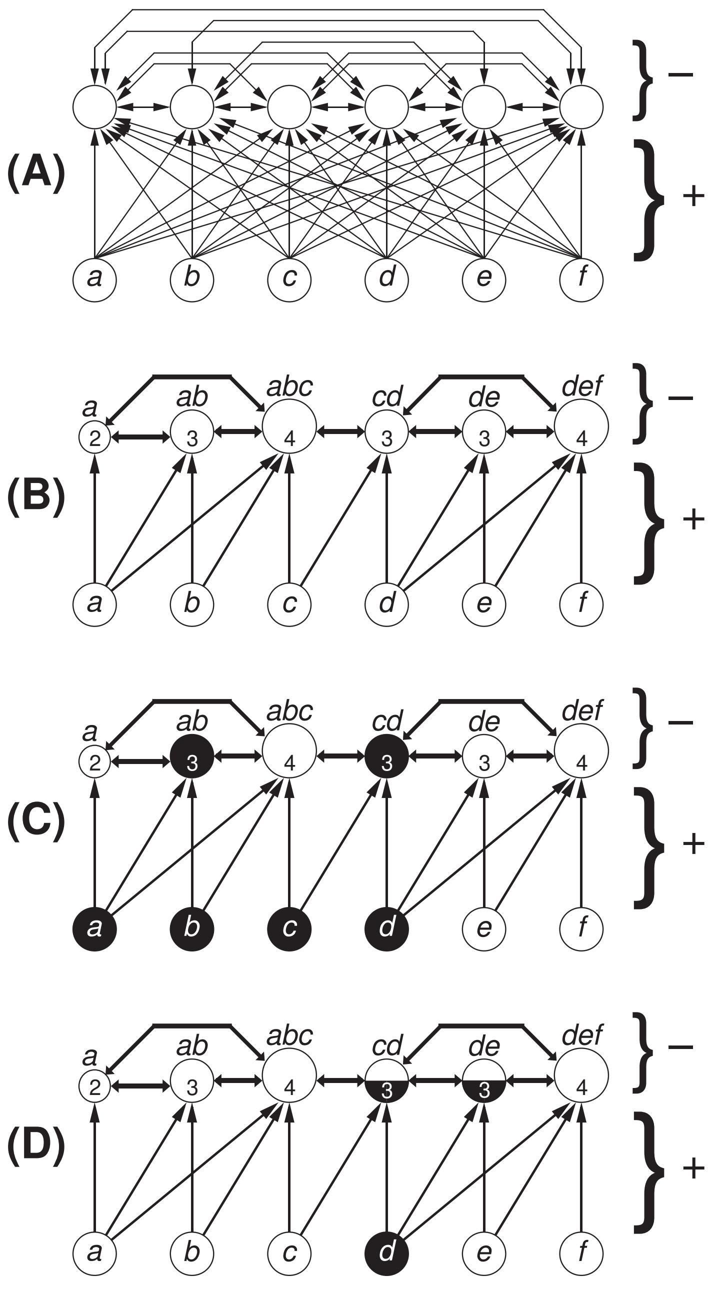

Figure 3: Simulation I. (A) Network for Simulation I is initially nonspecific. (B) The remaining strong connections are shown after 3000 training presentations of input patterns drawn from the set $(a, ab, abc, cd, de, def)$. The input pattern coded by each Layer 2 neuron is listed above the neuron body. The approximate "size" normalization factor of each Layer 2 neuron is shown inside the neuron. Each input pattern is coded by a different Layer 2 neuron. Strong (weights between 0.010 and 0.033) reciprocal inhibitory connections (thick lateral arrows) remain between neurons coding patterns that overlap; the other inhibitory connections are weak (0 to 0.006) and are omitted from the figure. (C) Patterns ab and cd do not overlap, so reciprocal inhibitory connections between neurons ab and cd are very weak. On a rare occasion when patterns ab and cd are presented simultaneously as $abcd$, *both* neurons ab and cd become fully active (filled circles). (D) When the ambiguous pattern d is presented, both neurons cd and de become moderately active (half-filled circles). Neither suppresses the activity of the other because they project only moderately strong reciprocal inhibitory connections and because neither receives its full excitation.

large, enabling the equilibrium "size" equation (5) to be used directly. It is further assumed for computational simplicity that in the growth equation (4), the activity-dependence term $l(x_i) \equiv 1$ for all neuron activation values x_i. Neuron growth is thus modeled as occurring while the neuron is inactive, as well as while it is active.

The network's training input set for Simulation I consisted of six discrete binary patterns presented repeatedly in random order: *a*, *ab*, *abc*, *cd*, *de*, and *def*. These abstract input patterns were chosen to probe the network's behavior in response to varying degrees of spatial overlap. The patterns are binary; each of the six Layer 1 neurons is either active at a fixed level or inactive.

After approximately 3000 input pattern presentations (about 500 presentations of each of the six patterns), the structure illustrated in Figure 3B appeared. The weights are more precisely displayed in Figure 4. This structure was stable, fluctuating only slightly (because of random variations in the pattern sequence) during an additional exposure test through 50,000 input pattern presentations. Several noteworthy features of this final structure can be discerned:

- each Layer 2 neuron receives excitatory connections that code exactly one of the six input patterns;
- each of the six input patterns activates its corresponding Layer 2 neuron;
- each Layer 2 neuron grows to a size corresponding to its input pattern's scale;
- reciprocal inhibitory weights are approximately symmetric;
- inhibition is strongest between neurons coding overlapping patterns and weakest between neurons coding nonoverlapping patterns.

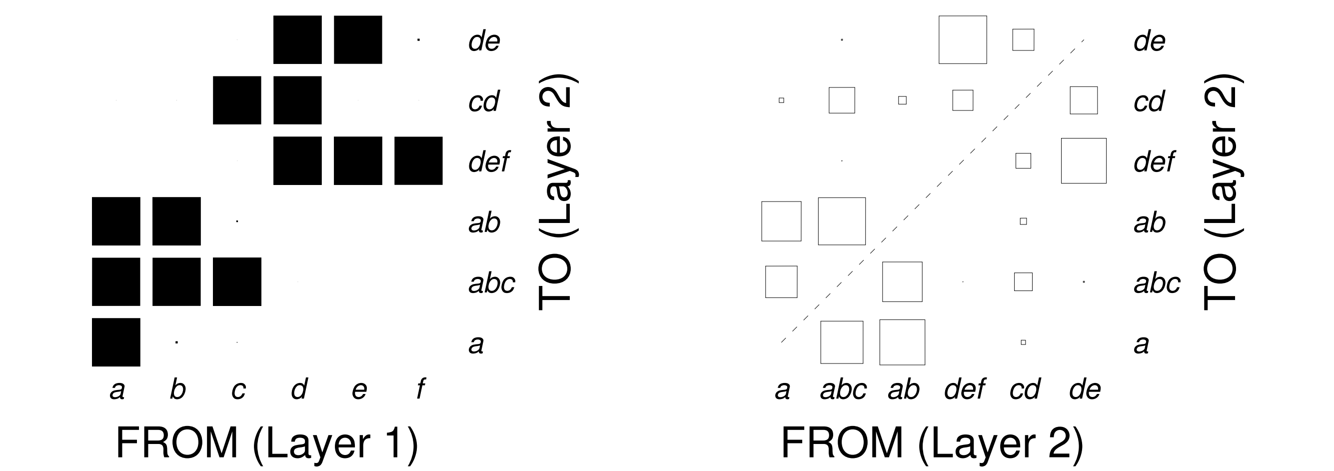

Figure 4: Excitatory and inhibitory weights in Simulation I. Weight values are proportional to the length of the sides of each square. Excitatory weight values of Layer 1→Layer 2 connections are indicated by filled squares. Inhibitory weight values of Layer 2→Layer 2 connections are indicated by open squares.

When each input pattern was presented separately to the completely developed network of Simulation I, only the single corresponding neuron coding the whole pattern became active – not neurons coding a subset or superset of the pattern. For example, when pattern *abc* was presented, neuron *abc* became fully active and inhibited the activity of the other Layer 2 neurons. When *ab* was presented, neuron *ab* became fully active and inhibited the activity of other Layer 2 neurons, including that of neuron *abc*. In this manner, the network exhibits the Gestalt property of representing "wholes" as different from the sum of the "parts." The sequence masking property (Cohen & Grossberg, 1986, 1987; Grossberg, 1978, 1986) is thus upheld: a neuron representing a whole pattern suppresses other neurons representing superset, subset, and overlapping patterns.

The lateral inhibitory connections within an EXIN network represent an adaptive *constraint satisfaction network*; they capture the allowed and disallowed combinations of Layer 2 neuron activations. The additional tests described below illustrate some novel characteristics of such EXIN constraint satisfaction networks.

4. Parsing of Multiple Superimposed Patterns

Further tests were conducted examining the behavior of the network when unfamiliar input patterns were presented. When the unfamiliar input pattern *abcd* was presented to the completely developed network of Simulation I, both neurons *ab* and *cd* became fully active (Figure 3C), representing the network's recognition of the simultaneous presence of *both* the familiar patterns *ab* and *cd*. The network *parses* the unfamiliar pattern *abcd* in terms of the familiar patterns. The simultaneous distributed activation, or *multiplexing* (Grossberg & Marshall, 1989; Marshall, 1989ab, 1990acdef, 1991, 1992ab), of multiple codes was possible because of the low degree of reciprocal inhibition between neurons *ab* and *cd*.

EXIN networks thus allow *multiple* patterns to be represented simultaneously, in a distributed but highly regulated fashion. The new unsupervised inhibitory learning rule produces a SONN structure that allows multiple neurons to "win" a network competition, instead of forcing a single winner to "take all." Thus, when multiple independent input patterns are present, all of them can be represented by distinct activations. The excitatory and inhibitory learning rules fulfill different but complementary roles: they provide both a means by which perceptual groupings can be learned (EX) and a means by which the learned groupings can be either engaged competitively with or disengaged from one another (IN).

What constitutes a *pattern*? In an EXIN network, a given combination of inputs acquires its identity as a unitary pattern only if it is presented sufficiently often. The EXIN rules have the dual effects of identifying patterns by frequency of occurrence, or familiarity, and then building a mechanism that chooses a near-optimal representation of multiple superimposed patterns, by simultaneous distributed activation of multiple Layer 2 neurons.

5. Global Context-Sensitive Constraint Satisfaction

Context is an important determinant of the interpretations of perceptual data. However, WTA neural networks handle contextual information poorly. EXIN networks learn to process perceptual information in a context-sensitive manner (Marshall, 1992ab), as described below.

By definition, WTA neural networks are capable of representing each input pattern only as a single, lumped item. Suppose that a simple WTA neural network has learned to recognize three patterns: *ab*, *abc*, and *cd* (Nigrin, 1990a) (Figure 5B). The Layer 1 neurons a, b, c, d might represent the speech sounds ô, l, tĕr, n in utterances like "all," "alter," and "turn" (Nigrin, 1993). When one of these input patterns (say *abc*) is presented to Layer 1 of this neural network (by activating neurons a, b, and c), the corresponding Layer 2 neuron (labeled *abc*) becomes active, thereby indicating that the pattern is recognized.

Now suppose that occasionally the input pattern to the neural network is *abcd* ("all turn"). How is this pattern coded? In a WTA neural network, the best response is to activate neuron *abc*, because it represents the closest match to the input pattern (Figure 5E). Activation of other Layer 2 neurons is suppressed. However, such a response essentially ignores the presence of d, as if d were merely noise.

But in an EXIN network trained to recognize the same three patterns, lateral inhibitory strengths between neurons *ab* and *cd* become weakened (Figure 5D) according to the inhibitory learning rule, because patterns *ab* and *cd* do not overlap at all (thus, the neurons tend not to receive simultaneous excitation). On the other hand, inhibitory strengths between *ab* and *abc* and between *abc* and *cd* remain strong (Figure 5D,G,J), because the input patterns that they code overlap substantially.

When *abcd* is presented to an EXIN network (Figure 5G), *both ab* and *cd* become active, and *abc* becomes inactive. Neuron *abc* receives inhibition from both *ab* and *cd*, whereas *ab* and *cd* each

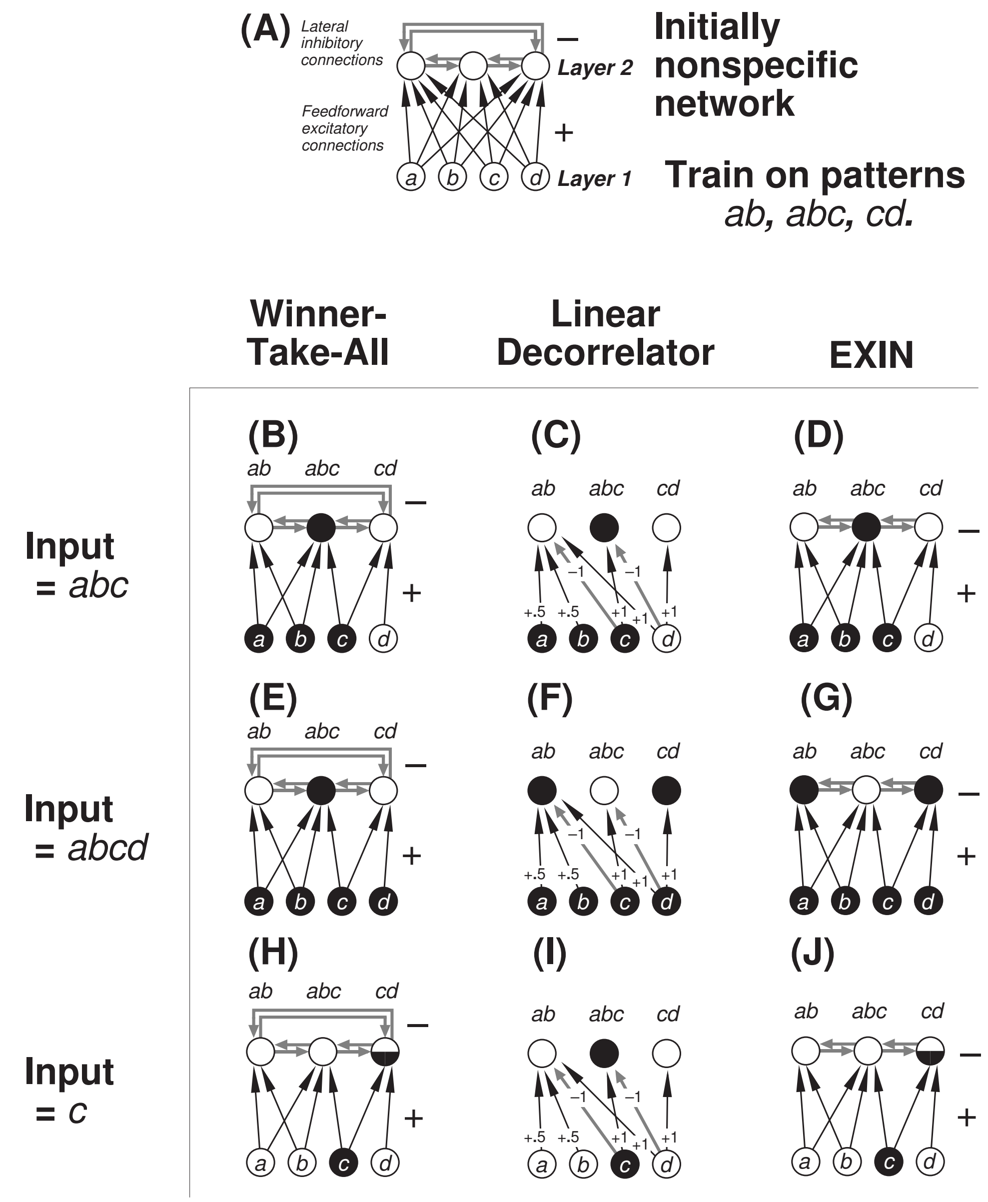

Figure 5: Response of various networks. (A) Initially, neurons in Layer 1 project excitatory connections nonspecifically to neurons in Layer 2. In addition, each neuron in Layer 2 projects lateral inhibitory connections nonspecifically to all its neighbors (lateral arrows). (B,C,D) The excitatory learning rule causes each type of neural network to become selective for patterns *ab*, *abc*, and *cd* after a period of exposure to those patterns; a different neuron becomes wired to respond to each of the familiar patterns. Each network's response to pattern *abc* is shown. (E) In the WTA neural network, the compound pattern *abcd* (filled lower circles) causes the single "nearest" neuron (*abc*) (filled upper circle) to become active and suppress the activity of the other Layer 2 neurons. (G) In an EXIN network, the inhibitory learning rule weakens the strengths of inhibitory connections between neurons that code nonoverlapping patterns, such as between neurons *ab* and *cd*. Then, when *abcd* is presented, both neurons *ab* and *cd* become active (filled upper circles), representing the simultaneous presence of the familiar patterns *ab* and *cd*. (F) Linear decorrelator network responds similarly to EXIN network for input pattern *abcd*. However, in response to the unfamiliar pattern *c*, both WTA (H) and EXIN (J) networks moderately activate (partially filled circles) the neuron whose code most closely matches the pattern (*cd*), whereas the linear decorrelator network (I) activates a more distant match (*abc*). (Reprinted with permission from Marshall, 1992a.)

receive inhibition *only* from *abc*. Because *abc* receives more inhibition, its activation is suppressed, and both *ab* and *cd* can become active. The simultaneous activation of both neurons *ab* and *cd* is made possible because of the weakened reciprocal inhibition between them.

The simultaneous activation of neurons *ab* and *cd* represents the EXIN network's recognition of the superimposed familiar patterns *ab* and *cd*. The EXIN network thus chooses a more complete representation of the input than is possible in the WTA neural network. The contextual presence of the single item *d* dramatically alters the multiplexed *parsing* of the input. When *abc* is presented, the network groups *a*, *b*, and *c* together as a unit, but when *d* is added, the network breaks *c* away from *a* and *b* and binds it with *d* instead, forming two separate groupings, *ab* and *cd*. This radical alteration of parsing depending on the presence/absence of small distinguishing features, or nuances, (like *d*) constitutes the EXIN network's *context-sensitivity* property. In a specific domain like visual perception, such contextual information can determine the segmentation or grouping of a set of visual features.

EXIN networks can even suppress the activation of some of the most-excited Layer 2 neurons (such as *abc* in Figure 5G) when necessary to achieve a maximally context-sensitive, global representation of the input within the learned environmental constraints. This ability makes EXIN networks a true form of *distributed* coding, unlike k-winner networks, which merely activate the k most-excited Layer 2 neurons, regardless of context, coactivation, or familiarity. This *global context-sensitive constraint satisfaction* property is extremely important in perceptual processing because it lets globally optimal representations overcome locally optimal ones.

EXIN networks implement a form of *exclusive allocation* (Bregman, 1990), whereby "a sensory element should not be used in more than one description at a time" (Bregman, 1990, p. 12). A sensory element can potentially participate in many different pattern groupings. For example, Layer 1 neuron *b* projects excitatory connections to several different Layer 2 neurons – *ab*, *abc*, and *bc*. But because activation of Layer 1 neuron *b* always causes *ab*, *abc*, and *bc* to receive simultaneous excitation, the inhibitory learning rule causes strong inhibitory connections to develop between *ab*, *abc*, and *bc*, forcing them to compete. These inhibitory connections enforce an exclusive allocation constraint, preventing *ab*, *abc*, and *bc* from becoming fully active at the same time. A fully active Layer 2 neuron receives "credit" for representing a Layer 1 neuron's activation by becoming fully active and thereby being selected to have its input connection weights adjusted.

The distributed representations allowed in EXIN networks also make them more efficient than k-winner networks, in the following sense: conjunctions (like *abcd*) of familiar patterns (ab, cd) do not require additional neurons to be accurately represented. To represent *abcd* as distinct from *abc*, a k-winner network would need an extra Layer 2 neuron, devoted to coding *abcd*. In fact, it would need an extra neuron for *every* possible conjunction of input patterns, whereas the EXIN networks can multiplex the coding of conjunctions by simultaneous activation of several neurons. The multiplexed distributed coding in EXIN networks therefore avoids or at least alleviates the "grandmother cell" dilemma of combinatorial explosion.

6. Uncertainty in Pattern Classification Systems

6.1. EXIN Network's Response to Ambiguous Patterns

Tests were also conducted examining the network's behavior in response to ambiguous input patterns. When pattern *d* was presented to the fully developed network of Simulation I, both neurons *cd* and *de* became moderately active (Figure 3D). Neurons *cd* and *de* represent the "nearest" familiar patterns to input pattern *d*. The activity of other neurons, including that of neuron *def*, was suppressed. Both neurons *cd* and *de* remain active because they receive roughly equal amounts of excitation and because they project only moderately strong reciprocal inhibition. The network represents its *uncertainty* about the classification again by multiplexing – simultaneously activating multiple codes for the input pattern.

The ability to represent multiple hypotheses about the classification of an ambiguous pattern is an extremely useful network property (Marshall, 1989ab, 1990acdef, 1991, 1992ab; Martin & Marshall, 1993; Szeliski, 1988). For instance, in various perceptual modalities, a pattern unfolds as a temporal stream (e.g., a spoken word), for which complete classification is initially uncertain. Only as new information arrives can the complete pattern be classified (e.g., My *vs.* Myself). In some perceptual environments, a network can successfully self-organize only when multiple hypotheses about a feature's uncertain classification can be simultaneously represented (Marshall, 1989ab, 1990ad; Martin & Marshall, 1993). If the network were forced to make WTA decisions too early, when it lacks adequate classification information, its decisions would often be incorrect, preventing the correct connections from developing. Below it is explained how an EXIN network's ability to represent multiple hypotheses, endowed by the inhibitory learning rule, allows it to sustain the activation sequences that it needs to acquire the proper connection weights.

6.2. Wrong Winners Lead to Wrong Learning

Perceptual uncertainty arises from a variety of sources, both internal and external. For example, an out-of-focus or distant image of a person's face might look like one of many familiar faces; its identification or classification is thus uncertain. In neural networks, uncertainty occurs when an incomplete, noisy, or ambiguous incoming signal pattern has more than one likely classification.

In complex, real-world environments, perceptual information is often initially ambiguous. Perceptual uncertainty can be resolved in such cases by the subsequent addition of disambiguating information. For instance, in vision, a monocular image is generated by the projection of a 3-D scene onto a 2-D retina. A given feature in the image could have been generated by an object at any depth in the scene. Until further information is added, the visual system cannot necessarily determine the feature's depth. Such additional information can be provided by motion parallax transformations, top-down size familiarity, or other cues. The addition of depth cues lets the visual system resolve uncertainty about the object's depth.

If a network makes a WTA "guess" about the classification of an ambiguous input pattern, its decision may later turn out to be wrong, as disambiguating information is added. Activation of an incorrect Layer 2 neuron may not, by itself, be bad. After all, the network could just pick a winner, and then if subsequent information warrants, change its earlier decision. However, the network's *learning* is a function of neuron activations. Because the operation of Hebbian-type excitatory learning rules (Grossberg, 1976ab; Hebb, 1949) is based on correlations in neuron activity, the wrong choices of active neurons could lead to wrong learning, thereby impairing or even preventing the development of stable pattern codes.

Ambiguous environments can thus seriously undermine the adaptive development of WTA networks. One method by which the wrong-winner problem may be solved is to allow the network to maintain a representation of its own uncertainty.

6.3. Classification of Ambiguous Patterns

These problems with WTA classification of ambiguous input patterns, and possible solutions using EXIN networks, are delineated in the following example. Suppose that too little inhibition is initially present between a set of Layer 2 neurons. Then an input pattern would activate multiple Layer 2 neurons (Figure 6A). Each active Layer 2 neuron would then learn the *same* input pattern (Figure 6B), thereby defeating the network's purpose as a self-organizing classifier. Because each Layer 2 neuron ought to acquire a different sensitivity, the inhibition weights between the Layer 2 neurons need to be strong enough (at least initially) to produce selectivity, so that

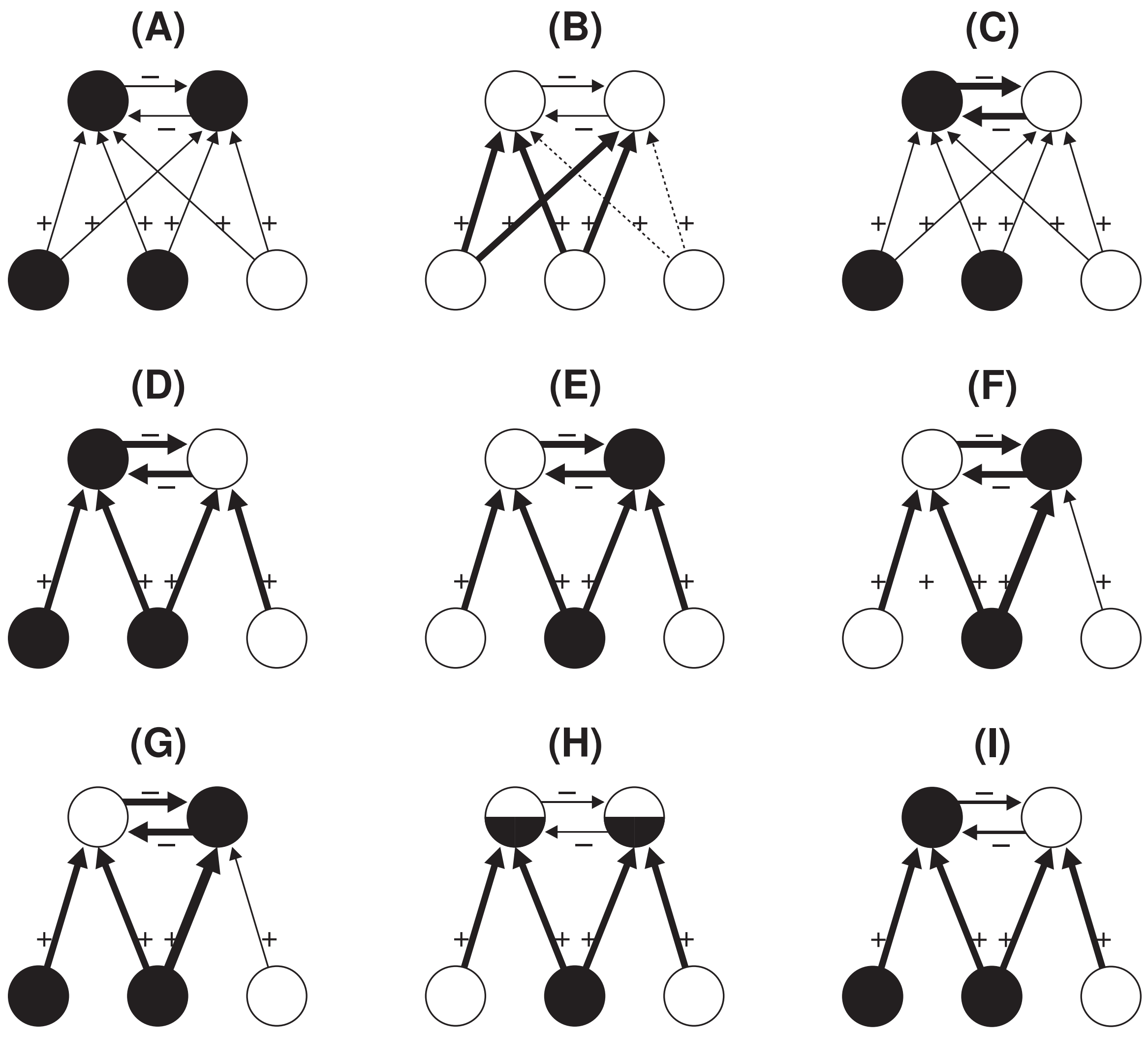

Figure 6: Wrong winners. (A) An input pattern (filled lower circles) across Layer 1 excites Layer 2 neurons (upper circles). The Layer 2 neurons are initially nonspecific (i.e., they all receive connections with roughly the same weights from the neurons in Layer 1). Inhibition between Layer 2 neurons is weak (thin horizontal arrows), so that all such neurons become active (filled circles) in response to the input pattern. (B) Learning would then cause the excitatory connections from the active Layer 1 neurons to the active Layer 2 neurons to strengthen (thick arrows), while the excitatory connections from the inactive Layer 1 neurons weaken (dotted arrows). Because the Layer 2 neurons were both active, they both learned the same input pattern. (C) If inhibition is strong (thick horizontal arrows), then only one Layer 2 neuron can become active at a time. (D) Each neuron then acquires its own sensitivities after repeated exposure to input patterns. (E) But then, if an ambiguous input pattern is presented (could be in either category), only one Layer 2 neuron can respond. (F) This leads to unwanted distortions of connection weights (thin and thicker arrows). (G) Even if the input pattern is disambiguated by subsequent additional information, the correct Layer 2 neuron may be unable to overcome hysteresis from the incorrect neuron's prior activation. (H) But if inhibition weight is then reduced (thinner arrows), multiple Layer 2 neurons could simultaneously respond to the ambiguous input pattern, possibly at a lower activation level (partially filled circles). (I) Then the new disambiguating information could cause the correct Layer 2 neuron to win, suppressing incorrect classifications. (Reprinted with permission from Marshall, 1990d.)

slight differences in neuron input levels result in great differences in neuron activations (Figure 6C). Over a sufficient number of exposures to input patterns, each neuron tends to acquire a different sensitivity (Grossberg, 1976ab; Kohonen, 1984) (Figure 6D).

However, if an initially ambiguous pattern (e.g., the intersection of two familiar patterns) is then presented (Figure 6E), then the strong inhibition forces the network to make an immediate choice (which may later turn out to be wrong) of a single active neuron. As learning proceeds, the development of the proper connection weights could be disrupted (Figure 6F). Furthermore, if additional disambiguating parts of the input pattern subsequently become available, then the correct Layer 2 neuron might be unable to overcome feedback inhibition from an incorrect neuron (Figure 6G). Thus, pattern ambiguity can cause *instability* in the structure of Hebbian-type WTA networks.

How should the network respond to ambiguous input patterns so that its behavior and structure remain stable? The problem stems from the WTA nature of the network. The problem might be solved if more than one neuron were allowed to remain active in response to an ambiguous input pattern, for instance by reducing the amount of inhibition (Figure 6H). Then representations of all possible correct classifications for the input could be maintained. The simultaneous activity of multiple neurons constitutes a representation of the network's uncertainty about the correct classification of the input pattern. Subsequent disambiguating information could enhance the activity of a single correct Layer 2 neuron, which in turn would more strongly inhibit alternate Layer 2 neurons (Figure 6I).

6.4. Weakening Inhibition to Permit Coactivation

How can the nearly identical neurons in a nonspecific network become differentiated, in the presence of ambiguous input patterns? Can the conflicting requirements for both strong inhibition (to produce selectivity) and weak inhibition (to allow coactivation) be reconciled? These questions can be answered in two parts.

First, strong inhibition is mainly needed only at the *outset* of the network's development, to ensure that all neurons do not respond to all input patterns. Afterward, according to the excitatory learning rule, the neurons' input pattern sensitivities become incorporated into the *excitatory* connection weights. Thereafter, less inhibition is needed, because a given input pattern tends not to fully coactivate the neurons anyway. The behavior of the inhibitory learning rule permits EXIN networks to choose appropriate intermediate levels of inhibition and thereby represent uncertainty via coactivation of multiple neurons.

Second, processing of inhibition could be specified in a manner so that inhibition causes WTA behavior only when the neuron is fully activated (Nigrin, personal communication, 1993). One way to implement this specification would be to rewrite the inhibitory summing equation (9) with a faster-than-linear half-rectification like $\max(0, x_j)^2$, instead of the linear half-rectification $\max(0, x_j)$. Then, if a coded pattern is perfectly represented, the corresponding neuron will become fully activated and will fully inhibit the activation of the other neurons. However, if a pattern is ambiguous, then no neuron will become fully active; the weaker inhibition will not fully shut off other neurons. Neurons coding similar patterns can thus become coactivated when none of the neurons receives its full excitatory input.

Suppose that initially the reciprocal inhibitory connections between two neurons are quite strong. Then even if both neurons have similar excitatory input connections, only one of the neurons, in general, becomes active in response to an input pattern. The two neurons are unlikely to become coactivated. Hence, according to the inhibitory learning rule, their reciprocal inhibitory connection weights are likely to weaken gradually. Meanwhile, each of the two neurons acquires a different excitatory input connection profile and begins to respond to a different pattern. Now, when an ambiguous pattern is presented, the reduced inhibition between the two neurons permits them *both* to become partially activated. Multiple *hypotheses* about the classification of a pattern are thereby represented.

On the other hand, if two neurons both respond frequently to the same pattern, then the weights of their reciprocal inhibitory connections tend to increase, reducing the likelihood that they will be coactivated in the future. The network's efficiency is thus promoted, because no two neurons can acquire the same pattern sensitivity.

6.5. Reduced Learning of Ambiguous Patterns

One advantage of such a multiplexed representation scheme is that the deleterious effects of wrong classifications on a network's learning can be reduced. If excitatory learning is allowed to occur primarily when a neuron is *fully* active (but not when it is partially active), then the network's structure can be altered only when such changes are fully warranted. This rule can be implemented by a faster-than-linear sampling function like $f(x_i) = \max(0, x_i)^2$ in the excitatory learning equation (6). Thus, the wrong-winner problem is resolved by the technique of allowing multiple neurons to become partially active under uncertainty.

Using an inhibitory learning rule in conjunction with a standard excitatory rule permits greater flexibility in representing both uncertainty and decision in pattern classification tasks. The EXIN inhibitory rule helps the process of self-organization operate stably in realistic situations, where ambiguous pattern information becomes completed and refined over time.

7. Further Simulation Results

7.1. Simulation I(a): Test on All Binary Patterns

Figures 7–8 show the multiplexed, context-sensitive response of the EXIN SONN of Simulation I to a variety of familiar and unfamiliar input combinations. All 64 possible binary input patterns were tested, and reasonable results were produced in each case. For example, the bottom row, third column of Figure 7 shows that pattern adf was parsed as a + (partial)def. A comparison of the network's responses to abc and $abcd$ demonstrates that the EXIN network of Simulation I indeed has learned to perform the context-sensitive parsing described in Figure 5D,G,J. Input pattern abc was parsed as abc, whereas $abcd$ was parsed as $ab + cd$.

Although there are no "correct" parsings (except for the networks's response to the six training patterns plus the null pattern), the EXIN network's responses to the 64 test patterns are reasonable, given the the six training patterns. These 64 parsings are a key result of this paper. The behavior of the network in response to many of the patterns can be altered somewhat by changing some of the network parameter values. However, in exploratory tests of EXIN networks, the general properties of sequence masking, multiplexed distributed coding, global context-sensitive constraint satisfaction, and uncertainty multiplexing have been robust with respect to such manipulations.

7.2. Simulation I(b): Test on Some Analog Patterns

The response of the same network was tested on analog patterns. The network was presented (after *training* on the six binary patterns) with a set of *test* input patterns drawn from a 1-D subspace environment of the space of all possible analog patterns. The binary patterns a, ab, abc, cd, de, and def were considered to be points equally spaced along a 1-D real-valued continuum ring (with wrap-around). Each analog pattern was selected from the continuum by choosing a real-valued number; if the number fell between the points corresponding to two of the binary patterns, then the analog pattern generated was a linearly weighted mixture of the two patterns. The mixed pattern was then normalized so that the greatest input neuron activation was 1. For example, the pattern corresponding to the real value 0.8 would fall between a (located at 0.0) and ab (located at 1.0); it would be a mixture of 20% a and 80% ab. This would generate analog activation values of 1.0 for Layer 1 neuron a and 0.8 for Layer 1 neuron b. This method did not generate all possible analog patterns, just a subset to probe the network's behavior.

With this method, "tuning curves" of the Layer 2 neurons could be plotted, by recording their response to successive analog patterns drawn from the continuum. The tuning curves of

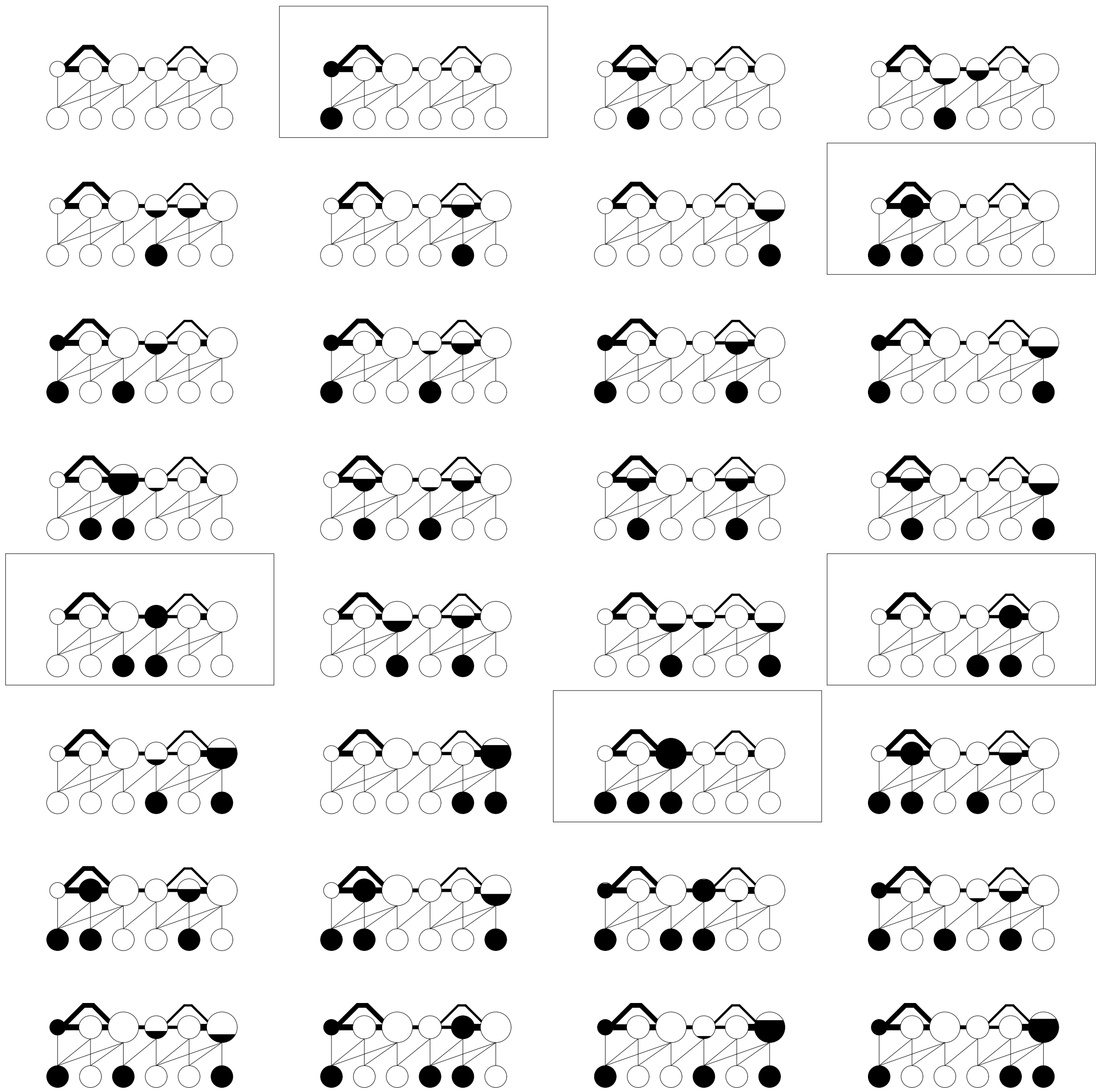

Figure 7: Parsing of all 64 possible binary input patterns (Patterns 1–32). In Figures 7 and 8, 64 copies of the EXIN network of Figure 3B are shown. Each copy of the network illustrates the network's response to a different input pattern. The neurons are not labeled here (see Figure 3B for the labels of the neurons). Strong excitatory connections (weights between 0.992 and 0.999) are indicated by lines from Layer 1 neurons (lower) to Layer 2 neurons (upper). All other excitatory connections (weights between 0 and 0.046) are omitted from the figure. Strong reciprocal inhibitory connections (weights between 0.010 and 0.033) are indicated by lines from Layer 2 neurons to other Layer 2 neurons. The thickness of these lines is proportional to the inhibitory connection weights. All other inhibitory connections (weights between 0 and 0.006) are omitted from the figure. Input patterns are indicated by filling of active Layer 1 neurons (lower). Network responses are indicated by filling of active Layer 2 neurons (upper); fractional height of filling within each circle is proportional to neuron activation value. Radius of circle is proportional to value of neuron's "size" (sensitivity) normalization factor. Rectangles are drawn around the networks that indicate the responses to the six training patterns. Study of Figures 7 and 8 reveals key aspects of EXIN network behavior, including sequence masking (a, ab, abc), multiplexed distributed coding ($abcdef$), global context-sensitive constraint satisfaction ($abcd$ vs. abc), and uncertainty multiplexing (d).

all six Layer 2 neurons are superimposed and displayed in Figure 9. Each of the six binary input patterns caused a different Layer 2 neuron to become fully active.

Note that all the "petals" of Figure 9 are separated by a gap, with the exception of the petals for the neurons coding a and def. The gaps arose because the neurons coding successive patterns

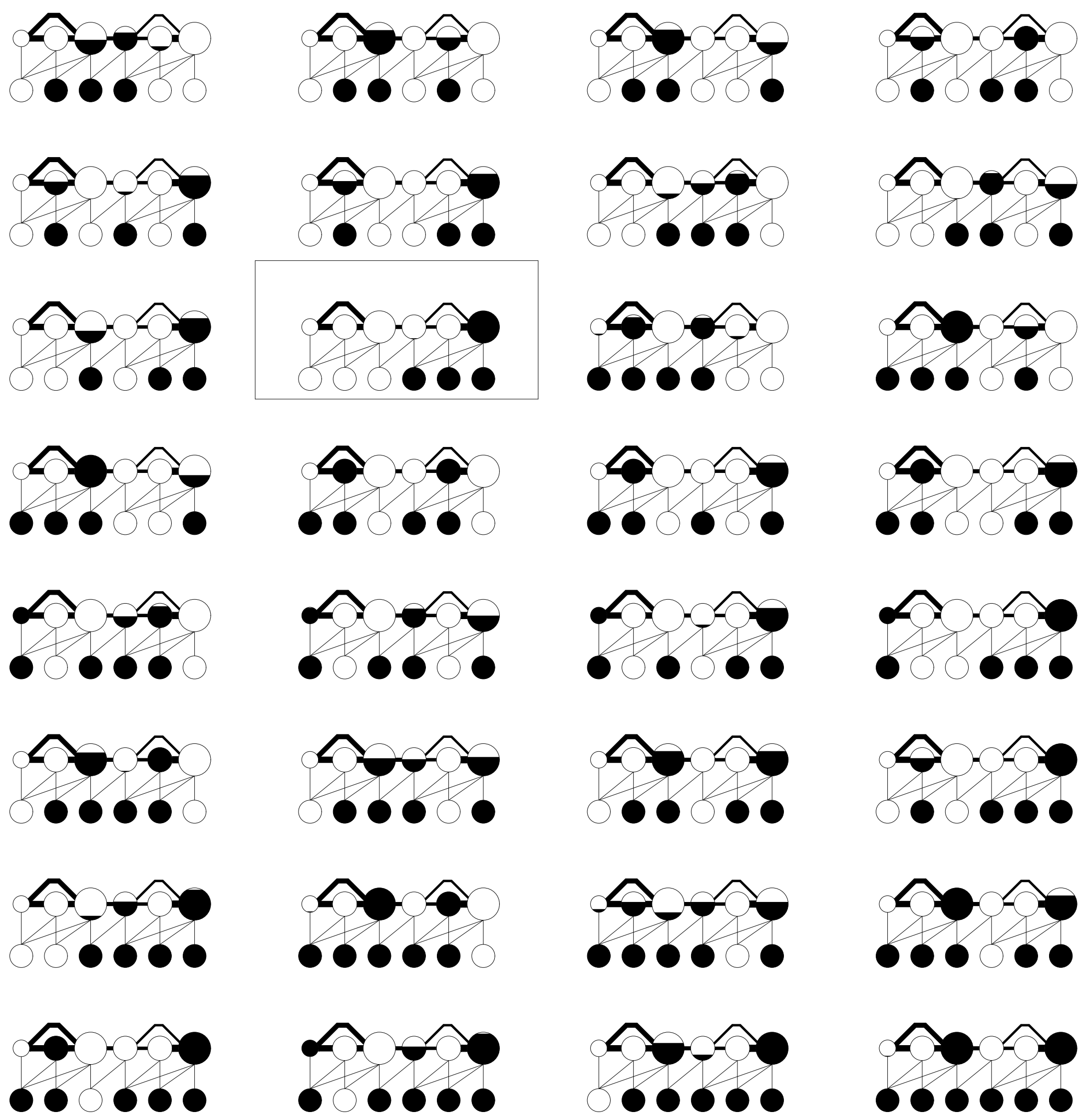

Figure 8: Parsing of all 64 possible binary input patterns (Patterns 33-64). (See caption for Figure 7.)

(like ab and abc) receive input connections from some Layer 1 neurons in common. Thus, those neurons projected strong reciprocal inhibitory connections and tended not to be coactivated. In the case of a and def, however, the corresponding neurons did not receive input connections from common Layer 1 neurons; thus, they projected only relatively weak inhibitory connections. When blended patterns (like $adef$) were presented, the neurons coding a and def activated independently, according to the respective strengths of input patterns a and def; therefore, those petals overlap. This coding by simultaneous activation of multiple neurons can be interpreted as a form of distributed coding of the analog patterns.

Note further that the procedure used in Simulation I(b) (and in Simulation II) is not ecologically realistic, in that the training data and test data are drawn from different statistical distributions. This illustrates that EXIN networks can respond reasonably to analog input patterns even when they are trained only on binary input patterns. Simulations I(a), III, and IV were run under the more realistic assumption that the neurons' training environment and operational environment are identical.

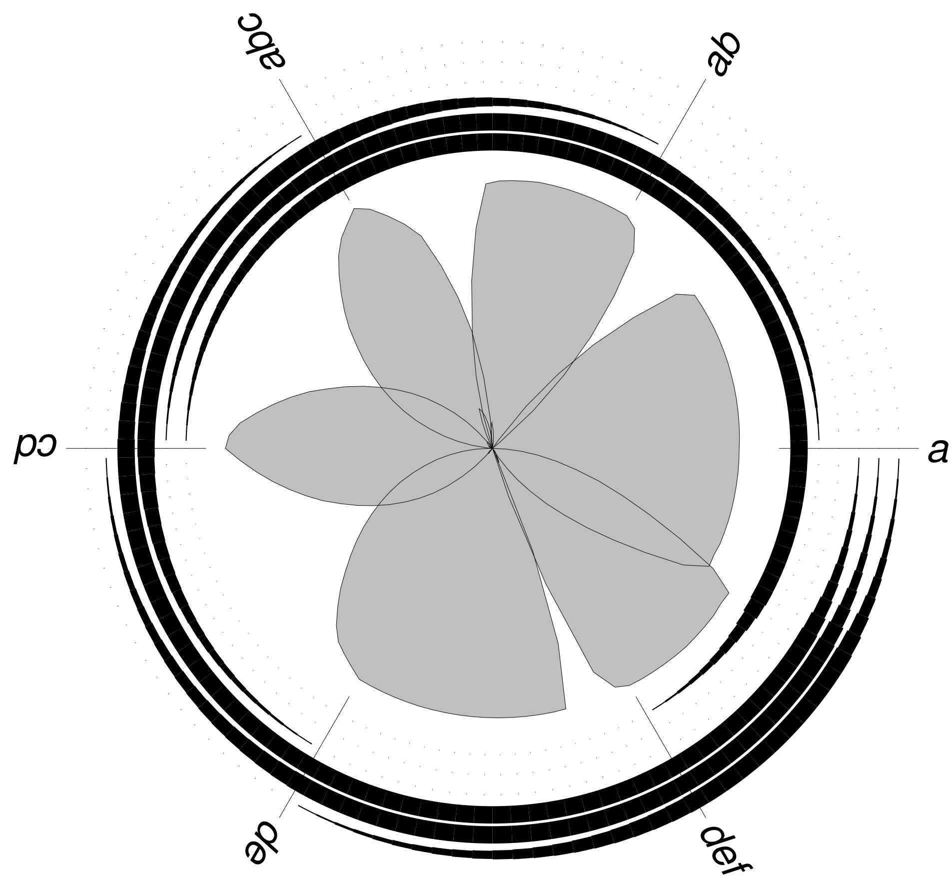

Figure 9: Tuning curves for Simulation I(b). Each angle represents an input pattern along the 1-D input pattern continuum. Thickness of each of the six outer concentric bands represents the activation value of the corresponding Layer 1 neuron (network *input*). The innermost band represents the activation of Layer 1 neuron a; the outermost band represents the activation of Layer 1 neuron f. Each Layer 2 neuron's response (network *output*) is plotted as the radius of the shaded region, at each angle. Responses of all six neurons in Layer 2 are superimposed. Note that a different neuron responds to each of the six training patterns.

7.3. Simulation II: Seven Overlapping Binary Patterns

In Simulation I, if *abcd* is presented only on rare occasions, then the network's structure remains stable. But if it is introduced often enough into the network's input stream, as shown by Simulation II, then it becomes processed as a new whole in its own right, appropriating one of the Layer 2 neurons and forcing two other patterns to share a Layer 2 neuron (Figure 10). In this example, patterns a and ab both caused the same neuron to become active. (Likewise, if d were presented often enough, then it too would appropriate its own Layer 2 neuron.) In Simulation II, the network was exposed during its development to seven binary input patterns: $a, ab, abc, abcd, cd, de, def$. Then it was tested on analog patterns drawn from a 1-D continuum environment that includes the seven binary patterns, as was done for Simulation I(b).

7.4. Simulation III: Six Overlapping Analog Patterns

The network of Simulation III was both trained *and* tested on the analog input patterns, constructed as described above for Simulation I(b). Points were randomly selected on the 1-D continuum and were used to generate blended analog patterns; the network was exposed to these patterns during its development. Unlike Simulations I and II, there is in effect a virtually infinite number of patterns from which the training inputs to Simulation III can be drawn.

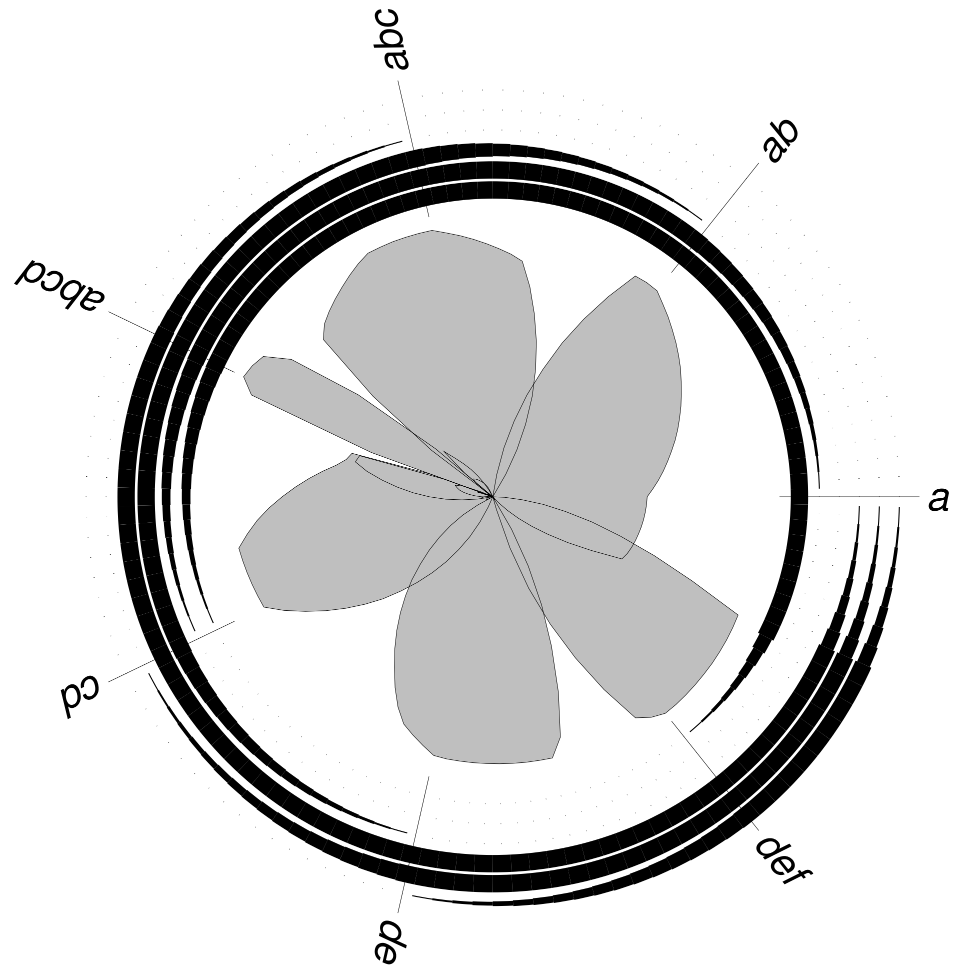

Figure 10: Tuning curves for Simulation II. (See Figure 9 for description of plot format.) Here the network was repeatedly given seven input patterns in random order during its training. Because only six neurons were available in Layer 2, patterns a and ab both caused the same Layer 2 neuron to become activated.

Simulations III and IV are designed to probe the network's response to the statistical *distribution* of inputs in different environments. Figure 11 shows the tuning curves of the six Layer 2 neurons in Simulation III. Note that the six base patterns a, ab, abc, cd, de, def have lost their ontological status as "familiar" patterns in Simulation III because they were no more likely to have been presented than any other pattern drawn from the continuum. One consequence is that the small pattern a is not coded directly by any single neuron; instead, the larger pattern $adef$ is coded by a neuron. The Weber Law scaling rule gives somewhat more weight ($\frac{4}{5}$) to $adef$ than to a ($\frac{1}{2}$).

7.5. Simulation IV: Skewed Temporal Distributions and Overlaps

Simulation IV is identical to Simulation III, except that the statistical distribution across the six patterns is no longer uniform. Patterns a and ab are presented substantially more often than the other four patterns, as indicated by the width of the angles corresponding to the patterns in Figure 12. Despite the skewing of the input distribution, the network still mostly distributed its codes according to the similarities and differences between the patterns, rather than the frequencies of the patterns. Thus, this EXIN network was largely insensitive to the skew of the training input pattern distribution.

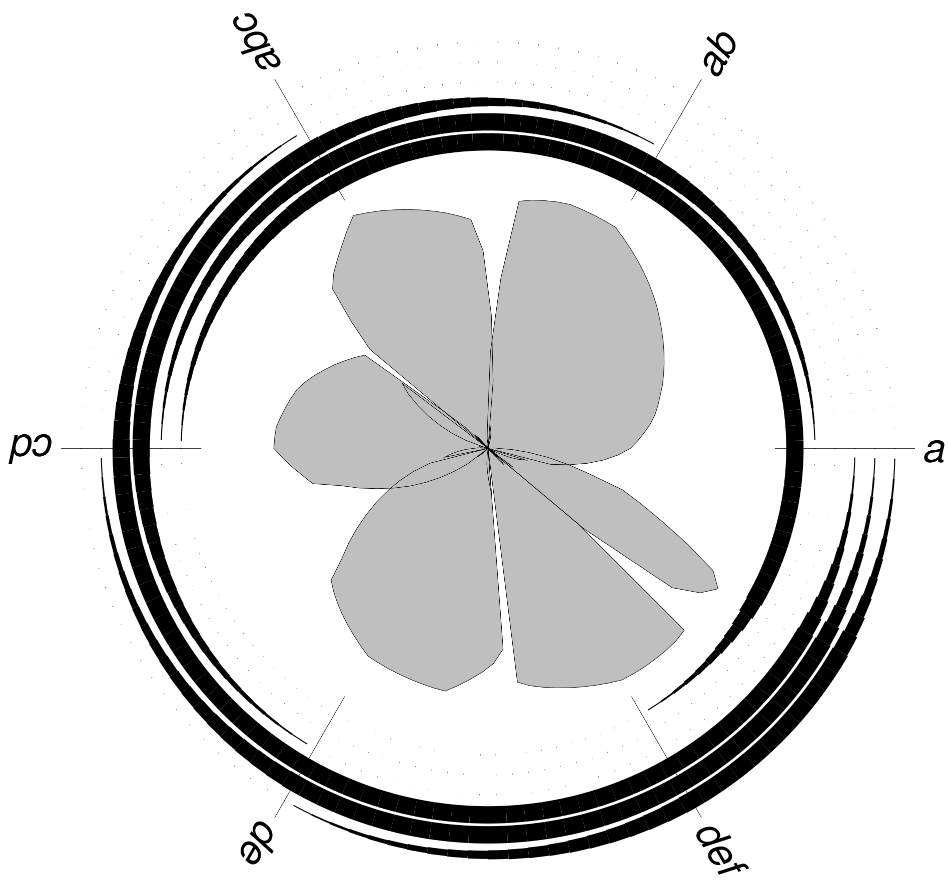

Figure 11: Tuning curves for Simulation III. (See Figure 9 for description of plot format.)

8. Discussion

8.1. Basis for Inhibitory Selectivity: Temporal Overlap and Common Input

The inhibitory selectivity provided by EXIN networks implements adaptively a fundamental and general principle of perception: the principle of exclusive allocation (Bregman, 1990), or *credit assignment* (Barto, Sutton, & Anderson, 1983). That is, the "credit" for a given input feature is "assigned" exclusively to a single representation. The learned inhibitory constraints tend to prevent more than one neuron strongly excited by the feature from becoming active simultaneously. The possible representations of a given feature are thus forced to compete with one another.

The inhibitory learning rule provides a basis for selectivity of inhibition: *temporal overlap*, or coactivation. If two neurons are frequently coactivated, then it is likely that they represent similar patterns; the inhibition between them then increases, thereby enhancing the selectivity of the network. Inhibition declines where it is not needed. One way in which two neurons might be frequently coactivated is if they receive *common input* – a Layer 1 neuron projects excitatory connections to both of them. The inhibitory learning rule ensures that all Layer 2 neurons that receive strong input connections from a common Layer 1 neuron will compete strongly with one another. Exclusive allocation constraints then arise by virtue of the selective inhibition within Layer 2. Conversely, lack of common input leads to weaker inhibition, which leads to the possibility of simultaneous activation of multiple Layer 2 neurons.

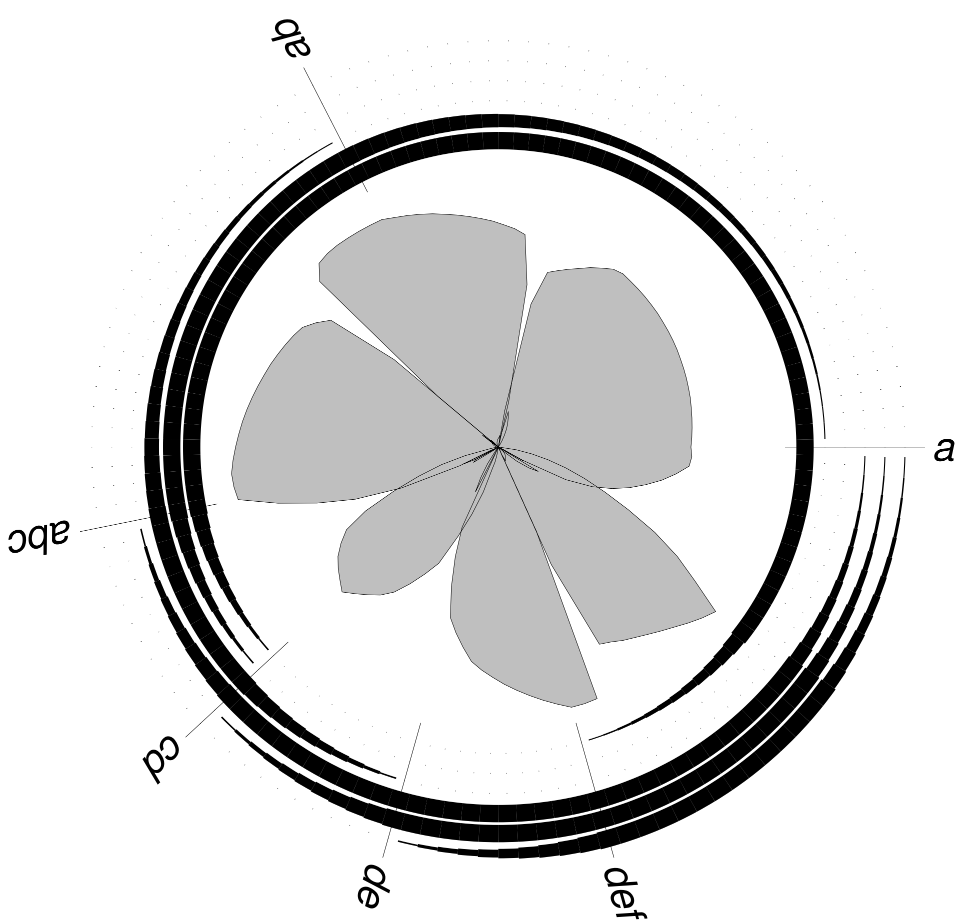

Figure 12: Tuning curves for Simulation IV. (See Figure 9 for description of plot format.)

8.2. Benefits of Exclusive Allocation

Exclusive allocation is a desirable property because it prevents any given piece of data from counting as evidence for *multiple* patterns simultaneously (Bregman, 1990). There are many examples from vision and other perceptual modalities where a given datum should be allowed to count as evidence for only *one* pattern at a time. The credit for a given datum should be assigned to a single pattern.

A good example of why exclusive allocation or credit assignment is a fundamental part of perceptual pattern recognition comes from visual stereopsis, where a visual feature seen by one eye can potentially be matched with many visual features seen by the other eye (the "correspondence problem"). Our visual systems assign at most one unique binocular match for each such monocular visual feature; this property is known as the uniqueness constraint (Marr, 1982; Marr & Poggio, 1976).

Many pattern recognition algorithms, e.g., the Hough Transform (HT), lack a credit assignment capability (Duda & Hart, 1972; Hough, 1962). The HT gathers evidence across space for the presence of a visual pattern, such as a binocular feature match or the presence of an oriented edge. Without an exclusive allocation capability, a feedforward algorithm like the Hough Transform, applied to a stereo-matching problem, would detect *all* the possible binocular feature matches, without a way to assign a single match to each feature.

On the other hand, an EXIN network maintains lateral feedback inhibition between neurons that code conflicting matches for any single feature and drops inhibition between neurons that code matches between different features (Marshall, 1990e, 1992b). Its exclusive allocation property thereby implements the stereo uniqueness constraint, forcing the EXIN network to assign at most a single match to each feature.

In visual edge detection, the same principle applies. For example, imagine a noisy image in which a vision system is trying to detect skimpy evidence for the presence of edges. Suppose there is a blob in the image that could belong to one of two edges. Suppose that if the blob is assigned to either edge, there isn't enough evidence to declare the other edge "detected." The Hough Transform would detect *both* edges, rather than choosing one edge at the expense of the other. In a noisy image, this process would lead to a combinatorial proliferation of possible edges being detected.

Thus, the exclusive allocation property helps constrain the number of groupings that a system must examine when it searches for a parsing of an input pattern: certain choices exclude others. The main benefit is parsimony, as Bregman (1990) notes: "It seems as if information is allocated exclusively to one organization or another. This might be a way of guaranteeing the most parsimonious perceptual description of the sensory input" (p. 691).

8.3. Criteria for Evaluating Neural Network Performance

How can one judge whether the parsings that a neural network generates are good ones? On what basis might it be claimed that the parsings that an EXIN network generates are better than those that another network generates? Several criteria (e.g., stability, dispersion, selectivity, convergence, and capacity) for benchmarking neural network performance have been proposed in the literature, and EXIN networks can be evaluated on these criteria. However, this paper introduces a new, additional performance criterion: an exclusive allocation measure. Networks can be evaluated and compared on how well they exhibit exclusive allocation behavior. Together with other criteria, exclusive allocation can be considered a measurement of the quality of parsings that a network generates.

Exclusive allocation describes a desirable way for neural networks to represent patterns. Because it applies to unfamiliar patterns as well as familiar ones, exclusive allocation is a form of *generalization*. For instance, in Figure 5, training on patterns ab, abc, and cd teaches the network how to represent effectively and self-consistently the novel pattern $abcd$.

One way to define an exclusive allocation criterion is to specify how input patterns (both familiar and unfamiliar) ideally should be parsed, in terms of a given training environment (the familiar patterns), and then to measure how well a network's actual parsings compare to the ideal.

Consider, for instance, the network shown in Figure 13, which has been trained to recognize patterns ab and bc. Each Layer 2 neuron is given a "label" (ab, bc) that reflects the *familiar* patterns to which the neuron responds. The parsings that the network generates are evaluated in terms of those labels. When ab or bc is presented, then the "best" parsing is for the correspondingly labeled Layer 2 neuron to become fully active and for the other Layer 2 neuron to become inactive (Figure 13A). When half a pattern is missing (say the input pattern is a), and the other half does not overlap with other familiar patterns, the corresponding Layer 2 neuron should become half-active (Figure 13B). But when the missing half renders the pattern's classification ambiguous (say the input pattern is b), the activation should be distributed equally among the partially-matching alternatives (Figure 13C), which results in two activations at 25% of the maximum level. It would be incorrect to parse pattern abc as ab (Figure 13D), because then the contribution from c is ignored. (That can happen if the inhibition between ab and bc is too strong.) It would also be incorrect to parse abc as $ab + bc$ (Figure 13E) because b would be represented twice. The best parsing in this case would

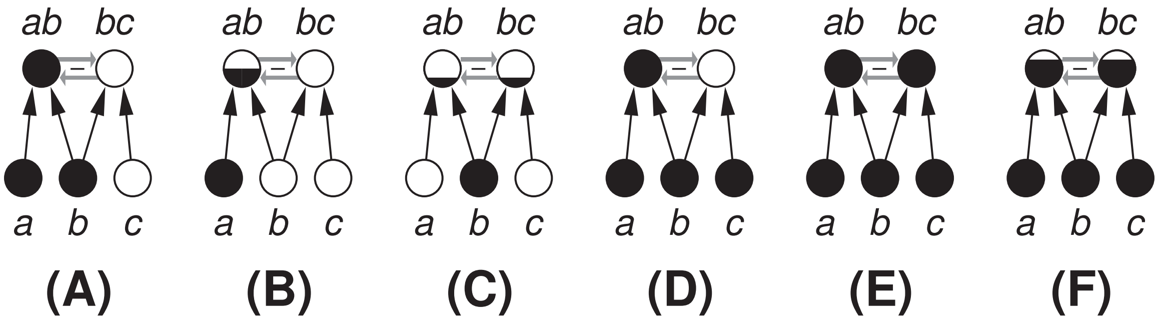

Figure 13: Parsings for exclusive allocation. (A) Normal parsing; the familiar pattern *ab* activates the correspondingly labeled Layer 2 neuron. (B) The unfamiliar pattern *a* half-activates the best-matching Layer 2 neuron, *ab*. (C) Because the unfamiliar input pattern *b* matches *ab* and *bc* equally well, its excitation "energy" is divided equally between the corresponding two Layer 2 neurons, correctly resulting in a 25% activation for each of the two neurons. (D) Incorrect parsing in response to unfamiliar pattern *abc*: neuron *ab* is fully activated, but the energy from input unit *c* is lost. (E) Another incorrect parsing of *abc*: the energy from unit *b* is counted twice, contributing to the full activation of both neurons *ab* and *bc*. (F) Correct parsing of *abc*: the energy from *b* is divided equally between the best matches *ab* and *bc*, resulting in a 75% activation of both neurons *ab* and *bc*.

be to equally activate neurons *ab* and *bc* at 75% of the maximum level (Figure 13F), to represent the uncertain allocation of *b* to *ab* or to *bc*.

Given the examples above, exclusive allocation can be more precisely formulated as the following trio of conditions:

- (1) the input from every Layer 1 neuron is accounted for exactly once in the Layer 2 activations;
- (2) the activation of every Layer 2 neuron is accounted for exactly once by the Layer 1 inputs; and
- (3) when there is more than one best match for an input pattern, they divide the input signals equally.

These three conditions can be formalized mathematically, and a network's parsings can be evaluated quantitatively. However, such analysis and quantitation takes considerable space and will not be shown here. Instead, the parsings for Simulation I shown in Figures 7 and 8 will be discussed informally.

8.4. Evaluation of EXIN Network Performance on Exclusive Allocation Criterion

Given the three exclusive allocation conditions described above, the performance of various networks can be evaluated. In particular, the performance of the EXIN network of Simulation I, shown in Figures 7 and 8, will be examined below. A sample of the most illustrative parsings, and the degree to which they satisfy the exclusive allocation conditions, will be discussed. The following paragraphs refer to Figures 7 and 8. The most complex example below, pattern *abcdf*, is examined in extra detail, introducing and using several concepts to evaluate the parsing.

Pattern *a*. The active Layer 2 neuron has the label *a*. It is fully active, so it fully accounts for the input pattern *a*. No other Layer 2 neuron is active, so the activations across Layer 2 are fully accounted for by the input pattern. Thus, Conditions 1 and 2 are satisfied. There is only one Layer 2 neuron whose label matches the input pattern, so the input does not have to be divided among more than one Layer 2 neuron. Thus, Condition 3 is satisfied.

Pattern *b*. Pattern *b* is part of familiar patterns *ab* and *abc*. The network activates neuron *ab* at about the 50% level. Because *b* constitutes 50% of the pattern *ab*, the activation of neuron *ab* fully accounts for the input pattern. Likewise, the activation of ab is fully accounted for by the input pattern. Pattern b is more similar to ab than to abc, so it is correct for neuron abc to be inactive in this case.

Pattern c. This case is similar to the previous example. Pattern c is part of abc and cd. However, neuron abc is slightly active, and neuron cd is active at a level slightly less than 50%. Conditions 1 and 2 are satisfied: the *sum* of the Layer 2 activations attributable to c is still the same as the activations attributable to b in the previous example, and the activation of neurons abc and cd are attributable to disjoint fractions (approximately 25% and 75%) of the activation of c. Condition 3 is not as well satisfied here as in the previous example. The difference can be explained by the weaker inhibition between abc and cd than between ab and abc; more coactivation is thus allowed.

To satisfy Condition 3, a network must determine which output neurons represent the best matches for an input pattern. Some degree of tolerance is necessary to let multiple neurons qualify as "best" matches even if they are not exactly equally good. Thus, Condition 3 can actually be computed only as a function of a parameter: the best match inexactness tolerance. The simultaneous partial activation of abc and cd is a manifestation of some inexactness tolerance in the network of Simulation I.

Pattern d. Pattern d is part of cd, de, and def, and it matches cd and de most closely. The corresponding two Layer 2 neurons are active, both between the 25% and 50% levels. Conditions 1 and 2 appear to be approximately satisfied: the activity of d is accounted for by a split activation across cd and de, and the activity of cd and de are accounted for by disjoint fractions of the activity of d. Condition 3 is also approximately (but not perfectly) satisfied, because the two neuron activations are nearly equal.

Pattern ac. This example is comparable to the example of pattern c. The input a is fully accounted for by the activation of Layer 2 neuron a, and vice versa. But the activation of Layer 2 neuron a strongly inhibits the activation of abc. Thus, the input c becomes fully accounted for by the 50% activation of cd, and vice versa. All three conditions are met here.

An alternative correct parsing would be to partially activate neuron abc and to inhibit neurons a and cd. Other parameter choices for the network might have yielded that parsing.

Pattern af. Pattern af can be compared to patterns a and f; the response to af is merely a superposition of the separate responses to a and f. Conditions 1, 2, and 3 are clearly met here.

Pattern ade. The response to ade is a superposition of the responses to a and de; all three conditions are met.

Pattern $abcd$. Pattern $abcd$ is parsed as $ab + cd$, which meets all three conditions.

By comparison, the winner-take-all network's behavior can also be analyzed on the basis of its adherence to the exclusive allocation criterion. As shown in Figure 5E, the WTA network's best response to pattern $abcd$ is to activate the neuron labeled abc (as determined by the network's responses to the training patterns). Here, Condition 1 is not satisfied, because the activation of input unit d is not accounted for by any activation in Layer 2. Thus, it appears that the WTA network would not satisfy the exclusive allocation criterion as well as the EXIN network does, based on this example input pattern.

Pattern $abcde$. When e is added to $abcd$, the parsing is completely altered; $abcde$ is represented as $abc+de$, which meets all three conditions.

Pattern $abcdf$. When f is added to $abcd$, a chain reaction alters the network's response, from def down to a in Layer 2. The presence of d and f causes the def neuron to become approximately

50% active. In turn, this inhibits the *cd* neuron more, which then becomes less active. As a result, the *abc* neuron receives less inhibition and becomes more active. This in turn inhibits the activity of the neuron *ab*. Because neuron *ab* is less active, neuron *a* then becomes more active.

These increases and decreases tend to balance one another, thereby keeping Conditions 1 and 2 satisfied. The dominant parsing appears to be $ab+cd+def$, but the overlap between *cd* and *def* prevents those two neurons from becoming fully coactive. As a result, the alternative parsings involving *abc* or *a* can become partially active. Because of the overlap between neurons *cd* and *def*, the input pattern is truly ambiguous; Condition 3 appears to be satisfied, by the distributed activation pattern. However, the precise degree to which Condition 3 is satisfied could be determined best in this case with a specific mathematical expression for the condition.

Some of the considerations that enter into the formulation of such a mathematical expression are illustrated in the table of Figure 14. The table describes approximate parsing coefficients for pattern *abcdf*. The coefficients shown in the table were estimated manually. These coefficients represent the portion of the credit for each Layer 1 neuron activation that can be attributed to each Layer 2 neuron activation. For example, the activation of Layer 1 neuron *a* is 1; 21% of this "energy" is allocated to Layer 2 neuron *a*, and 79% is allocated to *ab*. The input to neuron *ab*, $0.79 + 0.38$, is divided by the neuron's normalization factor (size), 2. This size factor is derived from the neuron's label, which is determined by the training (familiar) patterns to which the neuron responds. The resulting attributed activation value, 0.59, is very close to the actual activation, 0.58, of neuron *ab* in Simulation I. The existence of parsing coefficients (e.g., the ones shown in Figure 14) that produce attributed activations that are all close to the actual activations shows that Conditions 1 and 2 are well satisfied for input pattern *abcdf*.

To Layer 2 neuron	From Layer 1 neuron: a	b	c	d	e	f	Total parse energy	Neuron size	Attributed activation	Actual activation
a	.21						.21	/ 1 =	.21	.19
ab	.79	.38					1.17	/ 2 =	.59	.58
abc	.00	.62	.12				.74	/ 3 =	.25	.24
cd			.88	.30			1.18	/ 2 =	.59	.58
de				.00	.00		.00	/ 2 =	.00	.00
def				.70	.00	1.00	1.70	/ 3 =	.57	.56
	1	1	1	1	0	1				

Figure 14: Parsing coefficients and attributed activations. The table inside the rectangle shows a way in which the "energy" from the input pattern *abcdf* can be divided among the Layer 2 neurons to produce (approximately) the activations of the Layer 2 neurons shown in Figure 8. Because the results of this computation are very close to the EXIN network's simulated results (compare columns in the gray shaded area), it can be concluded that the EXIN network satisfies very well the Conditions 1 and 2 for exclusive allocation, on pattern *abcdf*.

Pattern *abcdef*. This pattern is parsed as $abc+def$, which meets all three conditions.

Other Patterns. The patterns listed above were selected for discussion on the basis of their interesting properties. The network's response to all the other patterns can also be evaluated using the exclusive allocation criterion. In each case, the EXIN network adheres well to the three exclusive allocation conditions. EXIN networks and other networks can be evaluated on other criteria, as well as on the exclusive allocation criterion. A complete analysis of the exclusive allocation behavior of any network should include a formal specification of the three conditions, plus a computation of the extent to which the network adheres to the conditions, across all possible input patterns. An example of one such computation was shown above for pattern *abcdf*. Thus, further testing is needed here before quantitative comparisons between networks can be made on the basis of exclusive allocation. Nevertheless, the discussion above illustrates qualitatively the high degree with which EXIN networks show exclusive allocation behavior.

8.5. Comparison With Decorrelators

In EXIN networks, both the excitatory and inhibitory learning rules are bounded: the functions $h(x_j)$ and $q(x_i)$ in equations (6) and (7) are rectified, so that connection weights cannot "change sign" (excitatory weights cannot become inhibitory, and vice versa). The boundedness produces key differences (described below) between the behavior of the networks in this work and that in others' work. In particular, the context-sensitivity property emerges as a consequence of the boundedness.

Unlike neural networks described by Kohonen (novelty detector, 1984), Földiák (linear decorrelator, 1989), and Rubner and Schulten (principal components analyzer, 1990), EXIN networks *do not allow connections to be converted from excitatory to inhibitory or vice-versa.* Besides being more biologically plausible, this restriction provides the key advantage of preventing some of the lateral inhibitory connection weights from vanishing. Because some of the lateral inhibitory connection weights remain strong, they enforce contextual constraints on allowable combinations of Layer 2 neuron activations.

Figures 5C,F,I show a linear decorrelator network, in which lateral connections have vanished and some connections from Layer 1 to Layer 2 have become inhibitory. The linear decorrelator network essentially responds to the differences, or *distinctive features* (Anderson, Silverstein, Ritz, & Jones, 1977; Sattath & Tversky, 1987), among the patterns, rather than to the patterns themselves. For instance, the neuron labeled *abc* actually becomes wired to respond optimally to pattern *c*-and-not-*d*. The neuron labeled *cd* becomes wired to respond optimally to pattern *d*. As a consequence, the linear decorrelator network does not activate the closest match to some unfamiliar patterns, such as pattern *c* (Figure 5I). On the other hand, EXIN networks become wired to represent the *common features* (Sattath & Tversky, 1987) among the input patterns, as illustrated in Figure 5D,G,J.

Besides the linear decorrelator, Földiák (1990, 1992) has more recently described a nonlinear decorrelator that incorporates an additive adaptive threshold. This method combines a Hebbian-variant learning rule for the feedforward connections from Layer 1 to Layer 2 with an anti-Hebbian learning rule for the lateral connections within Layer 2. Each neuron has an additive adaptive threshold T_i, which is modified dynamically to keep the neuron's frequency of activation close to a prespecified, fixed value. A neuron's total input signal value must exceed its threshold before it can become active. The threshold is modified on each simulation step according to the equation $\Delta T_i = \Gamma\,(x_i - P)$, where P is a fixed probability of firing and Γ is a small rate constant. Földiák's additive threshold is analogous to the multiplicative adaptive sensitivity (size) rule of EXIN networks (equation 1) in that it also allows different neurons to respond to different size patterns. However, one difference between additive and multiplicative rules is their response to weak ("low-contrast") patterns. An additive threshold prevents neurons from becoming active in response to weak patterns, whereas a multiplicative sensitivity allows neurons to respond. As shown in Figure 15, an EXIN network maintains its selectivity even when its input patterns are made extremely faint.

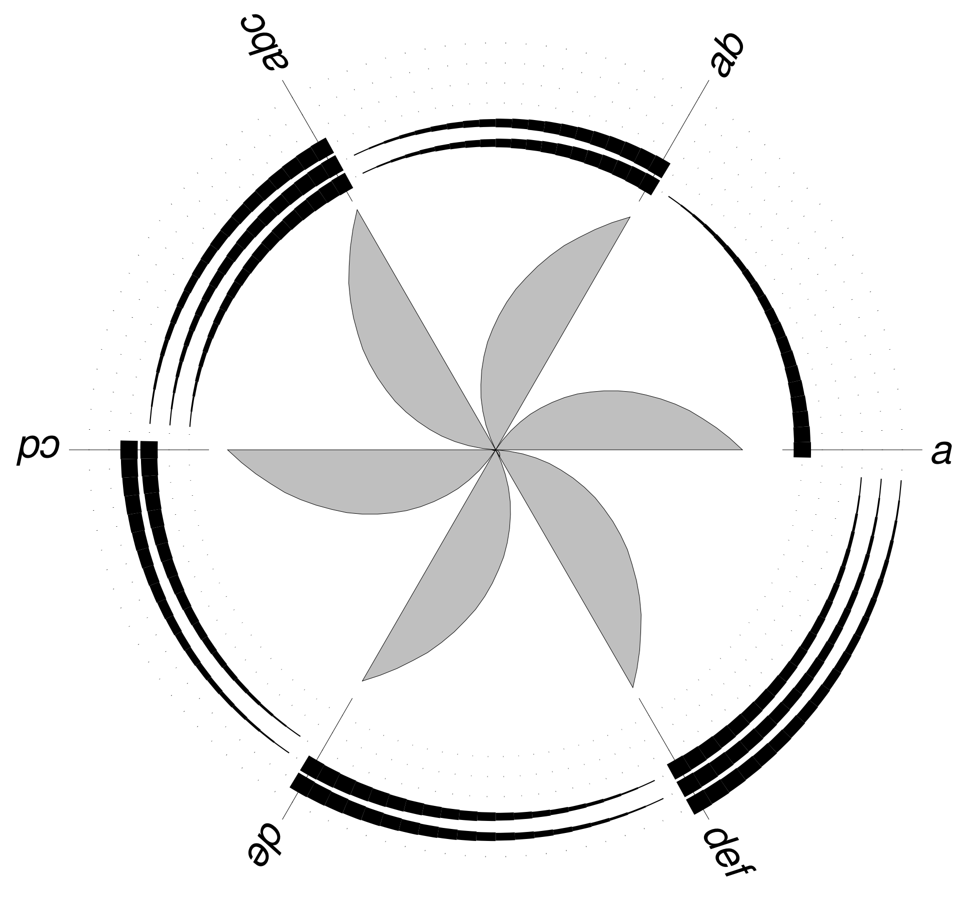

Figure 15: Response of EXIN network to patterns of different intensities. (See Figure 9 for description of plot format.) Each of the six training patterns is presented at varying intensities to the network of Simulation I. Even at low intensities, the neurons respond selectively and uniquely. As intensities increase, the neurons' responses increase monotonically.

The linear decorrelator network can also be analyzed on the basis of its adherence to the exclusive allocation criterion. The relevant case here is the network's behavior in response to pattern c, as shown in Figure 5I. As discussed before, each of the Layer 2 neurons is given a *label*, based on the network's responses to patterns in the training set. When c is presented to the linear decorrelator network, the only neuron that becomes fully active is neuron abc. Conditions 1 and 2 are not well satisfied because the maximum activation of neuron abc attributable to input c is $\frac{1}{3}$, yet neuron abc is fully active. Condition 3 is also not well satisfied because neuron cd represents a closer match than neuron abc to the input pattern c. Thus, it appears, at least qualitatively, that the linear decorrelator network does not adhere to the exclusive allocation criterion as well as the EXIN network does. Further quantitative analysis would be needed to make the full comparison between networks.

8.6. Comparison With Masking Fields: Purely Developmental Rules Do Not Suffice

The pattern selectivity that EXIN networks learn is similar to the fixed pattern selectivity of *masking fields* (Cohen & Grossberg, 1986, 1987; Grossberg, 1978, 1986). However, masking fields require extensive specific hardwiring (instead of learned wiring), using the static, predetermined spatial overlap of pattern codes to determine inhibitory weights. Masking fields are prewired to represent *all* possible binary input patterns up to a certain size (s) and consequently require an exponentially large number of neurons (at least $2^s - 1$). Much of the representational capacity of those neurons may never be used in some perceptual environments. EXIN networks, on the other hand, wire themselves adaptively in response to environmental demands. They tend to use all available representational capacity and are in that sense more biologically plausible and efficient than masking fields.

In the masking field simulations of Cohen and Grossberg (1986, 1987), inhibitory weights and neuron sizes are carefully chosen at the outset and remain fixed. Excitatory weights are modifiable only to a small degree by an "adaptive sharpening" process. Thus, their masking field networks are essentially prewired to respond selectively to all possible binary input patterns (up to a certain maximum scale) and are designed to retain most of their predetermined structure during exposure to input patterns. The prewiring of sensitivity to all patterns is not appropriate for many applications, for instance, when the patterns to be recognized are large (exponentially many Layer 2 neurons would be needed), or when the statistical distribution of input patterns is nonuniform (the allocation of neurons to patterns would be inefficient).

Nonetheless, the way in which Cohen and Grossberg specified inhibitory weights provided key insights that led to the design of EXIN networks. Inhibition between Layer 2 neurons in their masking field networks depends on several factors, including the size of the intersection, or spatial *overlap*, between the patterns coded by the neurons. Thus, neurons coding patterns abc and bcd mutually project much stronger reciprocal inhibitory connections than neurons coding abc and def. One intended feature of this scheme is that each Layer 2 neuron competes against other Layer 2 neurons only to the extent to which they share some of their input from Layer 1. On the other hand, EXIN networks rely on *temporal*, rather than spatial, overlap between pattern codes to control the development of inhibitory weights.

Cohen and Grossberg (1986) argue that the computation of the spatial overlap function emerges as a consequence of random and distance-dependent axonal outgrowth processes. They claim that their method of computing inhibitory connection weights is justified by the following local spatial distance principle:

> List nodes [Layer 2 neurons] *become* list nodes due to random growth of connections from item nodes [Layer 1 neurons]. Two list nodes therefore tend to be closer ... if they receive more input pathways from the same item nodes.... If a pair of list nodes ... is closer, then their axons can more easily contact each other, other things being equal (p. 19).

(In this principle, the word "closer" implies that there exists a spatial topology – most likely 1-D, 2-D, or 3-D – across the network, whereby the physical distance between any two neurons can be gauged.) This principle refers only to *spatial* distance between neurons as a function of the activity-dependent growth of individual neurons – not as a function of the *correlation* of the activity of pairs of neurons. The principle implies that each Layer 2 neuron tends to be spatially located near the Layer 2 position corresponding to the *centroid* of the spatial locations of its Layer 1 input neurons.

Random and distance-dependent processes may account for an overall tendency for neurons coding overlapping patterns to project stronger reciprocal inhibitory connections than neurons coding nonoverlapping patterns. However, as written in their original papers (Cohen & Grossberg, 1986, 1987), the complete computation of the overlap function is physically unrealizable unless activity correlation factors are also used. Suppose their network's Layer 1 neurons were spatially ordered: (a, b, c, d). Their local spatial distance principle therefore predicts that neurons coding patterns ad and bc would project *stronger* reciprocal inhibitory connections than neurons coding ab and cd (Figure 16A), even though the overlap, or intersection, of both pairs is the same (i.e., null). The reason is that the centroids of ad and bc spatially coincide, whereas the centroids of ab and cd are spatially separated. Thus, the Layer 2 neurons coding ad and bc should project stronger reciprocal inhibitory connections than ab and cd, according to their principle. But the *implementation* described by Cohen and Grossberg fails to follow their own local spatial distance principle; their implementation uses explicit nonlocal intersection information to produce the *same* inhibitory weights between ab and cd as between ad and bc. Their analysis based on local spatial distance between neurons is inconsistent with their implementation of an external, nonlocal process to specify inhibitory weights for both of the cases $ab \leftrightarrow cd$ and $ad \leftrightarrow bc$.

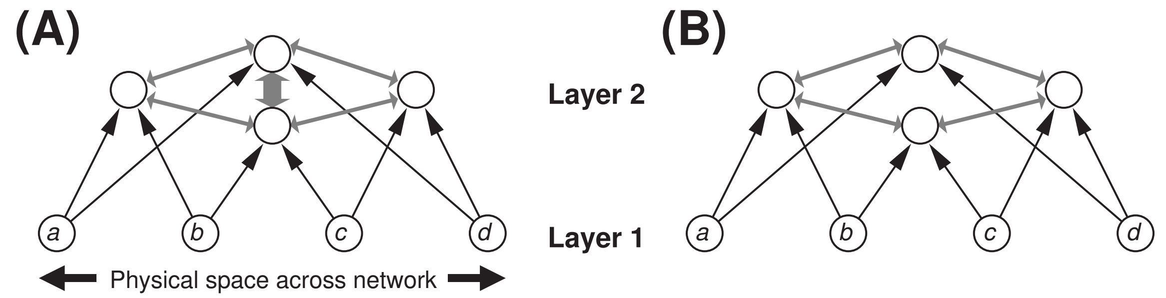

Figure 16: Development of inhibitory connections. (A) The developmental analysis for masking fields by Cohen and Grossberg (1986) predicts that the Layer 2 neurons coding patterns ad and bc should project very strong reciprocal inhibitory connections (thick shaded arrow) because their centroids are likely to be at the same position across the 1-D network topography. The Layer 2 neurons coding patterns ab and cd would project very weak reciprocal inhibitory connections (not drawn), because their centroids would most likely be relatively distant. Other pairs of Layer 2 neurons would project moderately strong reciprocal inhibitory connections (thin shaded arrows) because their centroids are likely to be located at intermediate distances. (B) Contrary to their analysis, the computational *implementation* of masking fields by Cohen and Grossberg (1986, 1987) would use very *weak* reciprocal inhibitory connections between the Layer 2 neurons coding patterns ad and bc. This weak inhibition cannot be produced using the random and distance-dependent outgrowth rules proposed by Cohen and Grossberg (1986), but it *can* be produced using a learning rule that uses pairwise activity correlations, such as the EXIN inhibitory learning rule.

Indeed, there is no single site in the network at which, for example, the information that one Layer 2 neuron codes abc and another codes bcd can be locally available within a single physical site, given only random, distance-dependent, or single-activity-dependent processes. For this reason, the inhibitory connection weights in their implementation cannot be said to fully self-organize. Although the spatial intersection implementation of Cohen and Grossberg is not local, it is valuable because it characterizes the desired end result: inhibitory weight as a function of pattern overlap.

EXIN networks circumvent the spatial distance principle by using local *temporal* pairwise activity correlation information available during exposure to the perceptual environment (Figure 16B). Because the EXIN inhibitory learning rule uses pairwise activity correlation information, EXIN networks can dynamically determine that the Layer 2 neurons coding ab and bc

should project strong reciprocal inhibitory connections (because those neurons are frequently coexcited), whereas those coding *ad* and *bc* should not (because they are less frequently coexcited). The failure of the spatial distance principle of Cohen and Grossberg suggests that the developmental specification of neural wiring in biological systems must be performed either by a nonlocal, detailed, genetically specified process or by a local, *pairwise* activity-dependent, environmentally guided process.

8.7. Stability of EXIN Networks

The network's connection weights constitute a coding of the structure of the external world. Like other competitive learning algorithms, the EXIN learning rules do not produce absolutely stable codings. The formation and persistence of the codings depend heavily on the statistics of the network's input history. If the probabilities of events are altered over a long enough period of time, then the connection patterns change adaptively to fit the new statistics of the environment. For example, if a specific region of the input pattern space starts to occur more frequently in the environment than usual, then the network responds by devoting more neurons to representing those patterns and fewer to other patterns.

The plasticity is an advantage for lower-level, perceptual functions because perceptual environments change rather slowly in general. For example, it would compensate for the systematic distortions produced by growth of the eyeball as a newborn animal ages.

However, the plasticity of competitive learning algorithms makes them unsuitable for modeling some aspects of brain function, such as higher-level, cognitive memory. For example, it would be disadvantageous for an actor's memory of Hamlet's "To be or not to be" soliloquy to become distorted by events occurring between performances. If an alteration of input statistics is only temporary or spurious, then the changes it induces might erode desirable connectivity patterns.

The networks in this paper control the degree of plasticity and stability of connection weights by insuring that presentation of a single input pattern can change connection weights just a small amount. Only the cumulative and systematic effects of many input presentations can significantly recode the network's connectivity. If the rate of weight change is made small enough, then one can be reasonably sure that the resultant connection patterns stably code the statistics of the input history rather than the adventitious correlations in a small number of input presentations. However, if the rate of weight change is made too small, then a very large number of input presentations would be needed to produce the desired adaptation effects.

More complex approaches to the tradeoffs between stability and plasticity (involving attentional mechanisms and weight "freezing") are explored by Carpenter and Grossberg (1987ab, 1990, 1992), Carpenter, Grossberg, and Rosen (1991ab), Grossberg (1980, 1982a), and Nigrin (1990abc, 1993). For the purposes of this paper, the simple expedient of fixing the rate of connection weight change was deemed sufficient.

The approach taken in Simulations I–IV was to choose a rate of weight change small enough that even several spuriously correlated input presentations would not change the connection topology – but no smaller. The result was that the connection pattern was quite stable once it settled into its final form; the network would reach its final structure after a computationally reasonable amount of exposure to its perceptual environment.

A different stability problem arises in EXIN networks when the number of Layer 2 neurons exceeds the number of patterns in the environment. Suppose a network contains three Layer 2 neurons and is presented with only two input patterns, say *ab* and *cd*. One Layer 2 neuron will respond to *ab*, another to *cd*, and the third to neither input pattern. But then the inhibitory learning rule will cause the third Layer 2 neuron to become progressively less inhibited, until it responds to either *ab* or *cd*. But then one of the other Layer 2 neurons will respond to neither input pattern, so it will become less inhibited until it again responds. Thus, a given input pattern is not stably coded by a particular Layer 2 neuron.

For this reason, EXIN networks are appropriate only where the number of Layer 2 neurons is less than or equal to the number of patterns in the environment. Perceptual environments generally contain an *infinitely* graded set of patterns, so EXIN networks work well as models of perceptual function.

8.8. Neurophysiological Evidence for Inhibitory Learning

There have been many neurophysiological studies of plasticity or learning in excitatory connections but fewer in inhibitory connections. Hendry, Fuchs, deBlas, and Jones (1990) found that the density of receptors for the inhibitory neurotransmitter GABA in adult monkey visual cortex declined in an activity-dependent manner with monocular deprivation. This effect is analogous to the behavior of EXIN networks, in that the networks compensate for lower overall activity across Layer 2 by producing lower inhibitory weights, which allows more activity.

Whitsel, Favorov, Kelly, and Tommerdahl (1990) found "input-specific, time-dependent pericolumnar lateral inhibitory interactions in somatosensory cortex" (p. 7). The "pericolumnar 'lateral interactions' are subject to dynamic regulation by repetitive environmental stimuli" (p. 10), where repetitive activity produces "greater-than-normal and longer-lasting inhibition.... The final outcome is enhanced local contrast in the cortical columnar activation pattern" (p. 7). Even though the effect that they found lasts only for a few seconds, the sign of the changes in inhibition is the same as that in EXIN networks, where local contrast enhancement is generated by increasing the inhibition between neurons coding similar or frequently coactivated patterns. They did not investigate longer-term changes in inhibition, which might arise by the "piling up" of persistent repetitive stimulation.

The network structures produced by the EXIN learning rules are consistent with neurophysiological measurements showing greater inhibition between neurons coding similar patterns than between neurons coding dissimilar patterns. Such inhibitory interactions have been found in many perceptual areas, including visual orientation detection, where reciprocal inhibition declines and detection thresholds drop with orientation dissimilarity (Blakemore, Carpenter, & Georgeson, 1970; Cannon & Fullencamp, 1990; J.I. Nelson, 1985; S.B. Nelson, 1991), and auditory pitch detection, where inhibition declines with preferred frequency dissimilarity (Voigt & Young, 1990).

8.9. EXIN Network Models in Perception

In vision, EXIN networks perform a context-sensitive segmentation, independently representing multiple objects or features at once. Marshall (1989ab, 1990abef, 1991, 1992b) has used EXIN networks to model several aspects of visual perception, including the development of disambiguation mechanisms for pattern-motion (Movshon, Adelson, Gizzi, & Newsome, 1985) in the aperture problem (Adelson & Movshon, 1982; Hildreth, 1983; Marr, 1982; Marr & Ullman, 1981), the development of end-stopping and length sensitivity (Hubel & Wiesel, 1977; Kato, Bishop, & Orban, 1978; Orban, Kato, & Bishop, 1979), and the development of orientation sensitivity (Hubel & Wiesel, 1962) in neurons capable of representing edge intersections (Walters, 1987).

Preliminary results have also been successful in using the EXIN learning rules to model the development of neural representation of binocular disparity surfaces (Marshall, 1990e, 1992b), including stereo transparency segmentation (Prazdny, 1985; Weinshall, 1989; Lehky & Sejnowski, 1990); in this case, the selectively reduced inhibition lets the network simultaneously represent multiple transparently overlaid surfaces in depth within the same 2-D image region. The reduced inhibition implements a form of perceptual *scission*, or dissociation, between the representations of the overlaid surfaces, so that multiple surface representations can be active simultaneously, without strong mutual interference. Other 2-D models, which use WTA dynamics (Becker & Hinton, 1992; Hinton & Becker, 1990; Marr & Poggio, 1976), either are limited to representing a single object or surface at a single depth value for any given retinotopic location or are forced to alternate via an attentional mechanism their representations of multiple objects (Fukushima, 1986).

The segmentation or parsing capabilities of EXIN networks may be useful in modeling audition and speech perception as well. For example, at a cocktail party (Cherry, 1953), human auditory systems extract and maintain simultaneous, relatively independent representations of multiple sound groupings (e.g., multiple speakers, music, clattering dinnerware), even though they are all superimposed on a single shared input channel (air vibrations). Even though one can selectively attend to a single speaker (even monaurally), if one's name is mentioned in another conversation, one notices it and can immediately switch attention. At some level, therefore, multiple sound groupings are extracted and processed in parallel. The EXIN learning rules can potentially be used in modeling the development of mechanisms capable of simultaneously extracting *multiple* input streams from a single channel.

9. Conclusions: Discovering Structure in Complex Perceptual Environments

Local excitatory learning rules (variants of Hebbian rules) have long been used in understanding the self-organization of featural selectivity in perceptual systems. The EXIN method of combining inhibitory with excitatory learning and with neuron "growth" provides further advantages:

- Multiple neurons can be activated simultaneously, under regulated circumstances, permitting uncertainty and multiplicity to be represented. This property implements a form of distributed coding, which avoids or at least alleviates the "grandmother cell" dilemma of combinatorial explosion.
- The inhibitory weights do not need to be prewired; they develop as required according to the inhibitory learning rule.
- Approximate symmetry of inhibitory weights is maintained dynamically, by the asymmetric learning rules. It is not necessary to impose artificial weight symmetry conditions.
- The network need not be explicitly configured in advance to represent all possible patterns. It develops to represent efficiently the set of input patterns according to their distribution in its input environment.
- The stability of learning is enhanced, because the multiplexing property allows a given number of neurons in an EXIN network to represent more patterns than the same number of neurons in a WTA network.
- Pattern codes need not be limited to integer sizes; fractional neuron sizes can develop, thereby representing analog patterns accurately.
- All connection weights, both excitatory and inhibitory, are computed strictly locally.
- EXIN networks exhibit global context-sensitive constraint satisfaction behavior. EXIN networks

thus allow small contextual changes in an input pattern to alter dramatically the optimal parsing of the pattern.

These properties of EXIN networks let them absorb and organize data in complex environments containing multiple superimposed patterns, ambiguous patterns, overlapping patterns at different scales, and contextually constrained patterns. It is interesting that subtle variations in the network's learning rules (e.g., allow vs. disallow connection weights to change sign) produce major differences in the neural mechanism that develops (e.g., neurons sensitive to common features vs. distinctive features). A likely fruitful research area is the study of additional SONN rule variations capable of yielding more sophisticated network development and network behavior.

References

Adelson, E.H. & Movshon, J.A. (1982). Phenomenal coherence of moving visual patterns. *Nature*, **300**, 523–525.

Amari, S. (1977). Neural theory of association and concept formation. *Biological Cybernetics*, **26**, 175–185.

Amari, S. & Takeuchi, A. (1978). Mathematical theory on formation of category detecting nerve cells. *Biological Cybernetics*, **29**, 127–136.

Anderson, J.A., Silverstein, J.W., Ritz, S.A., & Jones, R.S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, **84** (5), 413–451.

Barlow, H.B. (1980). The absolute efficiency of perceptual decisions. *Philosophical Transactions of the Royal Society of London, Ser. B*, **290**, 71–82.

Barto, A.G., Sutton, R.S., & Anderson, C.W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, **13**, 834–846.

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

Bienenstock, E.L., Cooper, L.N., & Munro, P.W. (1982). Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, **2**, 1, 32–48.

Blakemore, C., Carpenter, R.H.S., & Georgeson, M.A. (1970). Lateral inhibition between orientation detectors in the human visual system. *Nature*, 228, 37–39.

Bregman, A.S. (1990). *Auditory scene analysis: The perceptual organization of sound.* Cambridge, MA: MIT Press.

Cannon, M.W. & Fullencamp, S.C. (1990). Inhibitory interactions in suprathreshold vision. *Investigative Ophthalmology and Visual Science*, **31**, 4, 323.

Carlson, A. (1990). Anti-Hebbian learning in a non-linear neural network. *Biological Cybernetics*, **64**, 171–176.

Carpenter, G.A. & Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, **37**, 54–115.

Carpenter, G.A. & Grossberg, S. (1987b). ART 2: Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optics*, **26**, 4919-4930.

Carpenter, G.A. & Grossberg, S. (1990). ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks*, **3**, 129–152.

Carpenter, G.A. & Grossberg, S. (1992). Fuzzy ARTMAP: Supervised learning, recognition, and prediction by a self-organizing neural network. *IEEE Communications Magazine*, **30**, 38–49.

Carpenter, G.A., Grossberg, S., & Rosen, D.B. (1991a). ART2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, 4, 493–504.

Carpenter, G.A., Grossberg, S., & Rosen, D.B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4, 759–771.

Cherry, E.C. (1953). Some experiments on the recognition of speech with one and with two ears. *Journal of the Acoustical Society of America*, **25**, 975–979.

Cohen, M.A. & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-13**, 815–826.

Cohen, M.A. & Grossberg, S. (1986). Neural dynamics of speech and language coding: Developmental programs, perceptual grouping, and competition for short term memory. *Human Neurobiology*, **5**, 1–22.

Cohen, M.A. & Grossberg, S. (1987). Masking fields: A massively parallel neural architecture for learning, recognizing, and predicting multiple groupings of patterned data. *Applied Optics*, **26**, 1866–1891.

Coolen, A.C.C. & Kuijk, F.W. (1989). A learning mechanism for invariant pattern recognition in neural networks. *Neural Networks*, **2**, 495–506.

DeSieno, D. (1988). Adding a conscience to competitive learning. *IEEE international conference on neural networks*, **I.**, 117–124.

Duda, R.O. & Hart, P.E. (1972). Use of the Hough Transform to detect lines and curves in pictures. *Communications of the ACM*, **15**, 11–15.

Easton, P. & Gordon, P.E. (1984). Stabilization of Hebbian neural nets by inhibitory learning. *Biological Cybernetics*, **51**, 1–9.

Field, D.J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, **4**, 12, 2379–2394.

Földiák, P. (1989). Adaptive network for optimal linear feature extraction. *Proceedings of the International Joint Conference on Neural Networks,* Washington, DC, June 1989, **I.**, 401–405.

Földiák, P. (1990). Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics,* **64**, 2, 165–170.

Földiák, P. (1991). Learning invariance from transformation sequences. *Neural Computation,* **3**, 2, 194–200.

Földiák, P. (1992). Models of sensory coding. Technical Report CUED/F-INFENG/TR 91, Department of Engineering, University of Cambridge.

Fukushima, K. (1986). A neural network model for selective attention in visual pattern recognition. *Biological Cybernetics*, **55**, 5–15.

Grossberg, S. (1972). Neural expectation: Cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetik*, **10**, 49–57.

Grossberg, S. (1976a). On the development of feature detectors in the visual cortex with applications to learning and reaction-diffusion systems. *Biological Cybernetics*, **21**, 145–159.

Grossberg, S. (1976b). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, **23**, 121–134.

Grossberg, S. (1976c). Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, **23**, 187–202.

Grossberg, S. (1978). A theory of human memory: Self-organization and performance of sensory-motor codes, maps, and plans. In *Progress in Theoretical Biology*, **5**. San Diego: Academic Press, 233–374.

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, 87, 1–51.

Grossberg, S. (1982a). Processing of expected and unexpected events during conditioning and attention: A psychophysiological theory. *Psychological Review*, 89, 529–572.

Grossberg, S. (1982b). *Studies of mind and brain.* Boston: Reidel Press.

Grossberg, S. (1986). The adaptive self-organization of serial order in behavior: Speech, language, and motor control. In E.C. Schwab & H.C. Nusbaum (Eds.), *Pattern recognition by humans and machines, vol. 1: Speech recognition.* San Diego: Academic Press.

Grossberg, S. & Marshall, J.A. (1989). Stereo boundary fusion by cortical complex cells: A system of maps, filters, and feedback networks for multiplexing distributed data. *Neural Networks*, **2**, 29–51

Hebb, D.O. (1949). *The organization of behavior.* New York: Wiley.

Hendry, S.H.C., Fuchs, J., deBlas, A.J., & Jones, E.G. (1990). Distribution and plasticity of immunocytochemically localized $GABA_A$ receptors in adult monkey visual cortex. *Journal of Neuroscience*, **10**, 7, 2438–2450.

Hildreth, E.C. (1983). Computing the velocity field along contours. *Proceedings of the ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion*, Toronto: 26–32.

Hinton, G.E. & Becker, S. (1990). An unsupervised learning procedure that discovers surfaces in random-dot stereograms. *Proceedings of the International Joint Conference on Neural Networks*, Washington DC, January 1990, **I.**, 218–222.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, **79**, 2554–2558.

Hough, P.V.C. (1962). Method and means for recognizing complex patterns. *U.S. Patent* 3,069,654, U.S. Patent Office.

Hubel, D.H. & Wiesel, T.N. (1962). Receptive fields, binocular interactions, and functional architecture in cat's visual cortex. *Journal of Physiology*, **160**, 106–154.

Hubel, D.H. & Wiesel, T.N. (1977). Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London, Ser. B*, **198**, 1–59.

Kato, H., Bishop, P.O., & Orban, G.A. (1978). Hypercomplex and simple/complex cell classifications in cat striate cortex. *Journal of Neurophysiology*, **41**, 1071–1095.

Kersten, D. (1987). Predictability and redundancy of natural images. *Journal of the Optical Society of America A*, **4**, 12, 2395–2400.

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

Kohonen, T. (1984). *Self-organization and associative memory.* New York: Springer-Verlag.

Lehky, S.R. & Sejnowski, T.J. (1990). Neural model of stereoacuity and depth interpolation based on a distributed representation of stereo disparity. *Journal of Neuroscience*, **10**, 7, 2281–2299.

Linsker, R. (1986a). From basic network principles to neural architecture: Emergence of spatial-opponent cells. *Proceedings of the National Academy of Sciences of the U.S.A.*, **83**, 7508–7512.

Linsker, R. (1986b). From basic network principles to neural architecture: Emergence of orientation-selective cells. *Proceedings of the National Academy of Sciences of the U.S.A.*, **83**, 8390–8394.

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

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: W.H. Freeman and Company.

Marr, D. & Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, **194**, 283–287.

Marr, D. & Ullman, S. (1981). Directional selectivity and its use in early visual processing. *Proceedings of the Royal Society of London, Ser. B*, **211**, 151–180.

Marshall, J.A. (1989a). *Neural networks for computational vision: Motion segmentation and stereo fusion.* Ph.D. Dissertation, Boston University Computer Science Department. Ann Arbor, Michigan: University Microfilms Inc.

Marshall, J.A. (1989b). Self-organizing neural network architectures for computing visual depth from motion parallax. *Proceedings of the International Joint Conference on Neural Networks*, Washington DC, June 1989, **II.**, 227–234.

Marshall, J.A. (1990a). Self-organizing neural networks for perception of visual motion. *Neural Networks*, **3**, 45–74.

Marshall, J.A. (1990b). Development of length-selectivity in hypercomplex-type cells. *Investigative Ophthalmology and Visual Science*, **31**, 4, 397.

Marshall, J.A. (1990c). A self-organizing scale-sensitive neural network. *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, **III.**, 649–654.

Marshall, J.A. (1990d). Representation of uncertainty in self-organizing neural networks. *Proceedings of the International Neural Network Conference*, Paris, France, July 1990, 809–812.

Marshall, J.A. (1990e). Self-organizing neural network for computing stereo disparity and transparency. *Optical Society of America Annual Meeting Technical Digest*, Boston, November 1990, 268.

Marshall, J.A. (1990f). Adaptive neural networks for multiplexing oriented edges. In D.P. Casasent (Ed.), *Intelligent robots and computer vision IX: Neural, biological, and 3-D methods*, Proceedings of the SPIE **1382**, Boston, November 1990, pp.282–291.

Marshall, J.A. (1991). Challenges of vision theory: Self-organization of neural mechanisms for stable steering of object-grouping data in visual motion perception. *Stochastic and neural methods in signal processing, image processing, and computer vision*, S.-S. Chen, Ed., Proceedings of the SPIE **1569**, San Diego, pp. 200–215,

Marshall, J.A. (1992a). Development of perceptual context-sensitivity in unsupervised neural networks: Parsing, grouping, and segmentation. *Proceedings of the International Joint Conference on Neural Networks*, Baltimore, MD, III., pp. 315–320.

Marshall, J.A. (1992b). Unsupervised learning of contextual constraints in neural networks for simultaneous visual processing of multiple objects. In S.-S. Chen (Ed.), *Neural and stochastic methods in image and signal processing*, Proceedings of the SPIE **1766**, San Diego, pp. 84–93.

Martin, K.E. & Marshall, J.A. (1993). Unsmearing visual motion: Development of long-range horizontal intrinsic connections. In S.J. Hanson, J.D. Cowan, & C.L. Giles (Eds.), *Advances in neural information processing systems*, 5, San Mateo, CA: Morgan Kaufmann Publishers, pp. 417–424.

Movshon, J.A., Adelson, E.H., Gizzi, M.S., & Newsome, W.T. (1985). The analysis of moving visual patterns. In C. Chagas, R. Gattass, & C. Gross (Eds.), *Pattern recognition mechanisms.* Vatican City: Pontifical Academy of Sciences, 117–151.

Nelson, J.I. (1985). The cellular basis of perception. In D. Rose & V.G. Dobson (Eds.), *Models of the visual cortex.* Chichester: Wiley, 108–122.

Nelson, S.B. Temporal interactions in the cat visual system. I. Orientation-selective suppression in the visual cortex. *Journal of Neuroscience*, **11**, 2, 344–356.

Nigrin, A. (1990a). The real-time classification of temporal sequences with an adaptive resonance circuit. *Proceedings of the International Joint Conference on Neural Networks*, Washington, DC, January 1990, **I.**, 525—528.

Nigrin, A. (1990b). SONNET: A self-organizing neural network that classifies multiple patterns simultaneously. *Proceedings of the International Joint Conference on Neural Networks*, San Diego, June 1990, **II.**, 313–318.

Nigrin, A. (1990c). *The stable learning of temporal patterns with an adaptive resonance circuit.* Ph.D. Dissertation, Duke University Computer Science Department.

Nigrin, A. (1992). A new architecture for achieving translational invariant recognition of objects. *Proceedings of the International Joint Conference on Neural Networks*, Baltimore, June 1992, **III.**, 683–688.

Nigrin, A. (1993). *Neural networks for pattern recognition.* Cambridge, MA: MIT Press.

Orban, G.A., Kato, H., & Bishop, P.O. (1979). End-zone region in receptive fields of hypercomplex and other striate neurons in the cat. *Journal of Neurophysiology*, **42**, 3, 818–832.

Prazdny, K. (1985). Detection of binocular disparities. *Biological Cybernetics*, **52**, 93–99.

Price, D.J. & Zumbroich, T.J. (1989). Postnatal development of corticocortical efferents from area 17 in the cat's visual cortex. *Journal of Neuroscience*, **9**, 2, 600–613.

Reeke, G.N., Finkel, L.H., & Edelman, G.M. (1990). Selective recognition automata. In S.F. Zornetzer, J.L. Davis, & C. Lau (Eds.), *An introduction to neural and electronic networks*, New York: Academic Press, 203–226.

Rubner, J. & Schulten, K. (1990). Development of feature detectors by self-organization. *Biological Cybernetics*, **62**, 193–199.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by backpropagating errors. *Nature*, **323**, 533-536.

Sattath, S. & Tversky, A. (1987). On the relation between common and distinctive feature models. *Psychological Review*, **94**, 1, 16–22.

Sereno, M.E. (1986). Neural network model for the measurement of visual motion. *Journal of the Optical Society of America A*, **3**, 13, P72.

Sereno, M.E. (1987). Implementing stages of motion analysis in neural networks. *Program of the Ninth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum

Associates, 405–416.

Sereno, M.E., Kersten, D.J., & Anderson, J.A. (1988). A neural network model of an aspect of motion perception. *Science at the John von Neumann National Supercomputer Center: Annual Report FY 1988*, 173–178.

Sereno, M.I. (1989). Learning the solution to the aperture problem for pattern motion with a Hebb rule. In D. Touretzky (Ed.), *Advances in neural information processing systems*, **1**, San Mateo, CA: Morgan Kaufmann Publishers, 468–476.

Sereno, M.I. & Sereno, M.E. (1990). Learning to see rotation and dilation with a Hebb rule. In R.P. Lippmann, J.E. Moody, & D. Touretzky (Eds.), *Advances in neural information processing systems*, **3**, San Mateo, CA: Morgan Kaufmann Publishers, 320–326.

Singer, W. (1983). Neuronal activity as a shaping factor in the self-organization of neuron assemblies. In E. Basar, H. Flohr, H. Haken, & A. J. Mandell (Eds.), *Synergetics of the brain*. New York: Springer-Verlag.

Singer, W. (1985). Activity-dependent self-organization of the mammalian visual cortex. In D. Rose & V.G. Dobson (Eds.), *Models of the visual cortex*. New York: Wiley, 123–136.

Soodak, R.E. (1991). Reverse-Hebb plasticity leads to optimization and association in a simulated visual cortex. *Visual Neuroscience*, 6, 507–518.

Sun, G.Z., Chen, H.H., & Lee, Y.C. (1987). Learning stereopsis with neural networks. Preprint.

Szeliski, R.S. (1988). *Bayesian modeling of uncertainty in low-level vision*. Ph.D. Dissertation, Technical Report CMU-CS-88-169, Carnegie Mellon University Computer Science Department.

Van den Bout, D. & Miller, T.K. (1989). TInMANN: The integer Markovian artificial neural network. *Proceedings of the International Joint Conference on Neural Networks*, Washington DC, June 1989, **II.**, 205–211.

Voigt, H.F. & Young, E.D. (1990). Cross-correlation analysis of inhibitory interactions in dorsal cochlear nucleus. *Journal of Neurophysiology*, **64**, 5, 1590–1610.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, **14**, 85–100.

Walters, D. (1987). Rho-space: A neural network for the detection and representation of oriented edges. *Proceedings of the First International Conference on Neural Networks*, San Diego, June 1987.

Watson, A.B. (1987). Efficiency of a model human image code. *Journal of the Optical Society of America A*, **4**, 12, 2401–2417.

Weinshall, D. (1989). Perception of multiple transparent planes in stereo vision. *Nature*, **341**, 737–739.

Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.D. Dissertation, Harvard University.

Whitsel, B.L., Favorov, O.V., Kelly, D.G., & Tommerdahl, M. (1990). Mechanisms of dynamic peri- and intra-columnar interactions in somatosensory cortex: Stimulus-specific contrast enhancement by NMDA receptor activation. In O. Franzen & J. Westman (Eds.), *Information processing in the somatosensory system*. London: Macmillan Press.

Wilson, H.R. (1988). Development of spatiotemporal mechanisms in infant vision. *Vision Research*, **28**, 5, 611–628.

Appendix: Implementation Details

This section describes in detail the four simulations. The equations and parameters given below specify the layering of neurons in the network, the weights of excitatory and inhibitory connections between neurons, the manner in which each neuron's activity level changes according to its inputs, the manner in which connection weights vary according to neuron activity correlations, and the sequences of input to the network.

Define the notations

$$[x] \;\equiv\; \max(x,0), \qquad \lfloor x \rfloor \;\equiv\; \mathrm{floor}(x), \qquad \lceil x \rceil \;\equiv\; \mathrm{ceil}(x). \tag{A1}$$

The function floor(x) produces the greatest integer less than or equal to x, and the function ceil(x) produces the least integer greater than or equal to x.

Initial Network Structure

The initial structure of the networks in Simulations I–IV is constructed as follows. Let there be 12 neurons in the network, numbered 1–12. Let L_i represent the layer in which each neuron resides. Then

$$L_i \;=\; 1 + \Big\lfloor (i-1)/6 \Big\rfloor. \tag{A2}$$

Let $\mathcal{R}_{ji}$ be drawn pseudorandomly from the interval $[0,1)$. Then the initial excitatory connection weights are

$$z^+_{ji}(0) \;=\; Z^+_0\Big(1 + V^+(2\mathcal{R}_{ji} - 1)\Big) \quad \text{for all } i,j \text{ such that } L_j = 1 \text{ and } L_i = 2. \tag{A3}$$

The initial inhibitory connection weights are

$$z^-_{ji}(0) \;=\; Z^-_0\Big(1 + V^-(2\mathcal{R}_{ji} - 1)\Big) \quad \text{for all } i,j \text{ such that } L_j = L_i = 2 \text{ and } i \neq j. \tag{A4}$$

Input Patterns

The input to the network is generated as follows. Let P represent the number (6 or 7) of base patterns within a given Simulation. Let D_{pi} be a binary digit, 0 or 1, for $p = 1,\ldots,P$ and $i = 1,\ldots,6$. Let ϱ_p be chosen so that $0 = \varrho_0 < \varrho_1 < \varrho_2 < \ldots < \varrho_P = 1$. The ϱ_p values represent probability ranges for each pattern. Choose $\hat{\mathcal{R}}_{\lfloor t \rfloor}$ pseudo-randomly from the interval $[0,1)$. For Simulation I(a) and for the training inputs of Simulations I(b) and II, the strength of the pth pattern at time t is

$$W_p(t) \;=\; \begin{cases} 1 & \text{if } p \;=\; 1 + \Big\lfloor P\hat{\mathcal{R}}_{\lfloor t \rfloor} \Big\rfloor, \\ 0 & \text{otherwise.} \end{cases} \tag{A5}$$

For Simulations III–IV and for the test inputs of Simulations I(b) and II, the weighting of the pth pattern at time t is:

$$W_p(t) \;=\; \begin{cases} 1 - \dfrac{\hat{\mathcal{R}}_{\lfloor t \rfloor} - \varrho_{p-1}}{\varrho_p - \varrho_{p-1}} & \text{if } \varrho_p < \hat{\mathcal{R}}_{\lfloor t \rfloor} \le \varrho_{p-1}, \\[2ex] 1 - \dfrac{\varrho_{p-1} - \hat{\mathcal{R}}_{\lfloor t \rfloor}}{\varrho_{p-1} - \varrho_{p-2}} & \text{if } \varrho_{p-1} < \hat{\mathcal{R}}_{\lfloor t \rfloor} < \varrho_{p-2}, \\[2ex] 0 & \text{otherwise,} \end{cases} \tag{A6}$$

where $\varrho_k \;\equiv\; \varrho_{k+P}$ if $k < 0$.

Then during each presentation interval, the ith element of the input pattern becomes the activation value of the ith Layer 1 neuron:

$$x_i(t) \;=\; M \sum_{p=1}^{P} \Big(D_{pi} W_p(t) \Big). \tag{A7}$$

At the start of each presentation interval, Layer 2 neurons were given initial activation values of

$$x_i\Big(\lfloor t \rfloor\Big) \;=\; 0. \tag{A8}$$

Learning and Activation Rules

Activation changes for Layer 2 neurons were computed for each presentation during the interval $\lfloor t \rfloor \le t < \lfloor t+1 \rfloor$ using the equation

$$\frac{d}{dt} x_i \;=\; -A x_i \;+\; (B - x_i) \frac{\beta \sum_j \Big([x_j] z_{ji}^{+} \Big)}{\alpha + \sum_j z_{ji}^{+}} \;-\; \gamma (C + x_i) \sum_j \Big([x_j] z_{ji}^{-} \Big). \tag{A9}$$

Changes in the weights of connections projecting to (and within) Layer 2 ($L_i = 2$) were governed by the equations

$$\frac{d}{dt} z_{ji}^{+} \;=\; \epsilon\, f(x_i) \Big(-z_{ji}^{+} + h(x_j) \Big), \tag{A10}$$

$$\frac{d}{dt} z_{ji}^{-} \;=\; \delta\, g(x_j) \Big(-z_{ji}^{-} + q(x_i) \Big). \tag{A11}$$

In Simulations I–IV, the sampling and signal functions are defined as

$$f(x_i) \;=\; [x_i]^2, \qquad h(x_j) \;=\; H[x_j], \qquad g(x_j) \;=\; [x_j], \qquad q(x_i) \;=\; Q[x_i]. \tag{A12}$$

Learning was turned off ($\delta = \epsilon = 0$) during the test phase of each Simulation because the statistical pattern distribution of the test suite differed from that of the training environment.

Parameters

The following table lists the parameters common to all the Simulations:

$A = 22.5$	$B = 1$	$C = 0.1$
$\alpha = 1$	$\beta = 18.75$	$\gamma = 7500$
$\delta = 3.75$	$\epsilon = 1125$	$H = 100$
$Q = 50$	$Z_0^{+} = 1$	$Z_0^{-} = 0.25$
$V^{+} = 0.01$	$V^{-} = 0.01$	$M = 0.01$

Because the input patterns were presented at a relatively weak overall intensity ($M = 0.01$), the activation equation (A9) operated in a near-linear dynamic range. Although the numeric values of parameters $A, \beta, \gamma, \delta, \epsilon,$ and H were increased to compensate for the low value of M, they are still within a qualitatively normal range with respect to network dynamics.

In Simulations I, II, and III, $\varrho_i = i/P$ for $i = 0, \ldots, P$. In Simulation IV, $\varrho_i = 0, \frac{41}{126}, \frac{82}{126}, \frac{93}{126}, \frac{104}{126}, \frac{115}{126}, 1$ for $i = 0, \ldots, 6$. The following base pattern information was also used in training and testing the networks:

Simulations I, III, IV		Simulation II	
$P = 6$		$P = 7$	
$D_{0i} = 1, 0, 0, 0, 0, 0$	a	$D_{0i} = 1, 0, 0, 0, 0, 0$	a
$D_{1i} = 1, 1, 0, 0, 0, 0$	$a\ b$	$D_{1i} = 1, 1, 0, 0, 0, 0$	$a\ b$
$D_{2i} = 1, 1, 1, 0, 0, 0$	$a\ b\ c$	$D_{2i} = 1, 1, 1, 0, 0, 0$	$a\ b\ c$
$D_{3i} = 0, 0, 1, 1, 0, 0$	$c\ d$	$D_{3i} = 1, 1, 1, 1, 0, 0$	$a\ b\ c\ d$
$D_{4i} = 0, 0, 0, 1, 1, 0$	$d\ e$	$D_{4i} = 0, 0, 1, 1, 0, 0$	$c\ d$
$D_{5i} = 0, 0, 0, 1, 1, 1$	$d\ e\ f$	$D_{5i} = 0, 0, 0, 1, 1, 0$	$d\ e$
—		$D_{6i} = 0, 0, 0, 1, 1, 1$	$d\ e\ f$

Computer Implementation

The simulations were run on a MasPar MP-1 parallel computer with 4096 processors. The simulation software used the Euler method, with a step size of $\frac{1}{750}$, to numerically integrate the differential equations. As a verification of the numerical stability of the integration, Simulation I was re-run with a step size of $\frac{1}{2000}$; the numerical results were virtually identical and were functionally equivalent.