# (1982)
## J. A. Feldman and D. H. Ballard

### Connectionist models and their properties
*Cognitive Science* 6:205–254

Much of the progress in the fields constituting cognitive science has been based upon the use of explicit information processing models, almost exclusively patterned after conventional serial computers. An extension of these ideas to massively parallel, connectionist models appears to offer a number of advantages. After a preliminary discussion, this paper introduces a general connectionist model and considers how it might be used in cognitive science. Among the issues addressed are: stability and noise-sensitivity, distributed decision-making, time and sequence problems, and the representation of complex concepts.

#### 1. Introduction

Much of the progress in the fields constituting cognitive science has been based upon the use of concrete information processing models (IPM), almost exclusively patterned after conventional sequential computers. There are several reasons for trying to extend IPM to cases where the computations are carried out by a parallel computational engine with perhaps billions of active units. As an introduction, we will attempt to motivate the current interest in massively parallel models from four different perspectives: anatomy, computational complexity, technology, and the role of formal languages in science. It is the last of these which is of primary concern here. We will focus upon a particular formalism, connectionist models (CM), which is based explicitly on an abstraction of our current understanding of the information processing properties of neurons.

Animal brains do not compute like a conventional computer. Comparatively slow (millisecond) neural computing elements with complex, parallel connections form a structure which is dramatically different from a high-speed, predominantly serial machine. Much of current research in the neurosciences is concerned with tracing out these connections and with discovering how they transfer information. One purpose of this paper is to suggest how connectionist theories of the brain can be used to produce stable, detailed models of interesting behaviors. The distributed nature of information processing in the brain is not a new discovery. The traditional view (which we shared) is that conventional computers and languages were Turing universal and could be made to simulate any parallelism (or analog values) which might be required. Contemporary computer science has sharpened our notions of what is "computable" to include bounds on time, storage, and other resources. It does not seem unreasonable to require that computational models in cognitive science be at least plausible in their postulated resource requirements.

The critical resource that is most obvious is time. Neurons whose basic computational speed is a few milliseconds must be made to account for complex behaviors which are carried out in a few hundred milliseconds (Posner, 1978). This means that *entire complex behaviors are carried out in less than a hundred time steps.* Current AI and simulation programs require millions of time steps. It may appear that the problem posed here is inherently unsolvable and that there is an error in our formulation. But recent results in computational complexity theory (Ja'Ja', 1980) suggest that networks of active computing elements can carry out at least simple computations in the required time range. In subsequent sections we present fast solutions to a variety of relevant computing problems. These solutions involve using massive numbers of units and connections, and we also address the questions of limitations on these resources.

Another recent development is the feasibility of building parallel computers. There is currently the capability to produce chips with 100,000 gates at a reproduction cost of a few cents each, and the technology to go to 1,000,000 gates/chip appears to be in hand. This has two important consequences for the study of CM. The obvious consequence is that it is now feasible to fabricate massively parallel computers, although no one has yet done so (Fahlman, 1980; Hillis, 1981). The second consequence of this development is the renewed interest in the basic properties of highly parallel computation. A major reason why there aren't yet any of these CM machines is that we do not yet know how to design, assemble, test, or program such engines. An important motivation for the careful study of CM is the hope that we will learn more about how to do parallel computing, but we will say no more about that in this paper.

The most important reason for a serious concern in cognitive science for CM is that they might lead to better science. It is obvious that the choice of technical language that is used for expressing hypotheses has a profound influence on the form in which theories are formulated and experiments undertaken. Artificial intelligence and articulating cognitive sciences have made great progress by employing models based on conventional digital computers as theories of intelligent behavior. But a number of crucial phenomena such as associative memory, priming, perceptual rivalry, and the remarkable recovery ability of animals have not yielded to this treatment. A major goal of this paper is to lay a foundation for the systematic use of massively parallel connectionist models in the cognitive sciences, even where these are not yet reducible to physiology or silicon.

Over the past few years, a number of investigators in different fields have begun to employ highly parallel models (idiosyncratically) in their work. The general idea has been advocated for animal models by Arbib (1979) and for cognitive models by Anderson (Anderson et al., 1977) and Ratcliff (1978). Parallel search of semantic memory and various "spreading activation" theories have become common (though not quite consistent) parts of information processing modeling. In machine perception research, massively parallel, cooperative computational theories have become a dominant paradigm (Marr & Poggio, 1976; Rosenfeld et al., 1976) and many of our examples come from our own work in this area (Ballard, 1981; Sabbah, 1981). Scientists looking at performance errors and other nonrepeatable behaviors have not found conventional IPM to be an adequate framework for their efforts. Norman (1981) has recently summarized arguments from cognitive psychology, and Kinsbourne and Hicks (1979) have been led to a similar view from a different perspective. It appears to us that all of these efforts could fit within the CM paradigm outlined here.

One of the most interesting recent studies employing CM techniques is the partial theory of reading developed in (McClelland & Rumelhart, 1981). They were concerned with the word superiority effect and related questions in the perception of printed words, and had a large body of experimental data to explain. One major finding is that the presence of a printed letter in a brief display is easier to determine when the letter is presented in the context of a word than when it is presented alone. The model they developed (cf. Figure 1) explicitly represents three levels of processing: visual features of printed letters, letters, and words. The

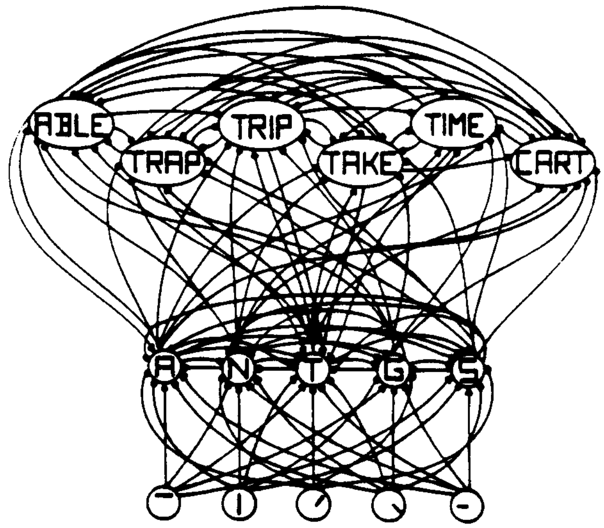



**Figure 1** A few of the neighbors of the node for the letter "t" in the first position in a word, and their interconnections (McClelland & Rumelhart, 1981).

model assumes that there are positive and negative (circular tipped) connections from visual features to the letters that they can (respectively, cannot) be part of. The connections between letters and words can go in either direction and embody the constraints of English. The model assumes that many units can be simultaneously active, that units form algebraic sums of their inputs and output values proportionally. The activity of a unit is bounded from above and below, has some memory and decays with time. All of these features, and several more, are captured in the abstract unit described in Section 2.

This idea of simultaneously evaluating many hypotheses (here words) has been successfully used in machine perception for some time (Hanson & Riseman, 1978). What has occurred to us relatively recently is that this is a natural model of computation for widely interconnected networks of active elements like those envisioned in connectionist models. The generalization of these ideas to the connectionist view of brain and behavior is that all important encodings in the brain are in terms of the relative strengths of synaptic connections. The fundamental premise of connectionism is that individual neurons *do not transmit large amounts of symbolic information*. Instead they compute by being *appropriately connected* to large numbers of similar units. This is in sharp contrast to the conventional computer model of intelligence prevalent in computer science and cognitive psychology.

The fundamental distinction between the conventional and connectionist computing models can be

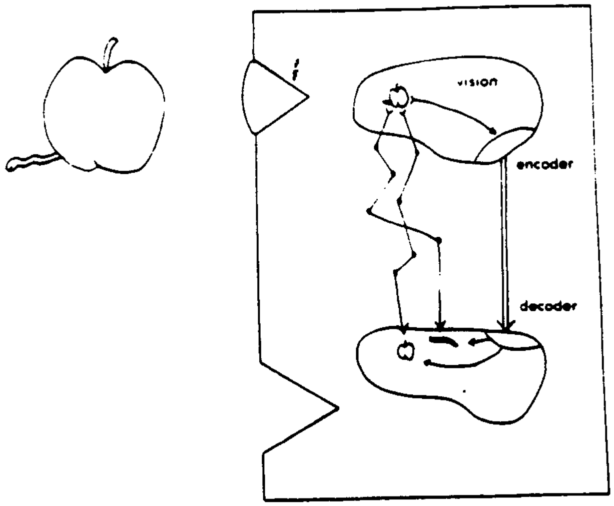



Figure 2 Connectionism vs. symbolic encoding. ⇒ assumes some general encoding. → assumes individual connections.

conveyed by the following example. When one sees an apple and says the phrase "wormy apple," some information must be transferred, however indirectly, from the visual system to the speech system. Either a sequence of special *symbols* that denote a wormy apple is transmitted to the speech system, or there are special *connections* to the speech command area for the words. Figure 2 is a graphic presentation of the two alternatives. The path on the right described by double-lined arrows depicts the situation (as in a computer) where the information that a wormy apple has been seen is encoded by the visual system and sent as an abstract message (perhaps frequency-coded) to a general receiver in the speech system which decodes the message and initiates the appropriate speech act. Notice that a complex message would presumably have to be transmitted sequentially on this channel, and that each end would have to learn the common code for every new concept. No one has yet produced a biologically and computationally plausible realization of this conventional computer model.

The only alternative that we have been able to uncover is described by the path with single-width arrows. This suggests that there are (indirect) links from the units (cells, columns, centers, or what-have-you) that recognize an apple to some units responsible for speaking the word. The connectionist model requires only very simple messages (e.g. stimulus strength) to cross a channel but puts strong demands on the availability of the right connections. Questions concerning the learning and reinforcement of connections are addressed in Feldman, (1981b).

For a number of reasons (including redundancy for reliability), it is highly unlikely that there is exactly one neuron for each concept, but the point of view taken here is that the activity of a small number of neurons (say 10) encodes a concept like apple. An alternative view (Hinton & Anderson, 1981) is that concepts are represented by a "pattern of activity" in a much larger set of neurons (say 1,000) which also represent many other concepts. We have not seen how to carry out a program of specific modeling in terms of these diffuse models. One of the major problems with diffuse models as a parallel computation scheme is cross-talk among concepts. For example, if concepts using units (10, 20, 30, ...) and (5, 15, 25, ...) were simultaneously activated, many other concepts, e.g., (20, 25, 30, 35, ...) would be active as well. In the example of Figure 2, this means that diffuse models would be more like the shared sequential channel. Although a single concept could be transmitted in parallel, complex concepts would have to go one at a time. Simultaneously transmitting multiple concepts that shared units would cause cross-talk. It is still true in our CM that many related units will be triggered by spreading activation, but the representation of each concept is taken to be compact.

Most cognitive scientists believe that the brain appears to be massively parallel and that such structures can compute special functions very well. But massively parallel structures do not seem to be usable for general purpose computing and there is not nearly as much knowledge of how to construct and analyze such models. The common belief (which may well be right) is that there are one or more intermediate levels of computational organization layered on the neuronal structure, and that theories of intelligent behavior should be described in terms of these higher-level languages, such as Production Systems, Predicate Calculus, or LISP. We have not seen a reduction (interpreter, if you will) of any higher formalism which has plausible resource requirements, and this is a problem well worth pursuing.

Our attempts to develop cognitive science models directly in neural terms might fail for one of two reasons. It may be that there really is an interpreted symbol system in animal brains. In this case we would hope that our efforts would break down in a way that could shed light on the nature of this symbol system. The other possibility is that CM techniques are directly applicable but we are unable to figure out how to model some important capacity, e.g., planning. Our program is to continue the CM attack on problems of increasing difficulty (and to induce some of you to join us) until we encounter one that is intractable in our terms. There are a number of problems that are known to be difficult for systems without an interpreted symbolic representation, including complex concepts, learning, and natural language understanding. The current paper is mainly concerned with laying out the formalism and showing how it applies in the easy cases, but we do address the problem of complex concepts in Section 4. We have made some progress on the problem of learning in CM systems (Feldman, 1981b) and are beginning to work seriously on natural language processing and on higher-level vision. Our efforts on planning and long-term memory reorganization have not advanced significantly beyond the discursive presentation in (Feldman, 1980).

We will certainly not get very far in this program without developing some systematic methods of attacking CM tasks and some building-block circuits whose properties we understand. A first step towards a systematic development of CM is to define an abstract computing unit. Our unit is rather more general than previous proposals and is intended to capture the current understanding of the information processing capabilities of neurons. Some useful special cases of our general definition and some properties of very simple networks are developed in Section 2. Among the key ideas are local memory, non-homogeneous and non-linear functions, and the notions of mutual inhibition and stable coalitions.

A major purpose of the rest of the paper is to describe building blocks which we have found useful in constructing CM solutions to various tasks. The constructions are intended to be used to make specific models but the examples in this paper are only suggestive. We present a number of CM solutions to general problems arising in intelligent behavior, but *we are not suggesting that any of these are necessarily employed by nature*. Our notion of an adequate model is one that accounts for *all* of the established relevant findings and this is not a task to be undertaken lightly. We are developing some preliminary sketches (Ballard & Sabbah, 1981; Sabbah, 1981) for a serious model of low and intermediate level vision. As we develop various building blocks and techniques we will also be trying to bury some of the contaminated debris of past neural modeling efforts. Many of our constructions are intended as answers to known hard problems in CM computation. Among the issues addressed are: stability and noise-sensitivity, distributed decision-making, time and sequence problems, and the representation of complex concepts. The crucial questions of learning and change in CM systems are discussed elsewhere (Feldman, 1981b).

## 2. Neuron-Like Computing Units

As part of our effort to develop a generally useful framework for connectionist theories, we have developed a standard model of the individual unit. It will turn out that a "unit" may be used to model anything from a small part of a neuron to the external functionality of a major subsystem. But the basic notion of unit is meant to loosely correspond to an information processing model of our current understanding of neurons. The particular definitions here were chosen to make it easy to specify detailed examples of relatively complex behaviors. There is no attempt to be minimal or mathematically elegant. The various numerical values appearing in the definitions are arbitrary, but fixed finite bounds play a crucial role in the development. The presentation of the definitions will be in stages, accompanied by examples. A compact technical specification for reference purposes is included as Appendix A. Each unit will be characterized by a small number of discrete states plus:

p—a continuous value in $[-10, 10]$, called *potential* (accuracy of several digits)
v—an *output value*, integers $0 \leq v \leq 9$
*i*—a vector of *inputs* $i_1, \ldots, i_n$

### P-Units

For some applications, we will be able to use a particularly simple kind of unit whose output v is proportional to its potential p (rounded) when $p > 0$ and which has only one state. In other words

$$p \leftarrow p + \beta \sum w_k i_k \qquad [0 \leq w_k \leq 1]$$
$$v \leftarrow \textit{if } p > \theta \textit{ then } \text{round}(p - \theta) \textit{ else } 0 \qquad [v = 0 \ldots 9]$$

where $\beta$, $\theta$ are constants and $w_k$ are weights on the input values. The weights are the sole locus of change with experience in the current model. Most often, the potential and output of a unit will be encoding its *confidence*, and we will sometimes use this term. The "$\leftarrow$" notation is borrowed from the assignment statement of programming languages. This notation covers both continuous and discrete time formulations and allows us to talk about some issues without any explicit mention of time. Of course, certain other questions will inherently involve time and computer simulation of any network of units will raise delicate questions of discretizing time.

The restriction that output take on small integer values is central to our enterprise. The firing frequencies of neurons range from a few to a few hundred impulses per second. In the 1/10 second needed for basic mental events, there can only be a limited amount of information encoded in frequencies. Then ten output values are an attempt to capture this idea. A more accurate rendering of neural events would be to allow 100 discrete values with noise on transmission (cf. Sejnowski, 1977). Transmission time is assumed to be negligible; delay units can be added when transit time needs to be taken into account.

The p-unit is somewhat like classical linear threshold elements (Minsky & Papert, 1972), but there are several differences. The potential, p, is a crude form of memory and is an abstraction of the instantaneous membrane potential that characterizes neurons; it greatly reduces the noise sensitivity of our networks. Without local memory in the unit, one must guarantee that all the inputs required for a computation appear simultaneously at the unit.

One problem with the definition above of a p-unit is that its potential does not decay in the absence of input. This decay is both a physical property of neurons and an important computational feature for our highly parallel models. One computational trick to solve this is to have an inhibitory connection from the unit back to itself. Informally, we identify the negative self feedback with an exponential decay in potential which is mathematically equivalent. With this addition, p-units can be used for many CM tasks of intermediate difficulty. The Interactive Activation models of McClelland and Rumelhart can be described naturally with p-units, and some of our own work (Ballard, 1981) and that of others (Marr & Poggio, 1976) can be done with p-units. But there are a number of additional features which we have found valuable in more complex modeling tasks.

### Disjunctive Firing Conditions and Conjunctive Connections

It is both computationally efficient and biologically realistic to allow a unit to respond to one of a number of alternative conditions. One way to view this is to imagine the unit having "dendrites" each of which depicts an alternative enabling condition (Figure 3). For example, one could extend the network of Figure 1 to allow for several different type fonts activating the same letter node, with the higher connections unchanged. Biologically, the firing of a neuron depends, in many cases, on local spatio-temporal summation involving only a small part of the neuron's surface. So-called dendritic spikes transmit the activation to the rest of the cell.

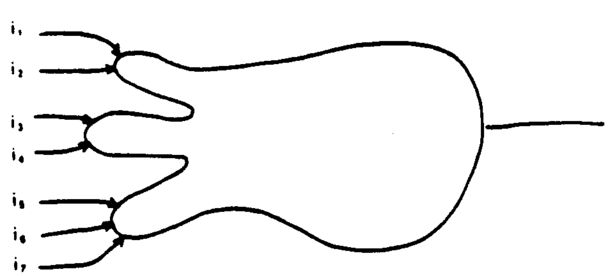



**Figure 3** Conjunctive connections and disjunctive input sites.

In terms of our formalism, this could be described in a variety of ways. One of the simplest is to define the potential in terms of the maximum of the separate computations, e.g.,

$$p \leftarrow p + \beta \operatorname{Max}(i_1 + i_2 - \varphi, i_3 + i_4 - \varphi, i_5 + i_6 - i_7 - \varphi)$$

where $\beta$ is a scale constant as in the p-unit and $\varphi$ is a constant chosen (usually $> 10$) to suppress noise and require the presence of multiple active inputs (Sabbah, 1981). The minus sign associated with $i_7$ corresponds to its being an inhibitory input.

It does not seem unreasonable (given current data, Kuffler & Nicholls, 1976) to model the firing rate of some units as the maximum of the rates at its active sites. Units whose potential is changed according to the maximum of a set of algebraic sums will occur frequently in our specific models. One advantage of keeping the processing power of our abstract unit close to that of a neuron is that it helps inform our counting arguments. When we attempt to model a particular function (e.g., stereopsis), we expect to require that the number of units and connections as well as the execution time required by the model are plausible.

The max-of-sum unit is the continuous analog of a logical OR-of-AND (disjunctive normal form) unit and we will sometimes use the latter as an approximate version of the former. The OR-of-AND unit corresponding to Figure 3 is:

$$p \leftarrow p + \alpha \operatorname{OR}(i_1 \& i_2, i_3 \& i_4, i_5 \& i_6 \& (\text{not } i_7))$$

This formulation stresses the importance that nearby spatial connections *all* be firing before the potential is affected. Hence, in the above example, $i_3$ and $i_4$ make a *conjunctive connection* with the unit. The effect of a conjunctive connection can always be simulated with more units but the number of extra units may be very large.

### Q-Units and Compound Units

Another useful special case arises when one suppresses the numerical potential, p, and relies upon a finite-state set $\{q\}$ for modeling. If we also identify each input of i with a separate named input signal, we can get classical finite automata. A simple example would be a unit that could be started or stopped from firing.

One could describe the behavior of this unit by a table, with rows corresponding to states in $\{q\}$ and columns to possible inputs, e.g.,

| | $i_1$ (start) | $i_2$ (stop) |
|---|---|---|
| Firing | Firing | Null |
| Null | Firing | Null |

One would also have to specify an output function, giving output values required by the rest of the network, e.g.,

$v \leftarrow$ if $q$ = Firing then 6 else 0.

This could also be added to the table above. An equivalent notation would be transition networks with states as nodes and inputs and outputs on the arcs.

In order to build models of interesting behaviors we will need to employ many of the same techniques used by designers of complex computers and programs. One of the most powerful techniques will be encapsulation and abstraction of a subnetwork by an individual unit. For example, a system that had separate motor abilities for turning left and turning right (e.g., fins) could use two start-stop units to model a turn-unit, as shown in Figure 4.

Note that the compound unit here has two distinct outputs, when basic units have only one (which can branch, of course). In general, compound units will differ from basic ones only in that they can have several distinct outputs.

The main point of this example is that the turn-unit can be described abstractly, independent of the details of how it is built. For example, using the tabular conventions described above,

| | Left | Right | Values Output |
|---|---|---|---|
| a gauche | a gauche | a droit | $v_1 = 7,\ v_2 = 0$ |
| a droit | a gauche | a droit | $v_1 = 0,\ v_2 = 8$ |

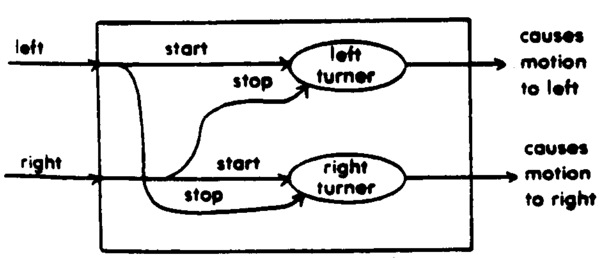



Figure 4 A Turn Unit.

where the right-going output being larger than the left could mean that we have a right-finned robot. There is a great deal more that must be said about the use of states and symbolic input names, about multiple simultaneous inputs, etc., but the idea of describing the external behavior of a system only in enough detail for the task at hand is of great importance. This is one of the few ways known of coping with the complexity of the magnitude needed for serious modeling of biological functions. It is not strictly necessary that the same formalism be used at each level of functional abstraction and, in the long run, we may need to employ a wide range of models. For example, for certain purposes one might like to expand our units in terms of compartmental models of neurons like those of (Perkel, 1979). The advantage of keeping within the same formalism is that we preserve intuition, mathematics, and the ability to use existing simulation programs. With sufficient care, we can use the units defined above to represent large subsystems without giving up the notion that each unit can stand for an abstract neuron. The crucial point is that a subsystem must be elaborated into its neuron-level units for timing and size calculations, but can (hopefully) be described much more simply when only its effects on other subsystems are of direct concern.

### Units Employing p and q

It will already have occurred to the reader that a numerical value, like our p, would be useful for modeling the amount of turning to the left or right in the last example. It appears to be generally true that a single numerical value and a small set of discrete states combine to provide a powerful yet tractable modeling unit. This is one reason that the current definitions were chosen. Another reason is that the mixed unit seems to be a particularly convenient way of modeling the information processing behavior of neurons, as generally described. The discrete states enable one to model the effects in neurons of polypeptide modulators, abnormal chemical environments, fatigue, etc. Although these effects are often continuous functions of unit parameters, there are several advantages to using discrete states in our models. Scientists and laymen alike often give distinct names (e.g., cool, warm, hot) to parameter ranges that they want to treat differently. We also can exploit a large literature on understanding loosely-coupled systems as finite-state machines (Sunshine, 1979). It is also traditional to break up a function into separate ranges when it is simpler to describe that way. We have already employed all of these uses of discrete states in our detailed work (Feldman, 1981b; Sabbah, 1981). One example of a unit employing both p and q non-trivially is the following crude neuron model. This model is concerned with saturation and assumes that the output strength, v, is something like average firing frequency. It is not a model of individual action potentials and refractory periods.

We suppose the distinct states of the unit $q \in$ {normal, recover}. In *normal* state the unit behaves like a p-unit, but while it is *recovering* it ignores inputs. The following table captures almost all of this behavior.

| | | $-1 < p < 9$ | $p > 9$ | Output Value |
|---|---|---|---|---|
| (incomplete) | normal | $p \leftarrow p + \Sigma i$ | $p \leftarrow -p/$ recover | $v \leftarrow \alpha p - \theta$ |
| | recover | normal | ⟨impossible⟩ | $v \leftarrow 0$ |

Here we have the change from one state to the other depending on the value of the potential, p, rather than on specific inputs. The recovering state is also characterized by the potential being set negative. The unspecified issue is what determines the duration of the recovering state—there are several possibilities. One is an explicit dishabituation signal like those in Kandel's experiments (Kandel, 1976). Another would be to have the unit sum inputs in the recovering state as well. The reader might want to conside how to add this to the table.

The third possibility, which we will use frequently, is to assume that the potential, p, decays toward zero (from both directions) unless explicitly changed. This implicit decay $p \leftarrow p_0 e^{-kt}$ can be modeled by self inhibition; the decay constant, k, determines the length of the recovery period.

The general definition of our abstract neural computing unit is just a formalization of the ideas presented above. To the previous notions of p, v, and *i* we formally add

{q}—a set of *discrete states*, < 10

and functions from old to new values of these

$p \leftarrow f(i, p, q)$
$q \leftarrow g(i, p, q)$
$v \leftarrow h(i, p, q)$

which we assume, for now, to compute continuously. The form of the f, g, and h functions will vary, but will generally be restricted to conditionals and simple functions. There are both biological and computational reasons for allowing units to respond (for example) logarithmically to their inputs and we have already seen important uses of the maximum function.

The only other notion that we will need is modifiers associated with the inputs of a unit. We elaborate the input vector *i* in terms of received values, weights, and modifiers:

$$\forall j,\ i_j = r_j \cdot w_j \cdot m_j \qquad j = 1, \ldots, n$$

where $r_j$ is the *value* received from a predecessor $[r = 0 \ldots 9]$; $w_j$ is a changeable *weight*, unsigned $[0 \leq w_j \leq 1]$ (accuracy of several digits); and m is a synapto-synaptic *modifier* which is either 0 or 1.

The weights are the only thing in the system which can change with experience. They are unsigned because we do not want a connection to change from excitatory to inhibitory. The modifier or gate simplifies many of our detailed models. Learning and change will not be treated technically in this paper, but the definitions are included in the Appendix for completeness (Feldman, 1981b).

We conclude this section with some preliminary examples of networks of our units, illustrating the key idea of mutual (lateral) inhibition (Fig. 5). Mutual inhibition is widespread in nature and has been one of the basic computational schemes used in modeling. We will present two examples of how it works to help aid in intuition as well as to illustrate the notation. The basic situation is symmetric configurations of p-units which mutually inhibit one another. Time is broken into discrete intervals for these examples. The examples are too simple to be realistic, but do contain ideas which we will employ repeatedly.

### Two P-Units Symmetrically Connected

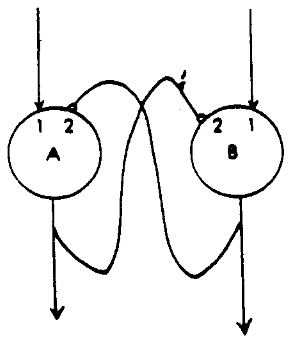
Suppose

$w_1 = 1, \quad w_2 = -.5$
$p(t + 1) = p(t) + r_1 - (.5)r_2 \qquad r_j = \text{received}$
$v = \text{round}(p)[0 \ldots 9]$

Referring to Figure 5a, suppose the initial input to the unit A.1 is 6, then 2 per time step, and the initial input to B.1 is 5, then 2 per time step. At each time step, each unit changes its potential by adding the external value ($r_1$) and substracting half the output value of its rival. This system will stabilize to the side of the larger of two instantaneous inputs.

### Two Symmetric Coalitions of 2-Units

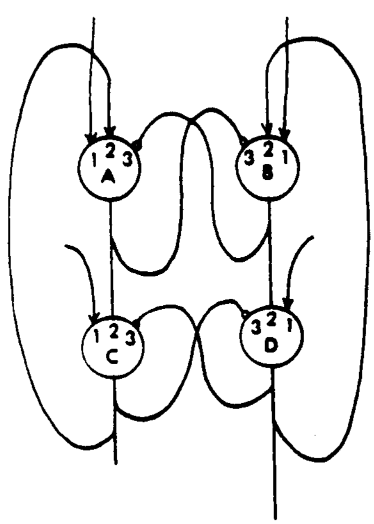
$w_1 = 1$
$w_2 = .5$
$w_3 = -.5$
$p(t + 1) = p(t) + r_1 + .5(r_2 - r_3)$
$v = \text{round}(p)$
A, C start at 6; B, D at 5;
A, B, C, D have no external input for $t > 1$




Suppose A, received an input of 6 units, then 2 per time step
Suppose B, received an input of 5 units, then 2 per time step

| t | P(A) | P(B) |
|---|---|---|
| 1 | 6 | 5 |
| 2 | 5.5 | 4 |
| 3 | 5.5 | 3.5 |
| 4 | 6 | 3 |
| 5 | 6.5 | 2 |
| 6 | 7.5 | 1 |
| 7 | 9.5 | 0 |
| 8 | Sat | 0 |

(a)




| t | P(A) | P(B) | P(C) | P(D) |
|---|---|---|---|---|
| 1 | 6 | 5 | 6 | 5 |
| 2 | 6.5 | 4.5 | 6.5 | 4.5 |
| 3 | 7.5 | 3.5 | 7.5 | 3.5 |
| 4 | 9.5 | 1.5 | 9.5 | 1.5 |
| | Sat | 0 | Sat | 0 |

(b)

**Figure 5** Small Symmetric Networks.

The connections for this system are shown in Figure 5b. This system converges faster than the previous example. The idea here is that units A and C form a "coalition" with mutually reinforcing connections. The competing units are A vs. B and C vs. D. The last example is the smallest network depicting what we believe to be the basic mode of operation in connectionist systems. The faster convergence is not an artifact; the *positive feedback* among members of a coalition will generally lead to faster convergence than in separate competitions. It is the amount of positive feedback rather than just the size of the coalition that determines the rate of convergence (Feldman & Ballard, 1982). In terms of Figure 1, this could represent the behavior of the rival letters A and T in conjunction with the rival words ABLE and TRAP, in the absence of other active nodes.

Competing coalitions of units will be the organizing principle behind most of our models. Consider the two alternative readings of the Necker cube shown in Figure 6. At each level of visual processing, there are mutually contradictory units representing alternative possibilities. The dashed lines denote the boundaries of coalitions which embody the alternative interpretations of the image. A number of interesting phenomena (e.g., priming, perceptual rivalry, filling, subjective contour) find natural expression in this formalism. We are engaged in an ongoing effort (Ballard, 1981; Sabbah, 1981) to model as much of visual processing as possible within the connectionist framework. The next section describes in some detail a variety of simple networks which we have found to be useful in this effort.

## 3. Networks of Units

The main restriction imposed by the connectionist paradigm is that no symbolic information is passed from unit to unit. This restriction makes it difficult to employ standard computational devices like parameterized functions. In this section, we present connectionist solutions to a variety of computational problems. The sections address two principal issues. One is: Can the networks be connected up in a way that is sufficient to represent the problem at hand? The other is: Given these connections, how can the networks exhibit appropriate dynamic behavior, such as making a decision at an appropriate time?

### Using a Unit to Represent a Value

One key to many of our constructions is the dedication of a separate unit to each value of each parameter of interest, which we term the unit/value principle. We

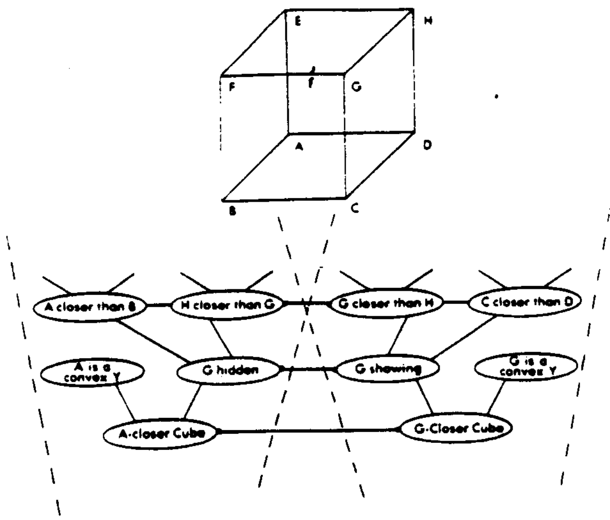



**Figure 6** The Necker Cube.

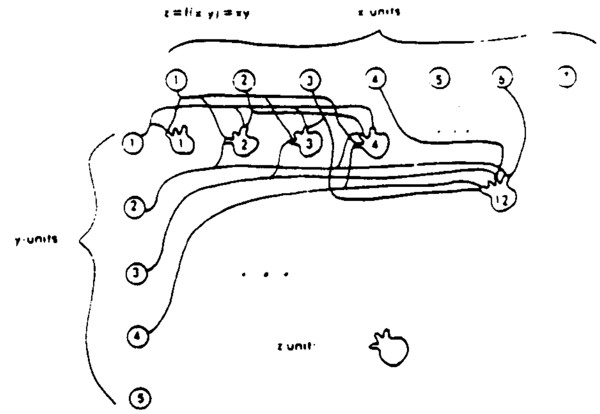



**Figure 7** Multiplication Units.

will show how to compute using unit/value networks and present arguments that the number of units required is not unreasonable. In this representation the output of a unit may be thought of as a confidence measure. Suppose a network of depth units encodes the distance of some object from the retina. Then if the unit representing depth = 2 saturates, the network is expressing confidence that the distance is two units. Similarly, the "G-hidden" node in Figure 6 expresses confidence in its assertion. There is much neurophysiological evidence to suggest unit/value organizations in less abstract cortical maps. Examples are edge sensitive units (Hubel & Wiesel, 1979) and perceptual color units (Zeki, 1980), which are relatively insensitive to illumination spectra. Experiments with cortical motor control in the monkey and cat (Wurtz & Albano, 1980) suggest a unit/value organization. Our hypothesis is that the unit/value organization is widespread, and is a fundamental design principle.

Although many physical neurons do seem to follow the unit/value rule and respond according to the reliability of a particular configuration, there are also other neurons whose output represents the range of some parameter, and apparently some units whose firing frequency reflects both range and strength information (Scientific American, 1979). Both of the latter types can be accommodated within our definition of a unit, but we will employ only unit/value networks in the remainder of this paper.

In the unit/value representation, much computation is done by table look-up. As a simple example, let us consider the multiplication of two variables, i.e., $z = xy$. In the unit/value formalism there will be units for *every* value of x and y that is important. Appropriate pairs of these will make a conjunctive connection with another unit cell representing a specific value for the product. Figure 7 shows this for a small set of units representing values for x and y. Notice that the confidence (expressed as output value) that a particular product is an answer can be a linear function of the maximum of the sums of the confidences of its two inputs. A major problem with function tables (and with CM in general) is the potential combinatorial explosion in the number of units required for a computation. A naive approach would demand $N^2$ units to represent all products of numbers from 1 to N. The network of Figure 7 requires many fewer units because each product is represented only once, another advantage of conjunctive connections. We could use even fewer units by exploiting positional notation and replacing each output connection with a conjunction of outupts from units representing multiples of 1, 10, 100, etc. The question of efficient ways of building connection networks is treated in detail in Section 4 (cf. also Hinton, 1981a; 1981b).

### Modifiers and Mappings

The idea of function tables (Fig. 7) can be extended through the use of *variable mappings*. In our definition of the computational unit, we included a binary modifier, m, as an option on every connection. As the definition specifies, if the modifier associated with a connection is zero, the value v sent along that connection is ignored. Thus the modifier denotes inhibition, or blocking. There is considerable evidence in nature for synapses on synapses (Kandel, 1976) and the modifiers add greatly to the computational simplicity of our networks. Let us start with an initial informal example of the use of modifiers and mappings. Suppose that one has a model of grass as green except in California where

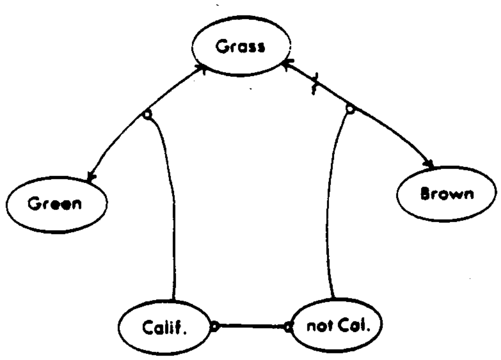



**Figure 8** Grass is Green connection modified by California.

it is brown (golden), as shown in Figure 8. Here we can see that grass and green are potential members of a coalition (can reinforce one another) except when the link is blocked. This use is similar to the cancellation link of (Fahlman, 1979) and gives a crude idea of how context can effect perception in our models. Note that in Figure 8 we are using a shorthand notation. A modifier touching a double-ended arrow actually blocks two connections. (Sometimes we also omit the arrowheads when connection is double-ended.)

Mappings can also be used to select among a number of possible values. Consider the example of the relation between depth, physical size, and retinal size of a circle. (For now, assume that the circle is centered on and orthogonal to the line of sight, that the focus is fixed, etc.) Then there is a fixed relation between the size of retinal image and the size of the physical circle for any given depth. That is, each depth specifies a *mapping* from retinal to physical size (see Fig. 9). Here we suppose the scales for depth and the two sizes are chosen so that unit depth means the same numerical size. If we knew the depth of the object (by touch, context, or magic) we would know its physical size. The network above allows retinal size 2 to reinforce physical size 2 when depth = 1 but inhibits this connection for all other depths. Similarly, at depth 3, we should interpret retinal size 2 as physical size 8, and inhibit other interpretations. Several remarks are in order. First, notice that this network implements a function phys = f(ret, dep) that maps from retinal size and depth to physical size, providing an example of how to replace functions with parameters by mappings. For the simple case of looking at one object perpendicular to the line of sight, there will be one consistent coalition of units which will be stable. The work does something more, and this is crucial to our enterprise; the network can represent the consistency relation R among the three quantities: depth, retinal size, and physical size. It embodies not only the function f, but its two in-

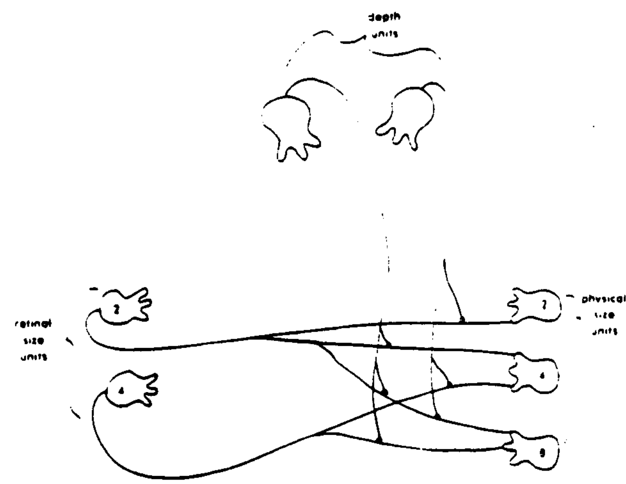



**Figure 9** Depth Network using Modifiers.

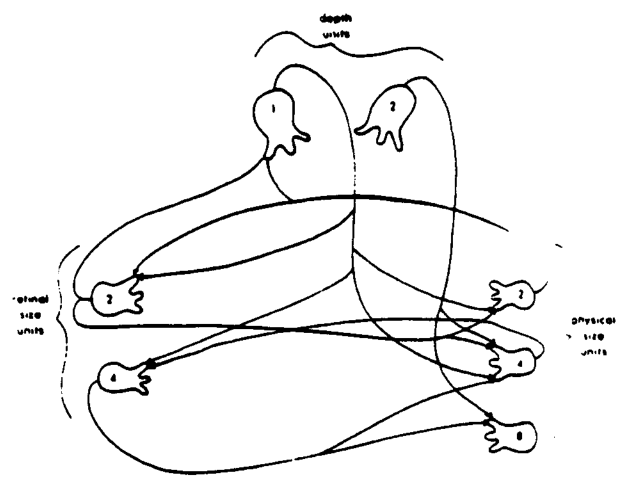



**Figure 10** Depth Network using Conjunctive Connections.

verse functions as well (dep = $f_1$(ret, phys), and ret = $f_2$(phys, dep)). (The network as shown does not include the links for $f_1$ and $f_2$, but these are similar to those for f.) Most of Section 5 is devoted to laying out networks that embody theories of particular visual consistency relations.

The idea of modifiers is, in a sense, complementary to that of conjunctive connections. For example, the network of Figure 9 could be transformed into the following network (Fig. 10). In this network the variables for physical size, depth, and retinal size are all given equal weight. For example, physical size = 4 and depth = 1 make a *conjunctive connection* with retinal size = 4. Each of the value units in a competing row could be connected to all of its competitors by inhibitory links and this would tend to make the network activate only one value in each category. The general issue of rivalry and coalitions will be discussed in the next two sub-sections.

When should a relation be implemented with modifiers and when should it be implemented with conjunctive connections? A simple, nonrigorous answer to this question can be obtained by examining the size of two sets of units: (1) the number of units that would have to be inhibited by modifiers; and (2) the number of units that would have to be reinforced with conjunctive connections. If (1) is larger than (2), then one should choose modifiers; otherwise choose conjunctive connections. Sometimes the choice is obvious: to implement the brown Californian grass example of Figure 8 with conjunctive connections, one would have to reinforce all units representing places that had green grass! Clearly in this case it is easier to handle the exception with modifiers. On the other hand, the depth relation R(phy, dep, ret) is more cheaply implemented with conjunctive connections. Since our modifiers are strictly binary, conjunctive connections have the additional advantage of continuous modulation.

To see how the conjunctive connection strategy works in general, suppose a constraint relation to be satisfied involves a variable x, e.g., $f(x, y, z, w) = 0$. For a particular value of x, there will be triples of values of y, z, and w that satisfy the relation f. Each of these triples should make a conjunctive connection with the unit representing the x-value. There could also be 3-input conjunctions at each value of y, z, w. Each of these four different kinds of conjunctive connections corresponds to an interpretation of the *relation* $f(x, y, z, w) = 0$ as a *function*, i.e., $x = f_1(y, z, w)$, $y = f_2(x, z, w)$, $z = f_3(x, y, w)$, or $w = f_4(x, y, z)$. Of course, these functions need not be single-valued. This network connection pattern could be extended to more than four variables, but high numbers of variables would tend to increase its sensitivity to noisy inputs. Hinton has suggested a special notation for the situation where a network exactly captures a consistency relation. The mutually consistent values are all shown to be centrally linked (Fig. 11). This notation provides an elegant way of presenting the interactions among networks, but must be used with care. Writing down a triangle diagram does not insure that the underlying mappings can be made consistent or computationally well-behaved.

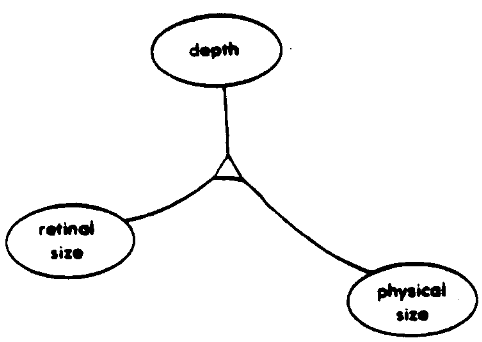



Figure 11 Notation for consistency relations.

**Winner-Take-All Networks and Regulated Networks**

A very general problem that arises in any distributed computing situation is how to get the entire system to make a decision (or perform a coherent action, etc.). Biologically necessary examples of this behavior abound; ranging from turning left or right, through fight-or-flight responses, to interpretations of ambiguous words and images. Decision-making is a particularly important issue for the current model because of its restrictions on information flow and because of the almost linear nature of the p-units used in many of our specific examples. Decision-making introduces the notions of *stable states* and *convergence* of networks.

One way to deal with the issue of coherent decisions in a connectionist framework is to introduce *winner-take-all* (WTA) networks, which have the property that only the unit with the highest potential (among a set of contenders) will have output above zero after some setting time (Fig. 12). There are a number of ways to construct WTA networks from the units described above. For our purposes it is enough to consider one example of a WTA network which will operate in one time step for a set of contenders each of whom can read the potential of all of the others. Each unit in the network computes its new potential according to the rule:

$p \leftarrow$ *if* $p > \max(i_j, .1)$ *then* p *else* 0.

That is, each unit sets itself to zero if it knows of a higher input. This is fast and simple, but probably a little too complex to be plausible as the behavior of a single neuron. There is a standard trick (apparently widely used by nature) to convert this into a more plausible scheme. Replace each unit above with two

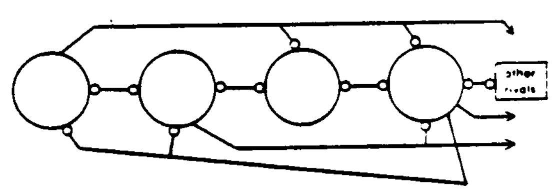



Figure 12 Winner-Take-All. Each unit stops if it sees a higher value.

units; one computes the maximum of the competitor's inputs and inhibits the other. The circuit above can be strengthened by adding a reverse inhibitory link, or one could use a modifier on the output, etc. Obviously one could have a WTA layer that got intputs from some set of competitors and settled to a winner when triggered to do so by some downstream network. This is an exact analogy of strobing an output buffer in a conventional computer.

One problem with previous neural modeling attempts is that the circuits proposed were often unnaturally delicate (unstable). Small changes in parameter values would cause the networks to oscillate or converge to incorrect answers. We will have to be careful not to fall into this trap, but would like to avoid detailed analysis of each particular model for delicacy in this paper. What appears to be required are some building blocks and combination rules that preserve the desired properties. For example, the WTA subnetworks of the last example will not oscillate in the absence of oscillating inputs. This is also true of any symmetric mutually inhibitory subnetwork. This is intuitively clear and could be proven rigorously under a variety of assumptions (cf. Grossberg, 1980). If every unit receives inhibition proportional to the activity (potential) of each of its rivals, the instantaneous leader will receive less inhibition and thus not lose its lead unless the inputs change significantly.

Another useful principle is the employment of lower-bound and upper-bound cells to keep the total activity of a network within bounds (Fig. 13). Suppose that we add two extra units, LB and UB, to a network which has coordinated output. The LB cell compares the total (sum) activity of the units of the network with a lower bound and sends positive activation uniformly to all members if the sum is too low. The UB cell inhibits all units equally if the sum of activity is too high. Notice that LB and UB can be parameters set from outside the nework. Under a wide range of conditions (but not all), the LB-UB augmented network can be designed to preserve order relationships among the outputs $v_j$ of the original network while keeping the sum between LB and UB.

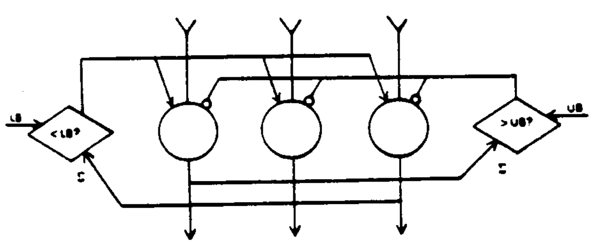


**Figure 13** Regulated Network. If sum exceeds UB all units get uniform inhibition.

We will often assume that LB-UB pairs are used to keep the sum of outputs from a nework within a given range. This same mechanism also goes far towards eliminating the twin perils of uniform saturation and uniform silence which can easily arise in mutual inhibition networks. Thus we will often be able to reason about the computation of a network assuming that it stays active and bounded.

### Stable Coalitions

For a massively parallel system to actually make a decision (or do something), there will have to be states in which some activity strongly dominates. Such stable, connected, high confidence units are termed *stable coalitions*. A stable coalition is our architecturally-biased term for the psychological notions of percept, action, etc. We have shown some simple instances of stable coalitions, in Figure 5b and the WTA network. In the depth networks of Figures 9 and 10, a stable coalition would be three units representing consistent values of retinal size, depth, and physical size. But the general idea is that a very large complex subsystem must stabilize, e.g., to a fixed interpretation of visual input, as in Figure 1. The way we believe this to happen is through mutually reinforcing coalitions which dominate all rival activity when the decision is required. The simplest case of this is Figure 5b, where the two units A and B form a coalition which suppresses C and D. Formally, *a coalition will be called stable when the output of all its members is non-decreasing.* Notice that a coalition is not a particular anatomical structure, but an instantaneously mutually reinforcing set of units, in the spirit of Hebb's cell assemblies (Jusczyk & Klein, 1980).

What can we say about the conditions under which coalitions will become and remain stable? We will begin informally with an almost trivial condition. Consider a set of units $\{a, b, \ldots\}$ which we wish to examine as a possible coalition, $\pi$. For now, we assume that the units in $\pi$ are all p-units and are in the non-saturated range and have no decay. Thus for each u in $\pi$,

$$p(u) \leftarrow p(u) + \text{Exc} - \text{Inh},$$

where Exc is the weighted sum of excitatory inputs and Inh is the weighted sum of inhibitory inputs. Now suppose that Exc$|\pi$, the excitation from the coalition $\pi$ only, were greater than INH, the largest possible inhibition receivable by u, for each unit u in $\pi$, i.e.,

$$\text{(SC)} \qquad \forall\, u \in \pi;\ \text{Exc}|\pi > \text{INH}$$

Then it follows that

$\forall u \in \pi : p(u) \leftarrow p(u) + \delta$ where $\delta > 0$.

That is, the potential of every unit in the coalition will increase. This is not only true instantaneously, but remains true as long as nothing external changes (we are ignoring state change, saturation, and decay). This is because $Exc|\pi$ continues to increase as the potential of the members of $\pi$ increases. Taking saturation into account adds no new problems; if all of the units in $\pi$ are saturated, the change, $\delta$, will be zero, but the coalition will remain stable.

The condition that the excitation from other coalition members alone, $Exc|\pi$, be greater than any possible inhibition INH for each unit may appear to be too strong to be useful. It is certainly true that coalitions can be stable without condition (SC) being met. The condition (SC) is useful for model building because it may be relatively easy to establish. Notice that INH is directly computable from the description of the unit; it is the largest negative weighted sum possible. If inhibition in our networks is mutual, the upper-bound possible after a fixed time $\tau$, INH$\tau$, will depend on the current value of potential in each unit u. The simplest case of this is when two units are "deadly rivals"—each gets all its inhibition from the other. In such cases, it may well be feasible to show that after some time $\tau$, the stable coalition condition will hold (in the absence of decay, fatigue, and changes external to the network). Often, it will be enough to show that the coalition has a stable "frontier," the set of units with outputs to some system under investigation.

There are a number of interesting properties of the stable coalition principle. First notice that it does not prohibit multiple stable coalitions nor single coalitions which contain units which mutually inhibit one another (although excessive mutual inhibition is precluded). If the units in the coalition had non-zero decay, the coalition excitation $Exc|\pi$ would have to exceed both INH and decay for the coalition to be stable. We suppose that a stable coalition yields control when its input elements change (fatigue and explicit resets are also feasible). To model coalitions with changeable inputs, we add boundary elements, which also had external "Input" and thus whose condition for being part of a stable coalition, $\pi$, would be:

$Exc|\pi + Input > INH$.

This kind of unit could disrupt the coalition if its Input went too low. The mathematical analysis of CM networks and stable coalitions continues to be a problem of interest. We have achieved some understanding of special cases (Feldman & Ballard, 1982) and these results have been useful in designing CM too complex to analyze in closed form.

### 4. Conserving Connections

It is currently estimated that there are about $10^{11}$ neurons and $10^{15}$ connections in the human brain and that each neuron receives intput from about $10^3$–$10^4$ other neurons. These numbers are quite large, but not so large as to present no problems for connectionist theories. It is also important to remember that neurons are not switching devices; the same signal is propagated along all of the outgoing branches. For example, suppose some model called for a separate, dedicated path between all possible pairs of units in two layers in size N. It is easy to show that this requires $N^2$ intermediate sites. This means, for example, that there are not enough neurons in the brain to provide such a cross-bar switch for substructures of a million elements each. Similarly, there are not enough neurons to provide one to represent each complex object at every position, orientation, and scale of visual space. Although the development of connectionist models is in its perinatal period, we have been able to accumulate a number of ideas on how some of the required computations can be carried out without excessive resource requirements. Five of the most important of these are described below: (1) functional decomposition; (2) limited precision computation; (3) coarse and coarse-fine coding; (4) tuning; and (5) spatial coherence.

#### Functional Decomposition

When the number of variables in the function becomes large, the fan-in or number of input connections could become unrealistically large. For example, with the function $t = f(u, v, w, x, y, z)$ implemented with 100 values of t, when each of its arguments can have 100 distinct values, would require an average number of inputs per unit of $10^{12}/10^2$, or $10^{10}$. However, there are simple ways of trading units for connections. One is to replicate the number of units with each value. This is a good solution when the inputs can be partitioned in some natural way as in the vision examples in the next section. A more powerful technique is to use intermediate units when the computation can be decomposed in some way. For example, if $f(u, v, w, x, y, z) = g(u, v) \circ h(w, x, y, z)$, where o is some composition, then separate networks of value units for $f(g, h)$, $g(u, v)$, and $h(w, x, y, z)$ can be used. The outputs from the g and h units can be combined in conjunctive connections according to the composition operator $\circ$ in a third network representing f. An example is the case of word recognition. Letter-feature units would have to

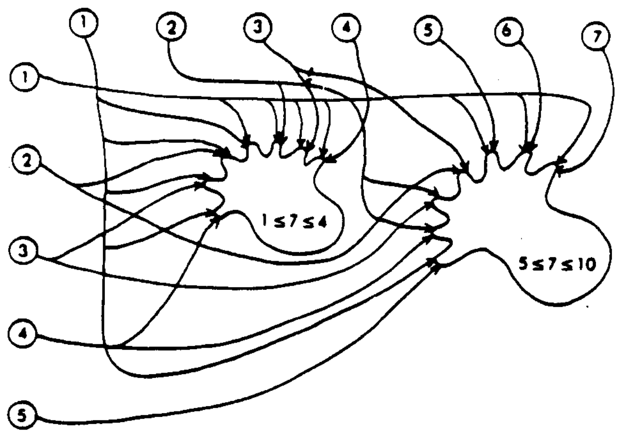


**Figure 14** Modified Multiplication Table using Less Units.

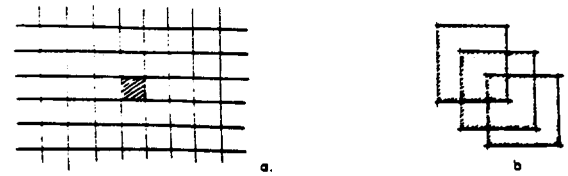


**Figure 15a** Coarse coding example. In a two-dimensional measurement space, the presence of a measurement can be encoded by making a single unit in the fine resolution space have a high confidence value. The same measurement can be encoded by making overlapping coarse units in three distinct coarse arrays have high confidence values.

connect to vastly more word units without the imposition of the intermediate level of letter units. The letter units limit the ways letter-feature units can appear in a word.

### Limited Precision Computation

In the multiplication example $z = xy$, the number of z units required is proportional to $N_xN_y$ even when redundant value units are eliminated, and in general the number of units could grow exponentially with the number of arguments. However, there are several refinements which can drastically reduce the number of required units. One way to do this is to fix the number of units at the *precision* required for the computation. Figure 14 shows the network of Figure 7 modified when less computational accuracy is required.

This is the same principle that is incorporated in integer calculations in a sequential computer: computations are rounded to within the machine's accuracy. Accuracy is related to the number of bits and the number representation. The main difference is that since the sequential computer is general purpose, the number representations are conservative, involving large number of bits. The neural units need only represent sufficient accuracy for the problem at hand. This will generally vary from network to network, and may involve very inhomogeneous, special purpose number representations.

### Coarse and Coarse-Fine Coding

Coarse coding is a general technical device for reducing the number of units needed to represent a range of values with some fixed precision, due to Hinton (1980). As Figure 15a suggests, one can represent a more precise value as the simultaneous activation of several (here 3) overlapping coarse-valued units. In general, D simultaneous activations of coarse cells of diameter D precise units suffice. For a parameter space of dimension k, a range of F values can be captured by only $F^k/D^{k-1}$ units rather than $F^k$ in the naive method. The coarse coding trick and the related coarse-fine trick to be described next both depend on the input at any given time being sparse relative to the set of all values expressible by the network.

The coarse-fine coding technique is useful when the space of values to be represented has a natural structure which can be exploited. Suppose a set of units represents a vector parameter v which can be thought of as partitioned into two components (r, s). Suppose further that the number of units required to represent the subspace r is $N_r$ and that required to represent s is $N_s$. Then the number of units required to represent v is $N_rN_s$. It is easy to construct examples in vision where the product $N_rN_s$ is too close to the upper bound of $10^{11}$ units to be realistic. Consider the case of trihedral (v) vertices, an important visual cue. Three angles and two position coordinates are necessary to uniquely define every possible trihedral vertex. (Two angles define the types of vertex (arrow, y-joint); the third specifies the rotation of the joint in space.) If we use 5 degree angle sensitivity and $10^5$ spatial sample points, the number of units is given by $N_r \approx 3.6 \times 10^5$ and $N_s = 10^5$ so that $N_rN_s \approx 3.6 \times 10^{10}$. How can we achieve the required representation accuracy with less units?

In many instances, one can take advantage of the fact that the *actual occurrence* of parameters is sparse. In terms of trihedral vertices, one assumes that in an image, such vertices will rarely occur in tight spatial clusters. (If they do, they cannot be resolved as individuals simultaneously.) Given that simultaneous proximal values of parameters are unlikely, they can be represented accurately for other computations, without excessive cost.

The solution is to decompose the space v into two

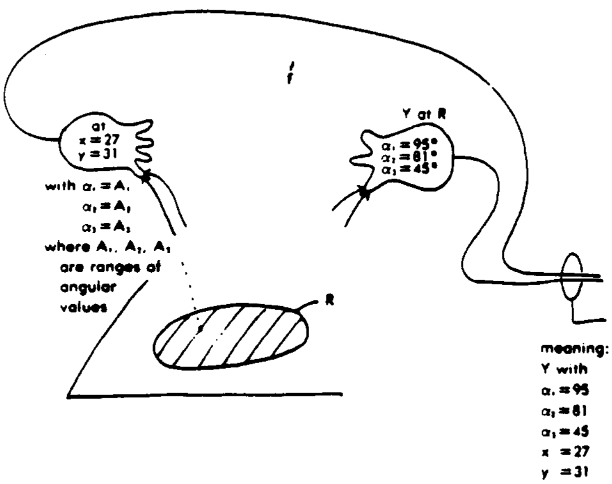



**Figure 15b** Coarse angle—fine position and coarse position—fine angle units combine to yield precise values of all five parameters.

subspaces, r and s, each with unilaterally reduced resolution.

Instead of $N_r N_s$ units, we represent v with two spaces, one with $N_{r'} N_s$ units where $N_{r'} \ll N_r$ and another with $N_r N_{s'}$ units where $N_{s'} \ll N_s$.

To illustrate this technique with the example of trihedral vertices we choose

$N_{s'} = 0.01N_s$ and $N_{r'} = 0.01N_r$.

Thus the dimensions of the two sets of units are:

$N_{s'} N_r = 3.6 \times 10^8$

and

$N_s N_{r'} = 3.6 \times 10^8$.

The choices result in one set of units which accurately represent the angle measurements and fire for a specific trihedral vertex anywhere in a fairly broad visual region, and another set of units which fire only if a general trihedral vertex is present at the precise position. The coarse-fine technique can be viewed as replacing the square coarse-valued covering in Figure 15a with rectangular (multi-dimensional) coverings, like those shown in Figure 16. In terms of our value units, the coarse-fine representation of trihedral vertices is shown in Figure 15b.

If the trihedral angle enters into another relation, say R(v, x), where both its angle and position are required accurately, one conjunctively connects pairs of appropriate units from each of the reduced resolution spaces to appropriate R-units. The conjunctive connection represents the *intersection* of each of its components' *fields*. Essentially the same mechanism will suffice for

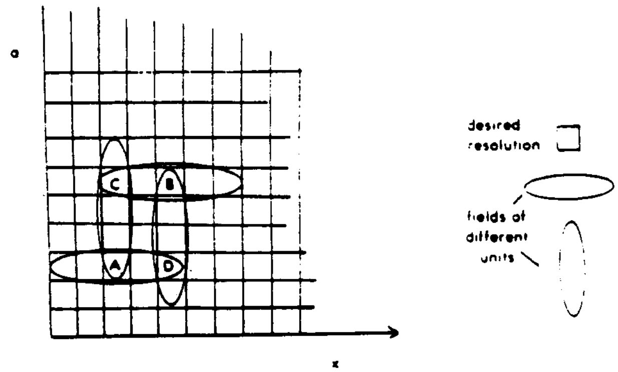



**Figure 16** Inputs at A & B cause ghosts at C & D.

conjoining (e.g.) accurate color with coarse velocity information.

An important limitation of these techniques, however, is that the input must be sparse. If inputs are too closely spaced, "ghost" firings will occur. In Figure 16, two sets of overlapping fields are shown, each with unilaterally reduced resolution. Actual input at points A and B will produce an erroneous indication of an input at C, in addition to the correct signals. The sparseness requirement has been shown to be satisfied in a number of experiments with visual data (Ballard & Kimball, 1981a, 1981b; Ballard & Sabbah, 1981).

The resolution device involves a units/connections tradeoff, but in general, the tradeoff is attractive. To see this, consider a unit that receives input from a network representing a vector parameter v. If n is the number of places where the output is used, and conjunctive connections are used to conjoin the D firing units, then Dn synapses are required. Thus if A is the number of non-coarse coded units to achieve a given acuity, then coarse coding is attractive when $A/D^{k-1} > Dn$, assuming connections and units are equally scarce. This result is optimistic in that, when other uses of conjunctive connections are taken into account, the number of conjunctive units could be unrealistically large.

## Tuning

The idea of tuning further exploits networks composed of coarsely- and finely-grained units. Suppose there are n fine resolution units of a feature A and n fine resolutions for a feature B. To have explicit units for feature values AB, $n^2$ units would be required. This is an untenable solution for large feature spaces (the number of units grows exponentially with the number of features), so alternatives must be sought. One solution to this problem is to vary the grain of the AB units so that they are only coarsely represented. This solution has its attendant disadvantages in that separate stimuli within the limits of the coarse resolution grain cannot be distinguished. Also, a set of weak stimuli can be misinterpreted. A better solution is to have a coarse unit that would respond only to a single saturated unit within its input range. In that way a collection of weak inputs is not misinterpreted.

This situation can be achieved by having the units in each finely-tuned network that are in the field of a coarse unit laterally inhibit each other, e.g., in the WTA network of Figure 5a. The outputs of these individual feature units then form disjunctive connections with appropriate coarse resolution multiple feature units. If m is the grain of the coarse resolution units along with each feature dimension, the number of disjunctions per coarse unit is $(n/m)^2$. The result of this connection strategy is that a coarse unit responds with a strength that varies as the strengths of the largest maximum in the subnetwork of each of the finely-tuned units that correspond to its field. The response of a coarse-tuned unit is the maximum of the sums of the conjunctive inputs from the finely tuned units which connect to it. In terms of Figure 15, a tuned coarse-angle cell would respond only to one high-confidence pair of angles in its range, and not to several weak ones (which couldn't correctly appear all at one position). This is a better property than just having unstructured coarse units and it will be exploited in the next section, when we deal with perceiving complex objects.

**Spatial Coherence**

The most serious problem which requires conserving connections is the representation of complex concepts. The obvious way of representing concepts (sets of properties) is to dedicate a separate unit to each conjunction of features. In fact, it first appears that one would need a separate unit for each combination at each location in the visual field. We will present here a simple way around the problem of separate units for each location and deal with the more general problem in the next section.

The basic problem can be readily seen in the example of Figure 17. Suppose there were one unit each for finally recognizing concepts like colored circles and squares. Now consider the case when a red circle (at $x = 7$) and a blue square (at $x = 11$) simultaneously appear in the visual field. If the various "colored figure" units simply summed their inputs, the incorrect "blue circle" unit would see two active inputs, just like the correct "red circle" and "blue square" units. This problem is known as cross-talk, and is always a potential hazard in CM networks. The solution presented in Figure 17 is quite general. Each unit is assumed to have a separate conjunctive connection site for each

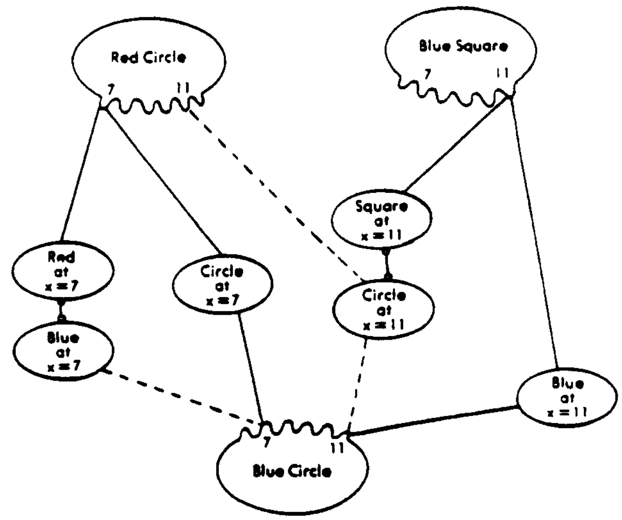



**Figure 17** Spatial coherence on inputs can represent complex concepts without cross-talk. Solid lines show active inputs and dashed lines (some of the) inactive inputs.

position of the visual field. In our example, the correct units get dual inputs to a single site (and are activated) while the partially matched units receive separated inputs and are not activated. Only sets of properties which are spatially coherent can serve to activate concept units. This example was meant to show how spatial coherence could be used with conjunctive connections to eliminate cross-talk. There are a number of additional ways of using spatial coherence, each of which involves different tradeoffs. These are discussed in the next section, which considers some sample applications in more detail.

## 5. Applications

This section illustrates the power of the CM paradigm via two groups of examples. The first shows how the various techniques for conserving connections can be used in an idealized form of perception of a complex object. Here the point is that an object has multiple features which are computed in parallel via the transform methodology. The second group of examples starts with a relatively simple problem, that of vergence eye movements, to illustrate motor control using value units. In this example, control is immediate; a visual signal produces an instantaneous output (within the settling time constants of the units). Extensions of this idea use space as a buffer for time. For motor output, space allows the incoporation of more complex motor commands. For speech input, spatial buffering allows for phoneme recognition based on *subsequent* information.

These examples were chosen to show that CM can provide a unified representation for both perception and motor control. This is important since an animal is hardly ever passively responding to its environment. Instead, it seems involved in what Arbib has called a perception-action cycle (Arbib, 1979). Perceptions result in actions which in turn cause new perceptions, and so on. Massive parallelism changes the way the perception-action cycle is viewed. In the traditional view, one would convert the input to a language which uses variables, and then use these variables to direct motor commands. CM suggests that we think of accomplishing the same actions via a transformation: sensory input is transformed (connected to) to abstract representational units, which in turn are transformed (connected to) to motor units. This will obviously work for reflex actions. The examples are intended to suggest how more flexible command and control structures can also be represented by systems of value units.

**Object Recognition**

The examples of Figures 1 and 6 are representative of the problem of gestalt perception: that of seeing parts of an image as a single percept (object). An "object" is indicated by the "simultaneous" appearance of a number of 'visual features" in the correct relative spatial positions. In any realistic case, this will involve a variety of features at several different levels of abstraction and complex interaction among them. A comprehensive model of this process would be a prototype theory of visual perception and is well beyond the scope of this paper. What we will do here is consider the prerequisite task of constructing CM solutions to the problems of detecting non-punctate visual features and of forming sets of the features which could help characterize a percept. We will refer throughout to the prototype problem of detecting Fred's frisbee, which is known to be round, baby-blue, and moving fairly fast. The development suppresses many important issues such as hierarchical descriptions, perspective, occlusion, and the integration of separate fixations, not to mention learning. A brief discussion of how these might be tackled follows the technical material.

The first problem is to develop a general CM technique for detecting features and properties of images, given that these features are not usually detectable at a single point in some retinotopic map. The basic idea is to find parameters which characterize the feature in question and connect each retinotopic detector to the parameter values consistent with its detectand.

Consider the problem of detecting lines in an image from short edge segments. Different lines can be represented by units having different discrete parameter values, e.g. in the line equation $p = x \cos\theta + y \sin\theta$, parameters are $p$ and $\theta$. Thus edge units at (x, y, x) could be connected to appropriate line units. Note that this example is analogous to the word recognition example (Fig. 1). Edges are analogous to letters and lines to words. As in the words-letter example, "top-down" connections allow the existence of a line to raise the confidence of a local edge. In our line detection example, lines in the image are high potential (confidence) units in a slope-intercept $(\theta, p)$ parameter space. High confidence edge units produce high confidence line units by virtue of the newtork connectivity. This general way of describing this relationship between parts of an image (e.g., edges) and the associated parameters (e.g., $p$, $\theta$ for a line) is a connectionist interpretation of the *Hough transform* (Duda & Hart, 1972). Since each parameter value is determined by a large number of inputs, the method is inherently noise-resistant and was invented for this purpose. A Hough transform network for circles (like Fred's frisbee) would involve one parameter for size plus two for spatial location, and exactly this method has been used for tumor detection in chest radiographs (Kimme et al., 1975). Notice that the circle parameter space is itself retinotopic in that the centers of circles have specified locations; this will be important in registering mult features.

The Hough transform is a formalism for specifying excitatory links between units. The general requirements are that part of an image representation can be represented by a parameter vector **a** in an image space A and a feature can be represented by a vector **b** which is an element of a feature space B. *Physical constraints* $f(\mathbf{a}, \mathbf{b}) = 0$ relate **a** and **b**. The space A represents spatially indexed units, and each individual element $\mathbf{a}_k$ is only consistent with certain elements in the space B, owing to the constraint imposed by the relation f. Thus for each $\mathbf{a}_k$ it is impossible to compute the set

$$\mathbf{B}_k = \{\mathbf{b} \mid \mathbf{a}_k \text{ and } f(\mathbf{a}_k, \mathbf{b}) \le \delta_b\}$$

where $\mathbf{B}_k$ is the set of units in the feature space network B that the $\mathbf{a}_k$ unit must connect to, and the constant $\delta_b$ is related to the quantization in the space B. Let H(**b**) be the number of active connections the value unit **b** receives from units in A. H(**b**) is the number of image measurements which are consistent with the parameter value **b**. The potential of units in B is given by $p(\mathbf{b}) \leftarrow H(\mathbf{b})/\Sigma_b H(\mathbf{b})$. The value p(**b**) can stand for the confidence that segment with feature value **b** is present in the image. If the measurement represented by **a** is realized as groups of units, e.g., $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2)$, then conjunctive connections are required to implemen constraint relation.

Implementing these networks often results in a set of *very sparsely distributed* high-confidence feature space units. In implementations of the line detection example, only approximately 1% of the units have maximum confidence values. This figure is also typical of other modalities. In general, each $\mathbf{a}_k$ and the relationship f will not determine a single unit in $\mathbf{B}_k$ as in the line detection example, but there still will be isolated high-confidence units. Figure 1 shows why this is the case: different $\mathbf{a}_k$ letter-feature units connect to common units in the letter space B.

We have found that parameter spaces combine with the growing body of knowledge on specific physical constraints to provide a powerful and robust model for the simultaneous computation of invariant object properties such as reflectance, curvature, and relative motion (Ballard, 1981).

Of course segmentation must involve ways of associating peaks in several different feature spaces and methods for doing this are discussed presently, but the cornerstone of the techniques are high-confidence units in the individual-modality feature spaces. In extending the single feature case to multiple features, the most serious problem is the immense size of the cross product of the spatial dimensions with those of interesting features such as color, velocity, and texture. Thus to explain how image-like input such as color and optical flow are related to abstract objects such as "a blue, fast-moving thing," it becomes necessary to use all the techniques of the previous sections.

Even if we assume that there is a special unit for recognizing images of Fred's frisbee, it cannot be the case that there is a separate one of these units for each point in the visual field. One weak solution to this kind of problem was given in Figure 17 of the last section. There could conceivably be a separate 3-way conjunctive connection on the Fred's frisbee unit for each position in space. Activation of one conjunct would require the simultaneous activation of circle, baby-blue, and fairly-fast in the same part of the visual field. The solution style with separate conjunctions for every point in space becomes increasingly implausible as we consider more complex objects with hierarchical and multiple descriptions. The spatially registered conjunctions would have to be preserved throughout the structure.

The problem of going from a set of descriptors (features) to the object which is the best match to the set is known in artificial intelligence as the *indexing problem*. The feature set is viewed as an index (as in a data base). There have been several proposed parallel hierarchical network solutions to the indexing problem (Fahlman, 1979; Hillis, 1981) and these can be

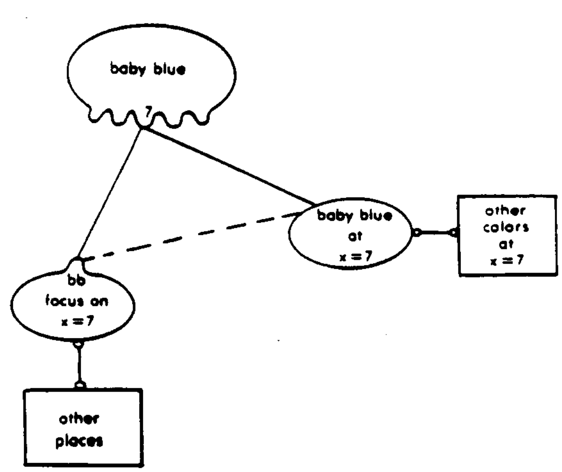



**Figure 18** Spatial focus unit can gate only input from attended positions.

mapped into CM terms. But these designs assume that the network is presented with sets of descriptors which are already partitioned; precisely the vision problem we are trying to solve. There are three additional mechanisms that seem to be necessary, two of which have already been discussed. Coarse coding and tuning (as discussed in Section 4) make it much less costly to represent conjunctions. In addition, some general concepts (e.g., blue frisbee) might be indexed more efficiently through less precise units. The new idea is an extension of spatial coherence that exploits the fact that the networks respond to activity that occurs together in time. If there were a way to focus the activity of the network on one area at a time, only properties detected in that area would compete to index objects.

The obvious way to focus attention on one area of the visual field is with eye movements, but there is evidence that focus can also be done within a fixation. The general idea of internal spatial focus is shown in Figure 18. In this network, the general "baby-blue" unit is configured to have separate conjunctive inputs for each point in space, like the blue-square units of Figure 17. The difference is that the second input to the conjunction comes from a "focus" unit, and this makes a much more general network. The idea of making a unit (e.g., baby blue) more responsive to inputs from a given spatial position can be implemented in different ways. The conjunctive connection at the x = 7 lobe of the baby-blue unit is the most direct way. But treating this conjunct as a strict AND would mean that all spatial units would have to be active when there was no focus. An alternative would be to have the "focus on 7" unit boost the output of the "baby blue at 7" unit (and all of its rivals) as shown by the dashed line; this would eliminate the need for separate spatial conjunctions on the baby-blue unit, but would alter the potential of all the units at the position being attended. The trade-offs become even trickier when goal-directed input is taken into account, but both methods have the same effect on indexing. If the system has its attention directed only to x = 7, then the only feature units activated at all will be those whose local representatives are dominant (in their WTA) at x = 7. In such a case, there would be a time when the only concept units active in the entire network would be those for x = 7. This does not "solve" the problem of identifying objects in a visual scene, but it does suggest that sequentially focusing attention on separate places can help significantly. There is considerable reason to suppose (Posner, 1978; Triesman, 1980) that people do this even in tasks without eye movement.

There are other ways of looking at the network of Figure 18. Suppose the system had reason to focus on some particular property (e.g., baby-blue). If we make bi-directional the links from "focus on x = 7" to "baby-blue" and "baby-blue at 7," a nice possibility arises. The "focus on 7" unit could have a conjunctive connection for each separate property at its position. If, for example, baby-blue was chosen for focus and was the dominant color at x = 7, then the "focus on x = 7" unit would dominate its rivals. This suggests another way in which the recognition of complex objects could be helped by spatial focus. Figure 19 depicts the fairly general situation.

In Figure 19, the units representing baby-blue, circular, and fairly-fast are assumed to be for the entire visual field and moderately precise. The dotted arrows to the "Fred's frisbee" node suggest that there might be more levels of description in a realistic system. The spatial focus links involving baby-blue are the same as in Figure 18, and are replicated for the other two properties. Notice that the position-specific sensing units do not have their potentials affected by spatial focus units, so that the sensed data can remain intact. The network of Figure 19 can be used in several ways.

If attention has been focused on x = 7 for any reason, the various space-independent units whose representatives are most active at x = 7 will become most active, presumably leading to the activation (recognition) of Fred's frisbee. If a top-down goal of looking for Fred's frisbee (or even just something baby-blue) is active, then the "focus on x = 7" will tend to defeat its WTA rivals, leading to the same result. A third possibility is a little more complicated, but quite powerful. Suppose that a given image, even in context, activates too many property units so that no objects are effectively indexed. One strategy would be to systematically scan each area of the visual field, eliminating confounding activity from other areas. But it is also possible to be more efficient. If some property unit (say baby-blue) were strongly activated, the network could focus attention on all the positions with that property. In this case it is like putting a baby-blue filter in front of the scene, and should often lead to better convergence in the networks for shape, speed, etc.

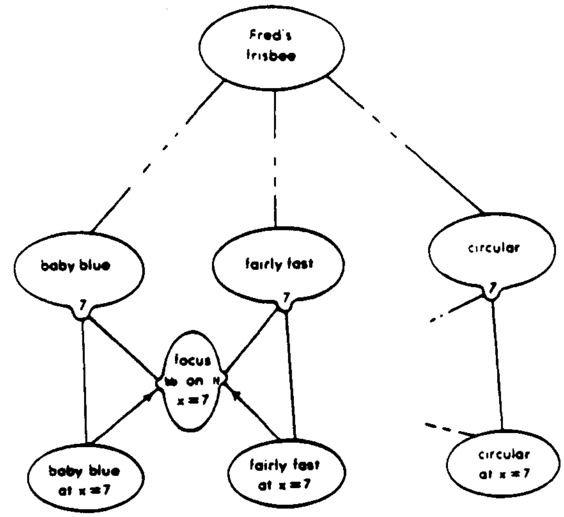



**Figure 19** Spatial focus and indexing.

One should compare the network of Figure 17 with Figures 18 and 19. In the former, parallel co-existing concepts are possible if we assume delicate arrangements of conjunctive connections. The latter networks are more robust but use sequentiality to eliminate cross-talk.

**Time and Sequence**

Connectionist models do not initially appear to be well-suited to representing changes with time. The network for computing some function can be made quite fast, but it will be fixed in functionality. There are two quite different aspects of time variability of connectionist structures. One is time-varying responses, i.e., long-term modification of the networks (through changing weights) and short-term changes in the behavior of a fixed network with time. The second aspect is sequence: the problem of analyzing inherently sequential output (such as speech) or producing inherently sequential output (such as motor commands) with parallel models. The problem of change will be deferred to (Feldman, 1981b). The problem of sequence is discussed here.

There are a number of biologically suggested mechanisms for changing the weight ($w_i$) of synaptic

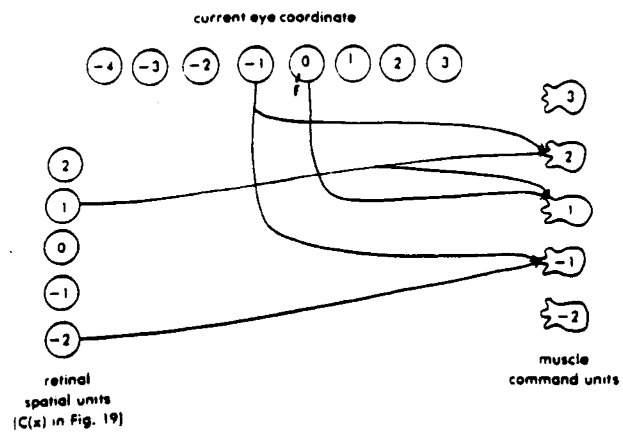



**Figure 20** Distributed Control of Eye Fixations.

connections, but none of them are nearly rapid enough to account for our ability to hear, read, or speak. The ability to perceive a time-varying signal like speech or to integrate the images from successive fixations must be achieved (according to our dogma) by some dynamic (electrical) activity in the networks. As usual, we will present computational solutions to the problems of sequence that appear to be consistent with known structural and performance constraints. These are, again, too crude to be taken literally but do suggest that connectionist models can describe the phenomena.

**Motor Control of the Eye** To see how the transform notion of distributed units might work for motor control, we present a simplistic model of vergence eye movements. (The same idea may be valid for fixations, but control probably takes place at higher levels of abstraction.) In this model retinotopic (spatial) units are connected directly to muscle control units. Each retinotopic unit can if saturated cause the appropriate contraction so that the new eye position is centered on that unit. When several retinotopic units saturate, each enables a muscle control unit independently and the muscle itself contracts an average amount.

Figure 20 shows the idea for a one-dimensional retina. For example, with units at positions 2, 4, 5, and 6 saturated, the net result is that the muscle is centered at 17/4 or 4.25. (This idea can be extended to the case where the retinotopic units have overlapping fields.) This kind of organization could be extended to more complex movement models such as that of the organization of the superior colliculus in the monkey (Wurtz & Albano, 1980).

Notice that each retinotopic unit is capable of enabling different muscle control units. The appropriate one is determined by the enabled x-origin unit which inhibits commands to the inappropriate control units via modifiers.

One problem with this simple network arises when disparate groups of retinotopic units are saturated. The present configuration can send the eye to an average position if the features are truly identical. The newtork can be modified with additional connections so that only a single connected component of saturated units is enabled by using additional object primitives. A version of this WTA motor control idea has already been used in a computer model of the frog tectum (Didday, 1976).

There are still many details to be worked out before this could be considered a realistic model of vergence control, but it does illustrate the basic idea: local spatially separate sensors have *distinct, active* connections which could be averaged at the muscle for fine motor control or be fed to some intermediate network for the control of more complex behaviors.

**Converting Space to Time** Consider the problem of controlling a simple physical motion, such as throwing a ball. It is not hard to imagine that in a skilled motor performance unit-groups fire each other in a fixed succession, leading to the motor sequence. The computational problem is that there is a unique set of effector units (say at the spinal level) that must receive input from each group at the rignt time. Figure 21a depicts a simple case in which there are two effector units ($e_1, e_2$) that must be activated alternatively. The circles marked 1–4 represent units (or groups of units) which activate their successor and inhibit their predecessor (cf. Delcomyn, 1980). The main point is that a succession of outputs to a single effector set can be modeled as a sequence of time-exclusive groups representing instantaneous coordinate signals. Moving from one time step to the next could be controlled by pure timing for ballistic movements, or by a proprioceptive feedback signal. There is, of course, an enormous amount more than this to motor control, and realistic models would have to model force control, ballistic movements, gravity compensation, etc.

The second part of Figure 21 depicts a somewhat fanciful notion of how a variety of output sequences could share a collection of lower level response units. The network shown has a single "Dixie" unit which can start a sequence and which joins in conjunctive connections with each note to specify its successor. At each time step, a WTA network decides what note gets sounded. One can imagine adding the rhythm network and transposition networks to other keys and to other modalities of output.

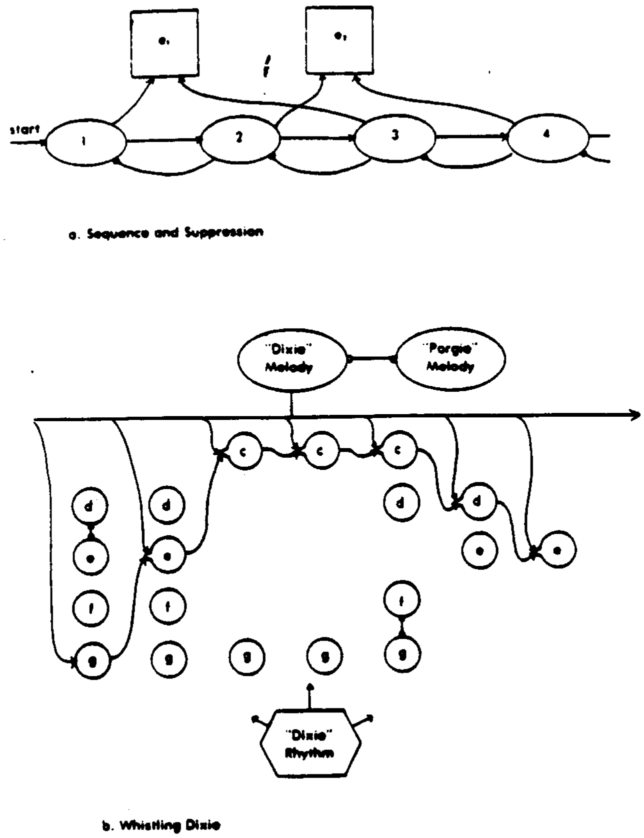



**Figure 21** Mapping Space to Time.

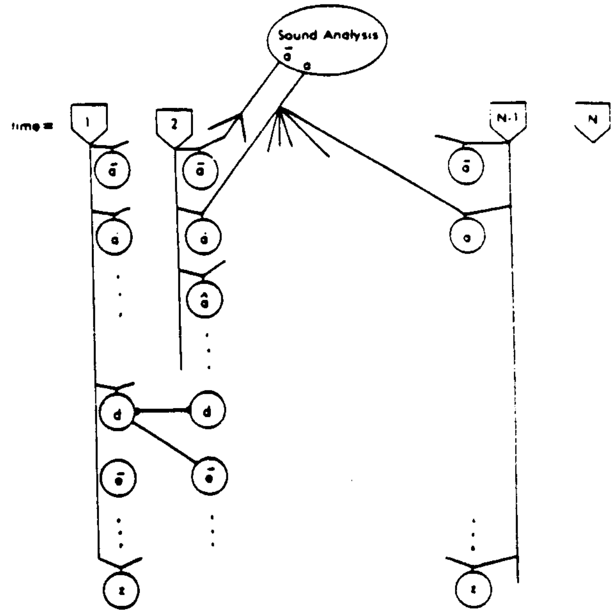



**Figure 22** Mapping Time to Space.

**Converting Time to Space** The sequencer model for skilled movements was greatly simplified by the assumption that the sequence of activities was pre-wired. How could one (still crudely, of course) model a situation like speech perception where there is a largely unpredictable time-varying computation to be carried out? One solution is to combine the sequencer model of Figure 21 with a simple vision-like scheme. We assume that speech is recognized by being sequenced into a buffer of about the length of a phrase and then is relaxed against context in the way described above for vision. For simplicity, assume that there are two identical buffers, each having a pervasive modifier ($m_j$) innervation so that either one can be switched into or out of its connections. We are particularly concerned with the process of going from a sequence of potential phonetic features into an interpreted phrase. Figure 22 gives an idea of how this might happen.

Assume that there is a separate unit for each potential feature for each time step up to the length of the buffer. The network which analyzes sound is connected identically to each column, but conjunction allows only the connections to the active column to transmit values. Under ideal circumstances, at each time step exactly one feature unit would be active. A phrase would then be laid out on the buffer like an image on the "mind's eye," and the analogous kind of relaxation cones (cf. Figure 1, 6) involving morphemes, words, etc., could be brought to bear. The more realistic case where sounds are locally ambiguous presents no additional problems. We assume that, at each time step, the various competing features get varying activation. Diphone constraints could be captured by (+ or −) links to the next column as suggested by Figure 22. The result is a multiple possibility relaxation problem—again exactly like that in visual perception. The fact that each potential feature could be assigned a row of units is essential to this solution; we do not know how to make an analogous model for a sequence of sounds which cannot be clearly categorized and combined. Recall that the purpose of this example is to indicate how time-varying input could be treated in connectionist models. The problem of actually laying out detailed models for language skills is enormous and our example may or may not be useful in its current form. Some of the considerations that arise in distributed modeling of language skills are presented in Arbib and Caplan, (1979).

## Conclusions

The CM paradigm advanced in this paper has been applied successfully only to relatively low-level tasks. There is no reason, as yet, to be confident that an intermediate symbolic representation will not be required for modeling higher cognitive processes. There is, however, the beginning of a collection of efforts which can be interpreted as attempting CM approaches to higher level tasks. These include work which explicitly uses parallelism in planning (Stefik, 1981) and deduction, and work which incorporates more connectionist architectural notions of value units (Forbus, 1981) and coarse coding (Garvey, 1981).

We have now completed six years of intensive effort on the development of connectionist models and their application to the description of complex tasks. While we have only touched the surface, the results to data are very encouraging. Somewhat to our surprise, we have yet to encounter a challenge to the basic formulation. Our attempts to model in detail particular computations (Ballard & Sabbah, 1981; Sabbah, 1981) have led to a number of new insights (for us, at least) into these specific tasks. Attempts like this one to formulate and solve general computational problems in realistic connectionist terms have proven to be difficult, but less so than we would have guessed. There appear to be a number of interesting technical problems within the theory and a wide range of questions about brains and behavior which might benefit from an approach along the lines suggested in this paper.

### Appendix: Summary of Definitions and Notation

A **unit** is a computational entity comprising:

$\{q\}$—a set of *discrete states*, $< 10$

p—a continuous value in $[-10, 10]$, called *potential* (accuracy of several digits)

v—an *output value*, integers $0 \leq v \leq 9$

$i$—a vector of *inputs* $i_1, \ldots, i_n$

and functions from old to new values of these

$$p \leftarrow f(i, p, q)$$
$$q \leftarrow g(i, p, q)$$
$$v \leftarrow h(i, p, q)$$

which we assume to compute continuously. The form of the f, g, and h functions will vary, but will generally be restricted to conditionals and simple functions.

#### P-Units

For some applications, we will use a particularly simple kind of unit whose output v is proportional to its potential p (rounded) (when $p > 0$) and which has only one state. In other words

$$p \leftarrow p + \beta \Sigma w_k i_k \qquad [0 \leq w_k \leq 1]$$
$$v \leftarrow \textit{if } p > \theta \textit{ then } \text{round}(p - \theta) \textit{ else } 0 \qquad [v = 0 \ldots 9]$$

where $\beta, \theta$ are constants and $w_k$ are weights on the input values.

#### Conjunctive Connections

In terms of our formalism, this could be described in a variety of ways. One of the simplest is to define the potential in terms of the maximum, e.g.,

$$p \leftarrow p + \beta \,\text{Max}(i_1 + i_2 - \varphi, i_3 + i_4 - \varphi, i_5 + i_6 - i_7 - \varphi)$$

where $\beta$ is a scale constant as in the p-unit and $\varphi$ is a constant chosen (usually $> 10$) to suppress noise and require the presence of multiple active inputs. The minus sign associated with $i_7$ corresponds to its being an inhibitory input. The max-of-sum unit is the continuous analog of a logical OR-of-AND (disjunctive normal form) unit and we will sometimes use the latter as an approximate version of the former. The OR-of-AND unit corresponding to the above is:

$$p \leftarrow p + \alpha \,\text{OR}\,(i_1 \& i_2, i_3 \& i_4, i_5 \& i_6 \& (\text{not } i_7))$$

**Winner-take-all (WTA) networks** have the property that only the one with the highest potential (among a set of contenders) will have output above zero after some settling time.

A **coalition** will be called stable when the output of all of its members is non-decreasing.

#### Change

For our purposes, it is useful to have all the adaptability of networks be confined to changes in weights. While there is known to be some growth of new connections in adults, it does not appear to be fast or extensive enough to play a major role in learning. For technical reasons, we consider very local growth or decay of connections to be changes in existing connection patterns. Obviously, models concerned with developing systems would need a richer notion of change in connectionist networks (cf. von der Malsburg & Willshaw, 1977). We provide each unit with a memory vector $\mu$ which can be updated:

$$\mu \leftarrow c(i, p, q, x, w, \mu)$$

where $\mu$ is the intermediate-term memory vector, w is the weight vector, $i$, p, and q are as always, and x is an additional single integer intput $(0 \leq x \leq 1)$ which captures the notion of the importance and value of the current behavior. Instantaneous establishment of long-term memory imprinting would be equivalent to having $\mu = w$. The assumption is that the consolidation of long-term changes is a separate process.

We postulate that important, favorable or unfavorable, behaviors can give rise to faster learning. The rationale for this is given in (Feldman, 1980, 1981a), which also lays out informally our views on how short- and long-term learning could occur in connectionist networks. A detailed technical discussion of this material, along the lines of this paper, is presented in (Feldman, 1981b). Obviously enough, a plausible model of learning and memory is a prerequisite for any serious scientific use of connectionism.

## References

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

Arbib, M. A. *Perceptual structures and distributed motor control.* COINS (Tech. Rep. 79-11). University of Massachusetts, Computer and Information Science, and Center for Systems Neuroscience, June 1979.

Arbib, M. A., & Caplan, D. Neurolinguistics must be computational. *The Brain and Behavioral Sciences*, 1979, 2, 449–483.

Ballard, D. H. Parameter networks: Towards a theory of low-level vision. *Proceedings of the 7th IJCAI*, Vancouver, BC, August 1981.

Ballard, D. H., & Kimball, O. A. *Rigid body motion from depth and optical flow* (Tech. Rep. 70). New York: University of Rochester, Computer Science Department, in press, 1981. (a)

Ballard, D. H., & Kimball, O. A. *Shape and light source direction from shading* (Tech. Rep.). Rochester, NY: University of Rochester, Computer Science Department, in press, 1981. (b)

Ballard, D. H., & Sabbah, D. On shapes. *Proceedings of the 7th IJCAI*, Vancouver, BC, August 1981.

Collins, A. M., & Loftus, E. F. A spreading-activation theory of semantic processing. *Psychological Review*, November 1975, *82*, 407–429.

Delcomyn, F. Neural basis of rhythmic behavior in animals. *Science*, October 1980, *210*, 492–498.

Dell, G. S., & Reich, P. A. Toward a unified model of slips of the tongue. In V. A. Fromkin (Ed.), *Errors in Linguistic Performance: Slips of the Tongue, Ear, Pen, and Hand.* New York: Academic Press, 1980.

Didday, R. L. A model of visuomotor mechanisms in the frog optic tectum. *Mathematical Bioscience*, 1976, *30*, 169–180.

Duda, R. O., & Hart, P. E. Use of the Hough transform to detect lines and curves in pictures. *Communications of the ACM 15*(1), January 1972, 11–15.

Edelman, G., & Mountcastle, B. *The Mindful Brain.* Boston, MA: MIT Press, 1978.

Fahlman, S. E. *NETL, A System for Representing and Using Real Knowledge.* Boston, MA: MIT Press, 1979.

Fahlman, S. E. The Hashnet interconnection scheme. Computer Science Department, Carnegie-Mellon University, June 1980.

Feldman, J. A. *A distributed information processing model of visual memory* (Tech. Rep. 52). Rochester, NY: University of Rochester, Computer Science Department, 1980.

Feldman, J. A. A connectionist model of visual memory. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel Models of Associative Memory.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1981. (a)

Feldman, J. A. *Memory and change in connection networks* (Tech. Rep. 96). Rochester, NY: University of Rochester, Computer Science Department, October 1981. (b)

Feldman, J. A. *Four frames suffice* (Tech. Rep. 99). Rochester, NY: University of Rochester, Computer Science Department, in press, 1982.

Feldman, J. A., & Ballard, D. H. *Computing with connections* (Tech. Rep. 72). Rochester, NY: University of Rochester, Computer Science Department, 1981; to appear in book by A. Rosenfeld & J. Beck (Eds.), 1982.

Forbus, K. D. Qualitative reasoning about physical processes. *Proceedings of the 7th IJCAI*, Vancouver, BC, August 1981, 326–330.

Freuder, E. C. Synthesizing constraint expressions. *Communications of the ACM*, November 1978, *21*(11), 958–965.

Garvey, T. D., Lowrance, J. D., & Fischler, M. A. An inference technique for integrating knowledge from disparate sources. *Proceedings of the 7th IJCAI*, Vancouver, BC, August 1981, 319–325.

Grossberg, S. Biological competition: Decision rules, pattern formation, and oscillations. *Proc. National Academy of Science USA*, April 1980, 77(4), 2238–2342.

Hanson, A. R., & Riseman, E. M., (Eds.). *Computer Vision Systems.* New York: Academic Press, 1978.

Hillis, W. D. The connection machine (Computer architecture for the new wave). AI Memo 646, M.I.T., September 1981.

Hinton, G. E. Relaxation and its role in vision. (Ph.D. thesis, University of Edinburgh, December 1977.)

Hinton, G. E. Draft of Technical Report. La Jolla, CA: University of California at San Diego, 1980.

Hinton, G. E. The role of spatial working memory in shape perception. *Proceeding of the Cognitive Science Conference*, Berkeley, CA, August 1981. (a) 56–60.

Hinton, G. E., & Anderson, J. A. (Eds.). *Parallel Models of Associative Memory.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1981.

Horn, B. K. P., & Schunck, B. G. Determining Optical Flow. AI Memo 572, AI Lab, MIT, April 1980.

Hubel, D. H., & Wiesel, T. N. Brain mechanisms of vision. *Scientific American*, September 1979, 150–162.

Ja'Ja', J., & Simon, J. Parallel algorithms in graph theory: Planarity testing. CS 80-14, Computer Science Department, Pennsylvania State University, June 1980.

Jusczyk, P. W., & Klein, R. M. (Eds.). *The Nature of Thought: Essays in Honor of D. O. Hebb.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Kandel, E. R. *The Cellular Basis of Behavior.* San Francisco, CA: Freeman, 1976.

Kimme, C., Sklansky, J., & Ballard, D. Finding circles by an array of accumulators. *Communications of the ACM*, February 1975.

Kinsbourne, M., & Hicks, R. E. Functional cerebral space: A model for overflow, transfer and interference effects in human performance: A tutorial review. In J. Requin (Ed.), *Attention and Performance* 7. Hillsdale, NJ: Lawrence Erlbaum Associates, 1979.

Kosslyn, S. M. *Images and Mind.* Cambridge, MA: Harvard University Press, 1980.

Kuffler, S. W., & Nicholls, J. G. *From Neuron to Brain: A Cellular Approach to the Function of the Nervous System.* Sunderland, MA: Sinauer Associates, Inc., Publishers, 1976.

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

McClelland, J. L., & Rumelhart, D. E. An interactive activation model of the effect of context in perception: Part 1. *Psychological Review*, 1981.

Minsky, M., & Papert, S. *Perceptrons.* Cambridge, MA: The MIT Press, 1972.

Norman, D. A. A psychologist views human processing: Human errors and other phenomena suggest processing mechanisms. *Proceedings of the 7th IJCAI*, Vancouver, BC, August 1981, 1097–1101.

Perkel, D. H., & Mulloney, B. Calibrating compartmental models of neurons. *American Journal of Physiology* 1979, *235*(1), R93–R98.

Posner, M. I. *Chronometric Explorations of Mind.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1978.

Prager, J. M. Extracting and labeling boundary segments in natural scenes. *IEEE Trans. PAMI*, January 1980, *2*(1), 16–27.

Ratcliff, R. A theory of memory retrieval. *Psychological Review*, March 1978, *85*(2), 59–108.

Rosenfeld, A., Hummel, R. A., & Zucker, S. W. Scene labelling by relaxation operations *IEEE Trans. SMC 6*, 1976.

Sabbah, D. Design of a highly parallel visual recognition system. *Proceedings of the 7th IJCAI* Vancouver, BC, August, 1981.

Scientific American. *The Brain.* San Francisco, CA.: W. H. Freeman and Company, 1979.

Sejnowski, T. J. Strong covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology*, 1977, *4*(4), 303–321.

Smith, E. E., Shoben, E. J., & Rips, L. J. Structure and process in semantic memory: A featural model for semantic decisions. *Psychological Review*, 1974, *81*(3), 214–241.

Stefik, M. Planning with Constraints (MOLGEN: Part 1). *Artificial Intelligence*, *16*(2), 1981.

Stent, G. S. A physiological mechanism for Hebb's postulate of learning. *Proc. National Academy of Science USA*, April 1973, *70*(4), 997–1001.

Sunshine, C. A. Formal techniques for protocol specification and verification. *IEEE Computer*, August 1979.

Torioka, T. Pattern separability in a random neural net with inhibitory connections. *Biological Cybernetics*, 1979, *34*, 53–62.

Triesman, A. M., & Gelade, G. A feature-integration theory of attention. *Cognitive Psychology*, 1980, *12*, 97–136.

Ullman, S. Relaxation and constrained optimization by local processes. *Computer Graphics and Image Processing*, 1979, *10*, 115–125.

von der Malsburg, Ch., & Willshaw, D. J. How to label nerve cells so that they can interconnect in an ordered fashion. *Proc. National Academy of Science USA*, November 1977, *74*(11), 5176–5178.

Wickelgren, W. A. Chunking and consolidation: A theoretical synthesis of semantic networks, configuring in conditioning, S-R versus cognitive learning, normal forgetting, the amnesic syndrome, and the hippocampal arousal system. *Psychological Review*, 1979, *86*(1), 44–60.

Wurtz, R. H., & Albano, J. E. Visual-motor function of the primate superior colliculus. *Annual Review of Neuroscience*, 1980, *3*, 189–226.

Zeki, S. The representation of colours in the cerebral cortex. *Nature*, April 1980, *284*, 412–418.