PARALLEL MODELS OF ASSOCIATIVE MEMORY

Edited by

Geoffrey E. Hinton
M. R. C. Applied Psychology Unit
Cambridge, England

James A. Anderson
Brown University

LEA LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
1981 HILLSDALE, NEW JERSEY

9 Notes on a Self-Organizing Machine

Stuart Geman
Brown University

9.1. INTRODUCTION

This chapter sets forth a design for a system whose purpose is to discover temporal and spatial regularities in a high-dimensional environment. Whereas the goal of this research is the realization of an "intelligent system," the model is based on principles of organization and self-modification widely believed to be in force in the nervous system. The design is of a parallel-processing machine composed of nonlinear and highly interconnected devices. As it is presented here, the model is a general one in the sense that it is not dedicated to any particular environment or task. A specific implementation of the model requires the specification of two sets of parameters: the "input primitives" and the "direction primitives." The input primitives represent the information about the environment that is available to the system. The direction primitives define what is "good" and what is "bad" relative to the system.

It may help to orient the reader if I briefly describe a particular implementation of the model that is now being developed. In this implementation the environment is the world of numbers, leading to input primitives such as odd, even, prime, sum, exponentiate, etc. Direction primitives specify that a non sequitur, such as attempting to perform a binary operation with only one available argument, is bad, that a conjecture that is well supported (as by trial and error) is good, and so on. The system interacts with a computer, requesting numbers and the results of operations, and attempts to discover regularities among these primitives.

Whereas it is my position that the principles of organization and modification used here are in force in the nervous system, I do not propose that this implementation mimics, in any specific sense, the techniques of a mathematician. This implementation is meant as an exercise toward developing a system that can discover regularities in complicated environements. The point of using numbers is that they provide an extremely convenient world in which to experiment, but the model is in no way dedicated to this world. In fact the presentation in this chapter will rarely refer to this implementation because it is not yet complete and there are few results to report.

Humans learn, and learn to discover, regularities in the complex and high-dimensional environments of the "real world." We cannot reasonably expect to invent another solution to this inference problem, at least not one that will approach the general application of human thought. Therefore it would seem to be expedient to look to the neural and cognitive sciences for clues about the proper architecture, and it is in this spirit that the model here has been developed. However, I am *not* proposing this model as a "neural network" model. It will be clear to the reader with even a rudimentary knowledge of neurophysiology and neuroanatomy that the basic units of information processing used here have little to do with real neurons. In fact I will completely ignore the problem of realizing these units in neurallike machinery, because I do not believe that the specification of such a realization would at this time be a useful exercise. Models have been formulated at the level of neural structure for the realization of a variety of "higher-level" functions presumed to be carried out by some part of the nervous system. Yet we can seriously question whether these models have improved our understanding of human intelligence. The real problem may lie in identifying which functions to realize. We can imagine numerous architectures of nonlinear neuronlike elements, communicating through modifiable connections, for the execution of virtually any well-specified procedure of information processing. But it would be difficult to choose between these architectures based on what little is known of the physiology of higher-level function in the nervous system. The right question now may be *what* to build rather than *how* to build it.

A problem that is fundamental to our understanding of the nervous system, and one that has implications for the design of Artificial Intelligence, is the proper interpretation of "local" activity in the brain. Many authors have argued that little significance can be attached to activity at any particular location; it is the pattern of activity across a neural system that embodies the system's interpretation of a stimulus. This point of view arises mainly as a corollary of the distributed memory hypothesis, which has its roots in the classical experiments of K.S. Lashley (1950). Others have taken the point of view that local activities have a very specific and often "high-level" interpretation. An often quoted paper by H. B. Barlow (1972) argues for the existence of "grandmother cells," whose activities signal the presence of specific stimuli, such as chairs, cars, or particular individuals. The semantic net approach to Artificial Intelligence can be interpreted in this way (see Fahlman, Chapter 5, this volume) if an individual node representing a specific concept, pattern, or operation is identified with an individual hardware unit.

The present model is based on the proposition that both interpretations are, at once, in force. It may be useful to anticipate now the discussion in this regard by outlining an argument for this point of view. It is well demonstrated that the efficacy of synaptic connections can be influenced by the activities of the neurons that communicate through these connections (see discussion in Chapter 1, this volume). It is just such changes that most investigators see as the neurophysiological analog of associative learning. Let us accept the point of view that synapses are, indeed, the site of the engram and that modification of synapses can be described by some function of the presynaptic and postsynaptic activities. Then, at any stage in development, what has been learned can depend only on pairwise relations among the activities of individual neurons. Suppose that these neuronal activities are at all levels as unselective, in terms of "events" and "objects," as the activities at the most peripheral levels. It is then difficult to see how a synaptic memory, based only on pairwise associations, could contain information about highly complex and specific relations among these events and objects, as the human memory most certainly does. It would seem that activities in some neurons need to signal selective events rather than a noninformative mixture that occurs as frequently as the primitives themselves.

It is widely believed that ontological development includes a process by which some cells of the visual cortex, initially not completely specific in their activities, come to signal selective events, or "features," in the environment. Many investigators have suggested that this process continues, in a hierarchical fashion, as one moves to deeper levels of processing in the brain. The system described in this chapter utilizes just such a process to create local activities that signal selective events. Of course, care must be taken as to which events are to be represented because we cannot possibly represent all such "high-order" relationships in an environment that has any appreciable number of dimensions. The precise mechanism used is described in some detail in the sections to come. The point that I wish to make here is that, as a result of this process, a familiar event achieves both distributed and local representation. The representation is distributed at the most peripheral levels, where the event is signaled by the activities in a particular set of primitives, whereas this representation is increasingly localized as we move deeper into the system. It will be seen that the model here continues to utilize all levels of this representation.

Whereas the presentation here is about the design of a system for the organization of information in complex environments, behind this design there is a model for the nature of the information to be processed. I begin in Section 9.2 with an attempt to identify some salient features of real-world environments, and this discussion guides the development of the system as it follows in later sections. Mechanisms for the associative learning and associative recall of spatial[1] relations are described in Section 9.3. It is these mechanisms that suggest the local/

[1] "Spatial" refers here to associations among events that occur simultaneously. It is not in specific reference to visual.

global representation scheme referred to earlier. I argue that an associative process will be effective provided that both forms of representation are available. Section 9.4 then outlines the means by which the system develops this representation. It may be seen that this is dependent on the particular experience of the network. The pieces, as they have been described up to this point, are then brought together in Section 9.5. The result is what I will call a *spatial coding module*, one of the three building blocks of the system. Temporal relations are organized and learned by a very similar structure called a *temporal coding module*, which is the topic of Section 9.6. The principles of architecture and function developed for the two types of coding modules are then applied to the problem of organizing and integrating the actions of the entire system. The result, a *control module*, is the main topic of the final section, Section 9.7. Scattered through the atricle are 11 propositions. These statements are intended as informal summaries of the main assumptions on which the design is based.

I do not attempt to describe the system in full detail. Certainly this would be premature. The current implementation already suggests changes in many of the particulars. Still we do expect, and have so far been able, to maintain the essential principles of organization as they are presented in this article.

Finally, let me anticipate two likely, and largely valid, criticisms—first, that the difficult problem of extracting appropriate primitives in real environments has been completely ignored. Perhaps an understanding of what to do with these primitives at "higher" levels will point to an understanding of what constitutes good peripheral machinery. It is possible that the peripheral hardware problem will prove to be more difficult than the problem of higher-level intelligent processing. A second objection is that the idealizations are absurd and that time and features do not have discrete representations in the nervous system. The model here has its generalizations to continuous time and continuous features. The philosophy is to work with a system that can be analyzed and simulated with relative ease and hope that it will suggest the correct generalizations.

9.2. ON THE NATURE OF ENVIRONMENT

This chapter proposes a mechanism for organizing the information of real-world environments, and this mechanism reflects a particular point of view concerning the nature of such information. This section is devoted to a discussion of the assumptions that comprise this point of view. This may appear to be a circuitous route to a description of the system, but the design of this system is, in large measure, based on the formulation developed here.

It is best to start with a discussion of features because it is the purpose of the system to learn relations among features. Actually a precise definition evolves from the description of the design, but for now we can think of a feature in pretty much the conventional "pattern recognition" sense: Features describe the status

(present or absent, present in what form or to what degree, etc.) of specific events in the system's environment. Mostly, I refer to "high-level" features. These may, for example, signal specific words, objects in a visual scene, or in medicine the outcome of a diagnostic test. Of course something must be said about the development of high-level features from more primitive features, but a discussion of this aspect of the model is better motivated later. The notion of a feature also includes representations of "actions," analogous to the way in which proprioception represents motor activity as part of our sensory environment. From this point of view learning the consequences of certain actions under certain circumstances is a special case of learning relations among features, for the consequences, the actions, and the circumstances all have a common representation. A good analogy to the "feature" defined here is the "cogit" of the Hayes-Roth (1977) theory.

The values of features form a representation of the system's environment. It is natural to think of all these values as being available at each instant, and this point of view is implicit in the pattern recognition formalism and in many of the current models of memory. The design here is based on a different point of view, one which explicitly recognizes the existence of an "unobserved" state, in which the value of a feature is not available. In a cognitive sense it is clearly not the case that at each instant every feature is examined or appreciated. At any instant we are unaware of most of the complement of sensory information potentially available. Also, features may be unobserved because their values are physically not available. We recognize words with just a subset of the acoustic information that we are capable of using, as when listening over a telephone. A good example, at a higher level, is the result of a test in diagnostic medicine. If the test is not performed, then we have no value for the feature that is the test result. If the test is performed, then we observe this feature, and it may be positive or negative or possibly any of a continuum of values.

My point is that features come in two states: *observed* and *unobserved*. The value of a feature is available only when the feature is in the observed state. The unobserved state generally carries little or no information. This distinction between the observed and unobserved states of a feature is the basis for a definition of "associative recall." Roughly, associative recall in this model is a process of predicting, or "filling in," the values of certain unobserved features. These statements lead then to the first proposition:

Proposition 1. The processing of a feature distinguishes two states: observed and unobserved. In the observed state the value of the feature is available. In the unobserved state this value is not available. In itself the *state* of a feature contains little information.

As a first approximation I usually assume that the state of features carry *no* information with respect to the values of features: The states and values of features are independent.

It is obvious that memory would have no purpose if past experiences could not be taken as evidence towards a correct interpretation of future experiences. Plainly there is a relationship between past observations and the course of events in the future. I have used a model of this relationship to guide the design of the system presented here. Formally this is a Bayesian model, in which the prior distribution (which is only partly specified) is on distributions among features. To speak loosely, the prior determines what relations among features we are likely to encounter. This approach has the advantage that it easily translates assumptions about the nature of evidence into precise statements about the prior distribution. In theory this precision should in turn dictate the details of mechanisms for the processing of information by the system. In fact what I have is only a heuristic connection between the system and this Bayesian model. Therefore I replace a formal description of this model with a looser, more intuitive discussion of its main assumptions.

In this model, the "environment" is a vector-valued random function of (discrete) time.[2] The components of this vector are the features, and the prior distribution is a distribution on the probability law for this process. The most complete possible observation is of the entire vector,

$$f_1(t), \ldots, f_n(t),$$

where $f_i(t)$ is the value of the ith feature at time t. If we use "?" to indicate an unobserved feature, an actual observation looks like

$$f_1(t), f_2(t), ?, ?, ?, \ldots, f_k(t), ?, \ldots, f_n(t),$$

for example. It is assumed that the ?s contain no information about the values of the unobserved features.

There are three main assumptions about this process, and these are formulated as conditions on the prior distribution. These are the assumptions of constancy, continuity, and consistency. Constancy demands that the rules do not change: The process is stationary. If the rules appear to change, then it is because context has changed or because associations were by chance. Memory would serve no purpose in an environment that did not respect some measure of stationarity. (Actually I assume something stronger—a type of ergodicity or mixing to insure convergent behavior of certain estimators introduced later.) By continuity I mean, roughly, that similar events tend to have similar implications. Of course this "rule" is not absolute; but it *tends* to be true; and this is exactly the notion that the Bayesian formulation captures.

It should be easy to appreciate that something like constancy and continuity is in force in the real world. Indeed it is difficult to imagine a model for a learning system that would not anticipate, implicitly, these conditions. Certainly there is a good deal more regularity in real environments. We find evidence in our experience for novel circumstances, and neither constancy nor continuity can account for this. It is easier to learn the diagnosis of a new disease if we are already familiar with other diseases. Why should this be true? The world of medicine allows us to utilize what we have already learned about the relations among symptoms in the context of this new disease. Thus the disease's manifestations can be largely inferred from only a partial description of its symptomatology. Roughly, consistency is the assumption that there is an environmental analog to the perceptual process of "filling-in" and the cognitive process of "stringing together associations":

[2] The reader unfamiliar with probability theory is cautioned against interpreting "random" as meaning "unstructured." Indeed a deterministic model is an example of the more general probabilistic approach.

Proposition 2. The probabilistic relations among features are consistent: It is more likely that A will be evidence for C when A is evidence for B and B is evidence for C.

This proposition may appear so natural that it involves no assumption at all. But the world need not have this property. Given a precise formulation of Proposition 2, it is not hard to demonstrate models that violate consistency. And, as a corollary, a learning machine can fail to take advantage of this presumed regularity.

As a simple example of a world that has the properties of constancy, continuity, and consistency, consider the "circles world," constructed as follows (see Fig. 9.1). Each feature in this world is associated with a circle on the unit torus. These circles are chosen, independently and once and for all, by first randomly choosing a center from the uniform distribution of the torus, and then randomly choosing a radius from some fixed distribution. Features are binary valued, with values + and − indicating the regions inside or outside of the associated circles. The choice of which value will indicate which region is made independently for each feature by a (fair) coin flip. Hence each feature is an independently generated binary-valued function defined on the unit torus. If we now put a uniform probability distribution on this torus, then the features can be viewed as random variables on the resulting probability space. Notice that these are not, in general, independent random variables. (In Fig. 9.1, for example, $f_1 = +$ implies, deterministically, $f_3 = -$; f_1 and f_3 are certainly not independent random variables.)

The process, $(f_1(t), \ldots, f_n(t))$, $t = 1, 2, \ldots$, is generated by first choosing a sequence of independent points, one for each t, on the unit torus using the uniform probability distribution. The values of the features at a given time are then determined by the position of the corresponding point. The observed process is generated at each time, t, by flipping independently for each feature, a (possibly biased) coin, and entering "?" (unobserved) if the result is heads or the value of the feature if the result is tails.

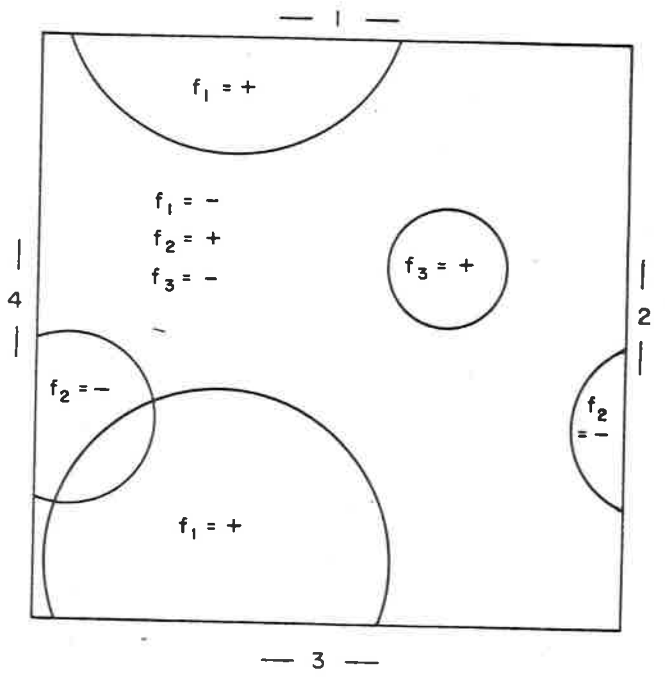

FIG. 9.1. A "circles world" with 3 features. The surface is a torus: sides 1 and 3 and sides 2 and 4 are identified.

It is clear that the condition of independence between the unobserved state and the values of features is satisfied. It is also true that the circles world satisfies the conditions of constancy, continuity, and consistency. I have found this structure to be a useful conceptual tool as well as a convenient device for creating simulated environments in which to test some of the learning and recall algorithms described in later sections.

9.3. ASSOCIATIVE LEARNING AND RECALL

Begin with an idealization: Features are to be taken as binary valued. By almost any interpretation, features in fact have many-valued, or continuous-valued, representations in the brain. But it is unlikely that the transition from "feature present" to "feature present at a particular strength or value" is fundamental. In fact, I believe that the binary idealization, because of its simplicity, can often "clear the air," and suggest the proper generalization to a continuous formalization. In any case, most of what is developed has a natural analog for continuous-valued features.

Features, for now, may be at any level of the cognitive hierarchy. They may represent phonemes, edges, words, diagnostic symptoms, or even phrases or thoughts. Section 9.4 attempts to make the connection from primitive to high-level feature. But for now, let us take features as fixed and given, and concentrate on the problem of learning and retrieving their interimplications.

Binary features can be thought of as indicating the presence or absence of some event. As proposed in the previous section the point of view here is that most of the features are most of the time unobserved; there is no direct information available on the presence or absence of the related event. Within this formalism recall has the natural interpretation of being a process by which the values of certain unobserved features are filled in (estimated). I am not suggesting that *all* unobserved features are estimated. Circumstances will define a collection of "target" features whose values are of particular importance at a particular time. In short:

Proposition 3. The purpose of recall is to estimate the values of a particular set (target set) of unobserved features.

We may think of each target feature as a classification. In the binary formalism, for each target feature, recall performs a two-way classification of the observed features, with the categories being the presence or absence of the event associated with that target feature.

A target feature in medicine might be a particular disease. The observed features correspond to observed symptoms (which may be "observed" to be present or absent) or to the results of completed tests. Estimating the values of the target features corresponds to establishing which diseases are present and which are not. Or, given the presence of a disease, certain symptoms may play the role of target features. Given ischemic heart disease, do we expect blockage of a particular coronary artery; do we expect hypertension; etc.? In vision we might think of the "label" as the natural target feature, given the image of an object (a collection of observed features). Or, with a partial observation of an object, we might think of the target features as being those unobserved features that are ordinarly associated with that object. In mathematics, concerning numbers, we may observe a set of features that define the event "odd plus odd" and wish to estimate the value of the target feature "even"—is it present or absent?

Target features are estimated using the information available: the values of the observed features. The straightforward approach is to estimate, individually, the value of each target feature using knowledge, from experinece, of the associations between the observed and target features. This procedure has no limitation if an infinite time of experience is available. The complete statistics between the observed and target features can then be known and an optimal (whatever the criterion) estimator constructed. But experience is finite, and in the most interesting cases, brief; an effective algorithm will anticipate certain structures. What sort of relations do we *expect* to find? In the language of Section 9.1, I am referring to the prior distribution, which determines what worlds we are likely to encounter and about which we have made certain assumptions. Here the relevant assumption is consistency, which suggests that a recall algorithm take advantage of, in a particular way, experience concerning certain of the unobserved features.

If cigarette smoking usually contributes to cardiovascular disease, and cardiovascular disease usually shortens life span; then, with everything else being equal, we take it for granted that cigarette smoking is likely to shorten life span. I want to emphasize that this does indeed involve an assumption; a prior distribution can be constructed so that such reasoning will prove unprofitable, or if we wish, *always fail*.

Let us suppose that we have defined a "local" mechanism for generating an opinion about the value of any particular unobserved feature given an arbitrary collection of observed features. One such mechanism will be discussed in detail presently. For now assume that it is available. Consistency suggests the following mechanism for obtaining the values of the target features: First, fill in those unobserved features about which the local algorithm has a "strong opinion," strong being defined with respect to some threshold value. If the target features are among these estimated features, then the procedure terminates. If not, then utilize the augmented set of observed features (truly observed plus filled-in features) to fill in another generation of unobserved features. Continue until either the target features are filled in or the process terminates by virtue of no further opinions having strength above threshold. In case of the latter, start again with a lower threshold. This is "associative recall," as it is defined in this model. Thus:

Proposition 4. Call a feature *decided* if it has been observed or its value has been estimated (filled in). The values of the target features are estimated by a recursive process that terminates when all target features are decided. A step in this recursion is the calculation of an *opinion* concerning the value of each undecided feature using the (observed and filled-in) values of the decided features. If, for a particular undecided feature, the opinion is above a threshold value, then this feature is filled in and becomes decided.

The relation of this model for recall to the notions of consistency and associative memory is clear. Loosely speaking, if A is associated with B and B is associated with C, then we come to associate A with C through the progression $A \rightarrow A \cap B \rightarrow A \cap B \cap C$, and furthermore the world is such that this association is usually appropriate.

(An appealing alternative way to incorporate information gained about the unobserved features in the estimation of target features is the "projection method": Experience is used to construct the projection operator onto the space spanned by that experience. The response to a stimulus is the action of this operator on that stimulus, and this action selects a combination of experience "close" to the stimulus (c.f. Kohonen, 1977). Although this recall algorithm is not based on a notion of consistency, it does use all of the information in the experience set rather than being limited to what has been learned about the relations between observed and target features. The difficulty, for our purposes, is that there is no apparent way to incorporate the unobserved state effectively. Suppose, for example, that "0" is used to indicate the unobserved state and that each feature has value 1 or 2. Then because of the distortion of patterns by the unobserved state, the span of the experience set will approach the *entire* space. For example, every feature may eventually be observed in isolation of all other features, at which time the experience set spans the entire feature space, and the projection operation returns the stimulus unaltered. Kohonen calls an experience set a set of *samples* and a stimulus a *key*. The point of view taken in this article is that the samples and the keys are of the exact same nature.)

I turn now to a discussion of the local algorithm, the appropriateness of which determines the accuracy of the proposed recall mechanism. The purpose of the local algorithm is to compute an opinion of the true value of an unobserved feature, given the values of a collection of observed features. The "strength" of this opinion should reflect the strength of the evidence available for this opinion.

Let us consider the nature of the information available in the collection of observed features. For this purpose it is useful to make a distinction between what I call *implicit* and *explicit* information. Consider a Venn diagram in which the regions are defined by particular features taking particular values. A point in this diagram can be thought of as a particular stimulus. (For the circles world, introduced in Section 9.1, the unit torus can serve as the Venn diagram when the circles associated with the collection of features have been drawn in.) Suppose that the observed features are f_i and $f_{i'}$, and that these observations define, respectively, the regions A and B in the Venn diagram. These observations contain, *implicitly*, the information that the stimulus is within A ∩ B. However unless there is a third observed feature, $f_{i''}$, such that a particular value of this feature indicates, precisely, the region A ∩ B, this information is not *explicitly* available (unless, of course, it should happen that A ⊂ B or B ⊂ A). The point is this: It is a practical limitation that the great majority of such information can only be available implicitly for a feature set of any appreciable size. If there are only 100 features in the circles world, then there are potentially 3^{100} ($> 10^{47}$) such regions, and even the human brain could not possibly have available an explicit representation for every such region. (This widely appreciated limitation is the focal point for the discussion in the next section.)

In the example we are given an observation of f_i and $f_{i'}$, and we wish to estimate the value of some unobserved feature, say f_j. The message from the previous paragraph is that we cannot reason that the stimulus is in A ∩ B and then ask for the most likely value of f_j because such information will in general not be explicitly available. We must somehow combine the *separate* information contained in the statements "stimulus in A" and "stimulus in B." In other words the problem is one of properly combining the information contained individually in each of the observed features (such as f_i) concerning the value of a particular unobserved feature (such as f_j). Presumably this information is gained from joint observations of features f_i and f_j in the past; that is, evidence concerning the relation between features f_i and f_j is gained each time these features are simultaneously observed, and probably not otherwise. "Probably not otherwise" in part reflects the assumption that there is no information in the fact of the unobserved states and in part reflects the assumption that a filled-in feature is not treated as an observed feature with respect to learning. (The value filled in for an unobserved feature is based entirely on the experience of the system and, as such, contains no new information concerning the relation between f_i and f_j.)

The local algorithm chooses a value for f_j given observations, say, of f_i and $f_{i'}$. On what basis can a "rational" choice be made? It should be emphasized that conventional statistical approaches have very little to offer in the present context. Either a Bayesian or a maximum likelihood estimator would require a knowledge of the joint conditional distribution of f_i and $f_{i'}$, given f_j. But as I have argued earlier, we must for practical reasons assume that higher-order information of this type is not explicitly available. This joint condition distribution could be reduced to a product of individual conditional probabilities with an assumption of conditional independence, but there is no reason for believing that this is even approximately true. (Certainly it is not true, for example, in the circles world.) It should also be recognized that even a knowledge of individual conditional probabilities cannot be assumed because we are interested here in estimation based on finite (and typically "small") samples. On the other hand, the *optimal* (minimum expected error rate) local decision function (based on pairwise observations) could be implemented if there were available a completely specified prior distribution for the Bayesian model developed in the previous section. This would in fact eliminate the motivation for a recursive scheme, the optimal decision being made by a direct application of the local algorithm to the target features. But it would be difficult indeed to argue for any such completely specified prior distribution.

The strategy that I take in developing a local algorithm is to assume first (temporarily) a complete knowledge of all pairwise statistics among features. Even then a "best" decision function is not defined—again because of an incomplete specification of the prior distribution. I argue instead for a particular decision function by virtue of its satisfying certain "commonsense" constraints on its general form. I then move to an approximation of this algorithm when given only partial knowledge of the relevant second-order statistics.

Concerning f_i and f_j, we can, at best, have available a complete description of the joint statistics of these two random variables. Let us say, for definiteness, that each feature, f_i, can have values + or −, as in the circles world. Then the most

complete possible information is summarized by the joint probability distribution function

$$P\ (f_i = a, f_j = b), \qquad \text{a} = + \text{ or } -, \qquad \text{b} = + \text{ or } -;$$

and this information would be available after an infinite number of joint observations of f_i and f_j. Let us suppose, for the time being, that all joint distributions are in fact known and ask by what function of these distributions would we obtain a local opinion for the value of an unobserved feature, f_j. Let O_j represent the strength of the conviction that $f_j = +$. Because f_j is binary, it will be enough to specify a means for computing O_j. Think of $+\infty$ as representing the strongest possible conviction and $-\infty$ as representing the weakest possible conviction (i.e., the strongest possible conviction that $f_j = -$). Suppose it is observed that $f_i = +$. It is evident that the following "boundary conditions" should be in force: If $P\ (f_j = +|f_i = +) = 1$, then O_j is maximal, that is, $O_j = +\infty$; if $P\ (f_j = +|f_i = +) = 0$, then $O_j = -\infty$, (we know that $f_j = -$); if $P\ (f_j = +|f_i = +) = ½$, then the observation $f_i = +$ does not contribute to O_j.

Before writing down an expression for computing O_j, which respects these boundary conditions, I need to introduce some new notation. Concerning a feature f_i, a "+" will be thought of as signaling the presence of an associated event whereas "−" will signal its absence. For each feature, two new variables, y_i and n_i, are defined by

$$y_i = \begin{cases} 1 & \text{if } f_i = +, \\ 0 & \text{if } f_i = - \text{ or } f_i \text{ is unobserved;} \end{cases}$$

$$n_i = \begin{cases} 1 & \text{if } f_i = -, \\ 0 & \text{if } f_i = + \text{ or } f_i \text{ is unobserved.} \end{cases}$$

In words: y_i indicates that the event associated with f_i is present (yes-activity); n_i indicates that this event is absent (no-activity). If both y_i and n_i are 0, then the ith feature is unobserved. Finally, for any pair, i and j, let $r_{ij}^+ = P(f_j = +|f_i = +)$ and let $r_{ij}^- = P(f_j = -|f_i = +)$.

Given complete second-order statistics, one possible functional form for O_j that is consistent with the preceding discussion is

$$\begin{aligned} O_j &= \frac{1}{m}\sum_{i=1}^{n} y_i \log\left(\frac{r_{ij}^+}{1-r_{ij}^+}\right) \\ &= -\frac{1}{m}\sum_{i=1}^{n} y_i \log\,(1-r_{ij}^+) + \frac{1}{m}\sum_{i=1}^{n} y_i \log\,(1-r_{ij}^-) \end{aligned} \tag{9-1}$$

where m = (# observed events) = (# features observed to have value +) = $\sum_{i=1}^{n} y_i$.

The purpose of the factor $1/m$ is to adjust for the effect on O_j of merely observing more events—which, in itself, should not provide evidence toward the true value of f_j. Thus (9-1) describes a procedure by which all observed events are "polled" and two sets of averaged opinions are computed: the opinion that $f_j = +$ and the opinion that $f_j = -$. The conclusion, O_j, is the difference between these. Notice that any observed event can itself determine the value of O_j by implying with certainty, $f_j = +$ or $f_j = -$. Given a threshold T, the local algorithm chooses $f_j = +$ if $O_j > T$, $f_j = -$ if $O_j < -T$, and does not fill in at j if $|O_j| < T$.

Notice that only those observed features with value + (the observed events) contribute to O_j. This asymmetry in the treatment of present versus absent events is mostly for convenience: It allows us to use the present notation without modification to describe temporal recall (first discussed in Section 9.6). There need not be any loss of generality because we can always introduce a new feature such that "+" indicates the *absence* of the event associated with some f_i.

It remains to define the local algorithm for finite experience. Here, of course, the numbers $P(f_j = +|f_i = +)$ and $P(f_j = -|f_i = +)$ are not available. However we may have available functions $r_{ij}^{+}(t)$ and $r_{ij}^{-}(t)$ (t being time) that will, eventually, approximate these conditional probabilities. In this case, O_j can still be computed from Eq. (9-1), with r_{ij}^{+} and r_{ij}^{-} replaced by $r_{ij}^{+}(t)$ and $r_{ij}^{-}(t)$, respectively.

For $r_{ij}^{+}(t)$, an obvious choice is

$$r_{ij}^{+}(t) = \frac{(\#\text{ simultaneous observations of } y_i = 1 \text{ and } y_j = 1 \text{ up to time } t)}{(\#\text{ simultaneous observations of } y_i = 1 \text{ and } y_j = 1 \text{ up to time } t) + (\#\text{ simultaneous observations of } y_i = 1 \text{ and } n_j = 1 \text{ up to time } t)}$$

and a similar expression for $r_{ij}^{-}(t)$ would replace $y_j = 1$ by $n_j = 1$ in the numerator. But, for our purposes, these sample estimators suffer a serious flaw. Suppose, for example, that for some i we have so far observed f_i and f_j simultaneously only once. Then either $r_{ij}^{+}(t)$ or $r_{ij}^{-}(t)$ will be 1, and in any future observation of f_i, the computation of O_j will be dominated by the ith term in one of the summations of (9-1).

Intuitively, the opinion of a feature f_i concerning the value of a feature f_j should be weighed against how often f_i and f_j have been observed simultaneously. How experienced is the opinion? If there have been very few joint observations, then the ith terms in (9-1) should not contribute heavily to the computation of O_j. We seek functions $r_{ij}^{+}(t)$ and $r_{ij}^{-}(t)$ that have the two properties:

1. $r_{ij}^{+}(t) \to P(f_j = +|f_i = +)$ and $r_{ij}^{-}(t) \to P(f_j = -|f_i = +)$ as $t \to$,; and
2. $r_{ij}^{+}(t)$ and $r_{ij}^{-}(t)$ are small when the number of simultaneous observations of f_i and f_j are small.

Notice that the second property would insure that the contribution to O_j by "inexperienced features" is small. One way to achieve these properties is through a realization of the following differential equations:

$$\frac{d}{dt} r_{ij}^{+} = \epsilon(n_i + y_i)(n_j + y_j)y_i(y_j - r_{ij}^{+})$$

and

$$\frac{d}{dt} r_{ij}^{-} = \epsilon(n_i + y_i)(n_j + y_j)y_i(n_j - r_{ij}^{-}). \tag{9-2}$$

ϵ is a small parameter that contributes to the stability of the functions r_{ij}^{+} and r_{ij}^{-}. The factor $(n_i + y_i)(n_j + y_j)$, which appears in each equation, is 1 when f_i and f_j are observed simultaneously and 0 otherwise. Modification, then, only occurs when f_i and f_j are observed simultaneously.

The equations in (9-2) are examples of a class of random equations that can be well approximated (for all time $t\epsilon[0,\infty)$ by a simpler, *deterministic*, system, provided that the sequence of stimuli satisfy a mixing (ergodiclike) assumption (see Geman (1979)). The deterministic system associated with (9-2) is

$$\frac{d}{dt} r_{ij}^{+} = \epsilon\{E[(n_i + y_i)(n_j + y_j)y_iy_j] - E[(n_i + y_i)(n_j + y_j)y_i]r_{ij}^{+}\}$$

$$\frac{d}{dt} r_{ij}^{-} = \epsilon\{E[(n_i + y_i)(n_j + y_j)y_in_j] - E[(n_i + y_i)(n_j + y_j)y_i]r_{ij}^{-}\} \tag{9-3}$$

where E means expected value. The smaller ϵ is, the better the approximation. Recall now the assumption of independence of the *states* (observed vs. unobserved) and *values* (+ or −) of features. One implication is that

$$E[(n_i + y_i)(n_j + y_j)y_iy_j] = pP(f_i = + \quad\text{and}\quad f_j = +),$$

$$E[(n_i + y_i)(n_j + y_j)y_in_j] = pP(f_i = + \quad\text{and}\quad f_j = -),$$

and

$$E[(n_i + y_i)(n_j + y_j)y_i] = pP(f_i = +)$$

where, for short, I have let p stand for the probability of simultaneously observing f_i and f_j. Hence, (9-3) can be written as

$$\frac{d}{dt} r_{ij}^{+} = \epsilon p[P(f_i = + \quad\text{and}\quad f_j = +) - P(f_i = +)r_{ij}^{+}]$$

$$\frac{d}{dt} r_{ij}^{-} = \epsilon p[P(f_i = + \quad\text{and}\quad f_j = -) - P(f_i = +)r_{ij}^{-}], \tag{9-4}$$

which has as solution

$$r_{ij}^{+}(t) = P(f_j = +|f_i = +)\,[1 - \exp(-\epsilon p P(f_i = +)t)]$$

$$r_{ij}^{-}(t) = P(f_j = -|f_i = +)\,[1 - \exp(-\epsilon p P(f_i = +)t)]. \tag{9-5}$$

The result is intuitive: $r_{ij}^{+}(t)$ is asymptotically close to $P(f_j = +|f_i = +)$, and the rate of approach is faster the more frequently that $f_i = +$ and the more frequently that f_i and f_j are observed simultaneously. Notice then that a realization of (9-2) achieves, approximately, properties 1 and 2. [If the values of features are *not* independent of the states of features, then the asymptote is $P(y_j = 1|y_i = 1$ and f_j observed) with rate constant $1/\epsilon\, P(y_i = 1$ and f_j observed).]

In summary I interpret "associative learning" to be a process by which information is gained about pairwise statistics among features. In recall this information is used to evaluate both the *opinion* of a particular decided feature about the value of an undecided feature, and the *experience* of that opinion. Necessarily, then, two pieces of information must be acquired in the learning process. In the design proposed here, the two parameters $r_{ij}^{+}(t)$ and r_{ij}^{-} (t) contain the necessary information. Thus:

Proposition 5. Associative learning is a process of gaining pairwise statistics among features. This information is utilized in associative recall to evaluate both the opinion of a particular decided feature and the experience behind that opinion. The net opinion about an undecided feature is influenced more by more experienced features.

9.4. A SECOND FORM OF LEARNING: DEVELOPMENT OF HIGH-ORDER FEATURES.

How effectively does the proposed process for recall estimate values of target features? The main limitations seem to be the absence of explicit representations for potentially important high-order statistics. Using the Venn diagram introduced earlier, we suppose that $f_i = +$ and $f_{i'} = +$ indicate regions A and B, respectively. Suppose further that A ∩ B is not explicitly available in the sense of the discussion of the previous section. It may be that for some feature *j:*

$P\ (f_j = +\ |\ f_i = +)$ is nearly 1,

$P\ (f_j = +\ |\ f_{i'} = +)$ is nearly 1, but

$P\ (f_j = +\ |\ f_i = +$ and $f_{i'} = +)$ is 0.

Eventually the recall algorithm choses $f_j = +$ whenever the observation is $f_i = +$ and $f_{i'} = +$. There is as yet no mechanism by which this error will be corrected.

The problem is a familiar one in statistics. In regression, for example, an equation involving only linear combinations of the independent variables is sometimes inadequate. The addition of variables that are themselves higher-order functions of the independent variables may considerably improve the performance of the equation. Loosely speaking, in the foregoing example we are attempting to regress the value of the feature f_j onto those of the features f_i and $f_{i'}$. Another way of saying that we do not have explicitly available A ∩ B is to say that second-order functions in the features f_i and $f_{i'}$ are not available. Or perhaps a closer analogy is to the pattern classification problem, if we think of f_j as representing a binary classification. A decision surface derived only from joint statistics between the classification and the individual features proves to be a poor classifier. It is often necessary to make explicit use of higher-order statistics among the features.

I take it for granted that:

Proposition 6. In processing information the nervous system makes explicit use of statistics that are of high order in the most primitive features.

Recall the notation introduced in the previous section: $y_i = 1$ indicates the observation $f_i = +$, and $n_i = 1$ indicates the observation $f_i = -$. Because we can always introduce new features in which the roles of the values + and − have been interchanged, I can without loss of generality assume that the important information is contained in the observations of the random variables y_i, (i = 1, 2, . . ., n). Then, an *explicit* representation of all potentially available (and important) information is equivalent to having a realization of every function of the form

$$z = y_{i_1} y_{i_2} \cdots y_{i_k} \tag{9-6}$$

where $1 \leqslant k \leqslant n$ and $1 \leqslant i_j \leqslant n$ for each j. In words, an explicit representation of all important information requires the indicator functions of intersections of arbitrary collections of the regions defined by $y_i = 1$, (i = 1, 2, . . ., n).

It is completely clear that all functions of the form (9-6) cannot be explicitly represented. The number of such functions is enormous in any but the most trivial examples. There are at least two obvious criteria that an indicator function of this type should satisfy before any "machinery" is committed to its explicit representation. One is frequency of occurrence. If an event never occurs there is certainly no purpose in developing for it an explicit representation. It may happen, for example, that the intersection of the sets defined by $f_i = +$ and $f_{i'} = +$ is empty, in which case $f_i = +$ and $f_{i'} = +$ will never occur simultaneously.

Everything else being equal, the events that are represented should be those occurring most frequently. But it is also true that some events are more important than others, and the statistics associated with such events should be given some measure of preference in committing the available machinery. More will be said later about what makes an event "important," but roughly what I mean is that certain events have associated with them a "hard-wired" *value*—they are painful or pleasurable; they may satisfy a particular need; etc. A possible example of such an event, relevant to the discussion found at the beginning of this section, is "prediction error"; it would be natural to assign to this occurrence a (negative) hard-wired value. Such events are specified by the direction primitives referred to in the introduction, and discussed in more detail in the final section, Section 9.6. The point to be made here is that events that are better correlated with important events should be better represented. Thus the event A ∩ B (introduced earlier in this section) would be given some priority as a candidate for explicit representation by virtue of its likely correlation with prediction errors.

These considerations suggest a second form of plasticity, one which is distinct, at least in purpose, from the associative-type modification postulated in the previous section. The purpose of this second form of plasticity is to commit machinery (perhaps cells or functionally grouped collections of cells) to the representation of statistics that are of increasingly high order. It is a common idea that such a process exists in the nervous system, and that, in a hierarchical fashion, units so-far committed form the "primitives" for the commitment of still higher-level units. I propose, in addition, that this process is biased towards more frequent and more "important" statistics, thus:

Proposition 7. There is a form of plasticity whose purpose is the commitment of neural machinery to the representation of high-order statistics. Successive levels in a hierarchy commit themselves to joint statistics among previously committed units of lower levels. At every level of the hierarchy commitment is to those statistics that are, by some measure, most frequent and most important. A statistic's "importance" is its correlation to important events, and these are innately defined.

Thus "importance" acts something like a "now print" (see Livingston, 1967), nonspecifically overweighting those statistics with which it is associated.

The hypothesis is that the commitment is hierarchical, not only in the sense that increasingly higher-order statistics become represented, but, as well, in the sense that the statistics of one stage form the substrates for the statistics at the next stage. One could imagine a hierarchy in which every stage drew solely from the first stage, seeking increasingly higher-order statistics. However in order to preserve the possibility of commitment to any statistic of a given order, one would need an unimaginably rich initial connectivity between the highest levels and the first level. The successive scheme suggested here avoids this difficulty but not

without a price. Successive commitment limits the repertoire of a given level by what has already been chosen in the previous levels.

For definiteness, I outline the method of commitment used in our simulations, but I do not mean to suggest that this method is in any sense physiological. The assumption here is that the *result* of commitment is to create a representation of those statistics that are a combination of frequent and important. No assumption is intended about the specific mechanism by which this may be accomplished in the nervous system. (The problem of identifying such a mechanism is closely related to some interesting theoretical work by Bienenstock, 1980; Cooper, Liberman, & Oja, 1979; and von der Malsburg, 1973.)

Think of features as being represented by units consisting of two nodes, a yes-node and a no-node. Activity in the yes-node of the *i*th unit represents $y_i = 1$; activity in the corresponding no-node represents $n_i = 1$. These units are organized into a series of levels with, say, level 1 at the bottom, level 2 just above this, and so on. (The levels do not necessarily have equal numbers of units.) Initially only level 1 units are active and are participating in associative learning and recall. Following a specified "critical period" (defined as an a priori fixed number of observations at level 1), level 2 units become active, representing newly defined features, which henceforth participate in the associative processes. The result of this commitment is that the yes-node of a level 2 unit signals an event of the form $y_i = 1$ *and* $y_j = 1$, where f_i and f_j are two features represented in the first level. The corresponding no-node signals that either $n_i = 1$ or $n_j = 1$ or both n_i and n_j are 1 (i.e., there is enough evidence to establish that not both f_i and f_j are 1). Formally, the new feature, say f_k, is defined by

$$y_k = y_i\, y_j, \quad \text{and} \quad n_k = n_i + n_j - n_i\, n_j.$$

For each pair of yes nodes in level one, the number of times that these nodes are simultaneously active is recorded. This number is augmented by the observed correlation between such activity and activity in the direction (good/bad) primitives referred to earlier. If level 2 has m units, then, at the end of the critical period, these units are committed to a representation of the m pairs of level 1 units with the largest so-computed indexes. Following the commitment of level 2, over the next critical period, level 3 is committed using pairs of *level 2* with level 1 units. And then, level 4 commits, using pairs from level 2; and then level 5, using pairs of level 3 with level 2; and so on (see fig. 9.2). For illustration, imagine that every level has m units. Then at any given time, we are maintaining a list of only order m^2 indexes, and this is perfectly manageable.

At all times all committed units participate in the processes of associative learning and associative recall. Early in the experience of the system the lowest levels of the committed units will dominate the associative recall process. This is a result of the preference given to experienced units in the calculation of local opinions, as discussed in Section 9.3. Later, higher-level units will exert the greatest influence on recall. This derives from the fact that the yes-nodes as-

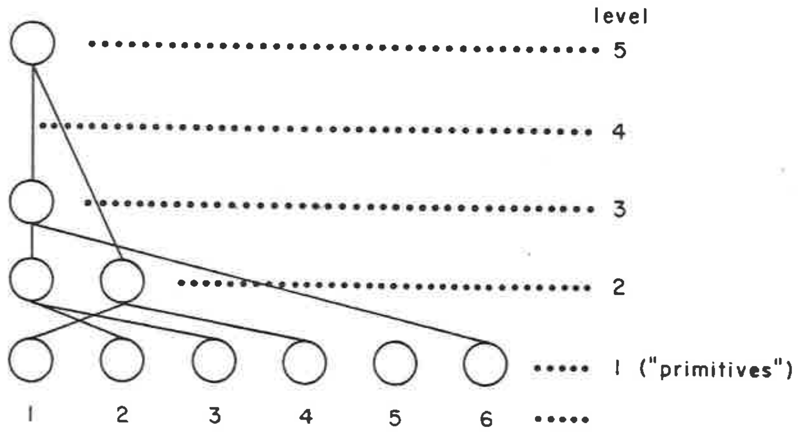

FIG. 9.2. Example of units committed to high order relations. The first level 5 unit, initially inactive, now indicates the simultaneous occurrence of the events associated with the first level 3 unit and the second level 2 unit (equivalently, the simultaneous occurrence of the events associated with units 1,2,3,4, and 6 of level 1).

sociated with these units represent more selective events (viz. the *intersections* of events associated with lower-level units), implying that the conditional probabilities given activities in these yes-nodes will tend to be closer to 0 or 1. Consequently, the associated r_{ij}s will have asymptotic values that are also closer to 0 or 1, and [by Eq. (9-1)] these coefficients will eventually dominate the calculation of local opinions.

I pointed out in the introduction that this model is based on a compromise between local and distributed representations of information. At the lowest (earliest) levels of processing, a stimulus has a distributed representation; the activities of numerous units each signal the presence of a lower-order event contained in the stimulus. At successive levels of processing the representation becomes better localized; more selective units signal the presence of more complex combinations of events peculiar to the stimulus.

9.5 A SPATIAL CODING MODULE

The discussion so far has been of a model for representing, updating, and retrieving information about those relations among features that may exist at a fixed time. As of yet there is no mechanism in this model for processing temporal relations among features. I call those relations that do not contain a temporal component, spatial relations, and the machinery proposed for processing such information a spatial coding module (SCM, for short). What I say about the processing of *temporal* information is closely patterned after the model for processing spatial information. As an introduction to a temporal coding module, as well as a summary of the model so far, this section briefly reviews the proposed mechanisms for representing, learning, and recalling spatial relationships.

The SCM consists of layers of possibly differing numbers of units. A unit is a pair of nodes: a yes-node and a no-node. Activity in a yes-node signals the occurrence of an event, whereas activity in the corresponding no-node indicates that the event has not occurred. Absence of activity in either node indicates that the event is unobserved—it may or may not have occurred. Initially only the first layer of units participates in information processing. These layer 1 units are primitives of the SCM; they represent the total information available to the module. Layer 2 units become participants in SCM processing by committing to the representation of a new feature over a specified critical period. As a result of the commitment process activity in a yes-node of a layer 2 unit comes to signal the simultaneous activity in two particular yes-nodes of layer 1. Call the units associated with these two yes-nodes the "substrates" of the layer 2 unit. The no-node in this layer 2 unit signals activity in the no-node of at least one of the substrate units. Layer 2 units commit to those pairs of units that, over the critical period, are most frequently observed to have simultaneous activities in their yes-nodes as well as being most highly correlated with activities in the direction primitives. Over ensuing critical periods, layer 3 units commit to pairs consisting of one layer 2 unit and one layer 1 unit; then layer 4 units commit to pairs of layer 2 units; then layer 5 units commit to pairs of one layer 3 unit and one layer 2 unit, and so on.

At all times, those units that are committed participate in the associative learning and recall processes. Associative learning is the calculation of conditional probabilities: Each yes-node in the module computes for every other node in the module (excepting its no-node pair), the conditional probability of activity in that node given that the corresponding unit is active (observed) and that the yes-node itself is active.

Recall is a process of filling in yes-or-no activities at unobserved units. Activity at any given unobserved unit can be filled in by examining the associations to the yes- and no-nodes of that unit from all active yes-nodes. Estimates of the true value (yes or no) of a particular collection of unobserved units (target units) are derived by a recursive process which repeatedly fills in all unobserved units for which the active yes-nodes (observed plus filled-in) strongly suggest a value. The process terminates when all target units are filled in.

Before closing this section, it may be worthwhile to anticipate some of the discussion of Section 9.7, on a control module. Evidently, mechanisms must be developed for the execution of such activities as choosing target units, setting a threshold for the determination of which units are filled in at a particular step in the recall process, deciding when the SCM should be in "recall mode" and when t should be learning associations and committing new units, etc. It is the status f these decisions—decisions that control the activity of the SCM—that form the rimitives (level 1 units) for the control module. In the design that I later propose he control module processes the activities associated with these primitives in much the same way as the SCM processes the activities that signal events in its environment.

9.6. A TEMPORAL CODING MODULE

I make a distinction between two types of memories, those that associate events that tend to occur simultaneously and those that associate events that tend to occur in sequence. Although this separation is largely artificial, it is a useful idealization, and it can be expected to suggest mechanisms that recognize the continuum between "spatial" and "temporal" information. Imagine that time is discrete and that a stimulus occurs at each instant. Stimuli are represented exactly as before: a set of values composing a feature vector with explicit account taken for observed versus unobserved states.

The purpose of the "temporal coding module" (TCM) is to learn, and be able to recall, temporal relations that may exist among the features. The design principles are those of the spatial coding module: the commitment of machinery to high-order statistics, the learning of second-order statistics (conditional probabilities) between nodes, the reconstruction of events by a local filling-in process together with a global recursion process. Given these parallels, and given the detail with which the SCM has been described, an outline of the TCM design will substitute for a complete description.

The nature of the information processed by a TCM is determined by the primitives for the module. As with the SCM these primitives are the layer 1 units, each of which consists of 2 nodes having the same interpretation as in the SCM. Following the previous development, we think of primitive only as relative to the rest of the module. Primitive may refer to the detection of a particular frequency or of a particular type of transition in an auditory signal, or an entire word or even phrase may be primitive. A primitive may be a *top*-level unit of an SCM or of another TCM.

In analogy to the need for an explicit representation of high-order spatial statistics, there is a need here for an explicit representation of high-order temporal statistics. The reasoning is the same: It would appear that learning is the computation of second-order relations—relations among pairs. If neither of these pairs is itself of higher order than the primitives, then the organism ignores forever high-order statistics, and it is obvious that the behavior of the nervous system is not merely a function of second-order relationships. In the SCM, with binary features, "high order" referred to the intersection of a collection of features. Activity in a yes-node of a level 2 unit represented the simultaneous occurrence of activities in the yes-nodes of two level one units. The analog for

the TCM is the hierarchical representation of *permutations* of pairs of lower-level units. Hence, a yes-node in the second level will come to indicate the completion of a sequence of activities in two consecutive yes-nodes in level 1. For example, a level 2 unit may have as substrate the (i, j) *ordered* pair of level 1 units. If activity in the i unit yes-node is followed immediately by activity in the j unit yes-node, then the yes-node in this level 2 unit is activated. (In our simulations, the timing is defined such that this level 2 unit is active simultaneously with the completion of the pair, i.e., the level 2 activity coincides with the activity in the second member of the level 1 pair.) If either the i level 1 unit no-node was active at the previous time or the j level 1 unit no-node is currently active, then the no-node in this level 2 unit is currently active. Any other sequence of (i, j) level 1 unit activities produces no activity (unobserved state) in this level 2 unit. Level 3 units represent permutations of level 2 followed by level 1 units; level 4 units represent permutations of pairs of level 2 units; level 5 units represent permutations of level 3 followed by level 2 units; and so on. (Activity in a level p unit is said to *follow* activity in a level q unit if it occurs during the pth period of time after activity in the level q unit. In other words, the sequence of primitives associated with the level q unit must immediately precede the sequence of primitives associated with the level p unit.)

We are again faced with the problem of selecting a small fraction of all possible statistics of a given order to which we will commit machinery. The solution, for the TCM, is the same as for the SCM. Levels commit following successive critical periods, and commitment is to those permutations most frequently observed and most highly correlated with activities in the direction primitives. Eventually activity in the yes-node of a unit in the kth level signals the completion of a particular sequence of k level 1 yes-node activites.

Associative learning in the TCM is by the same mechanism as in the SCM, except that the r_{ij}s update when activity in one unit "follows" (recall definition) yes-node activity in another unit rather than when activities occur simultaneously. Asymptotically, $r_{ij}^{+}(r_{ij}^{-})$ approximates (in the sense discussed in Section 9.3) the probability that activity in the yes- (no-) node at j follows activity in the yes-node at i, given that activity at j is observed.

From the collection of activities in a particular set of units in a TCM, an ensuing sequence is predicted using a mechanism similar to the filling-in process proposed for the SCM. The TCM first chooses that yes-node whose activity is most strongly suggested by the activities present in the network. The decision function for this choice is exactly the one used in the SCM, and I would support its appropriateness by the arguments used there. After a node is chosen the values of all other units are recomputed just as though the sequence of primitives associated with the chosen unit had been observed, and from here the prediction process can continue.

As in the discussion of an SCM, it is worth noting here that a set of primitives of a control (versus environmental) nature have been implicitly defined. These primitives determine the depth of a temporal prediction, perhaps the level from which the filled-in unit is drawn, perhaps which other modules should influence the prediction, and certainly when such a prediction should be attempted. It is primitives such as these that comprise level 1 of the control module.

9.7 INTEGRATION

Temporal and spatial coding modules can be integerated, in a parallel or hierarchical structure, without modification of the mechanisms so far hypothesized. A parallel structure requires a degree of connectivity which will permit the state (set of active units) of one module to influence the local prediction (filling-in) process in the other. A spatial-to-temporal connection associates by the rules of the TCM, computing, asymptotically, the conditional probability that the TCM node "follows" (as defined in Section 9.6) the SCM node, given that the TCM unit is observed. A temporal-to-spatial–type connection associates by the concurrence rule used in the SCM and is asymptotically equal to the conditional probability of the SCM node being active, given that the TCM node is active and that the SCM unit is observed. Of course, full connectivity between modules is, at some point, not practical. Given a constraint on connectivity, it would be natural to use the more selective, and presumably more informative, higher-level units from a given module for communication to another module. A hierarchical structure can be achieved by taking as primitives, for one module, the top-level units of one or several other modules. The processing of information through such a network is well defined whatever the identities (spatial or temporal) of the individual modules. It is the availability of this hierarchy, a hierarchy that demands no new principles of architecture or function, which I believe justifies the notion of feature as it is introduced in Section 9.2, and used throughout this chapter.

Yet the design remains that of an "open" system. There is no mechanism for determining when an SCM begins a recall process, or what threshold is used during this process, or to what depth a TCM search should go. These control-type decisions can be organized by a logical structure that is very much akin to the structures proposed for temporal and spatial coding. Let us represent each available control activity by a unit of the type used in the TCM and SCM. Yes-node activity means that the corresponding action is being executed. For example, such activity may initiate the associative selection of a unit in a particular level of a particular TCM, or the setting of a recall threshold to "high" in a particular SCM. I assume that these primitive actions have their analog in the nervous system:

Proposition 8. There is a special class of motor primitives (*control primitives*) whose activities effect changes in the state of neural machinery.

The *control module* is a layered network of units in which these primitives form the first row. Its purpose is to learn to choose actions that are "appropriate" given the current available information concerning the state of the environment and the state of the network it controls. The principles of operation are analogous to those for the SCM or TCM, with choosing the "next move" (or next sequence of moves) playing the role of filling-in in the SCM or predicting in the TCM.

In the control module, as in the SCM and the TCM, higher-level units come to represent sequences or combinations of level 1 primitives. The argument for the existence of such units is much the same: It is obvious that explicit account must be taken of high-order relations in the control primitives, and it would seem that a local representation of such relations is necessary if it is the connections between pairs of units that ultimately determine how we get from a currently active set of units to a newly activated unit.

I have said that the control module operates in a manner much like the SCM and the TCM. In particular the control module chooses a next move or sequence of moves in a manner analogous to the filling-in or prediction process of the SCM or TCM. The action of the control module, then, is largely determined by the strengths of the r_{ij}s associated with the various connections to its units. If it is the purpose of the control module to choose "appropriate" actions, then we must interpret the strengths of the r_{ij} coefficients in a way which is fundamentally different from their interpretations in the SCM or TCM. In particular, Eq. (9-2) can no longer apply to the updating of these coefficients. It is the control module itself, using these coefficients, that determines the sequence of units which are activated, and therefore these coefficients can not simply reflect the relative frequency with which one node has followed another.

The missing ingredient is, of course, a definition of *appropriate action*. On this point, even a superficial development could consume an entire article. Instead of attempting a proper defense for the way in which "appropriate action" is defined here, I am simply describing its functional realization in the control module and hope that the reader will not find this realization unituitive.

The notion of appropriate action is based on a special class of primitives (*direction primitives*) whose activities, through their influence on the control module, ultimately determine the direction of the network. These premitives model those inputs that can be interpreted as having an *inherently* good or bad meaning to the nervous system. In the system, good or bad meaning is defined operationally; it is the effect of activity in a unit representing a direction primitive that determines the extent to which that primitive is good or bad. Other inputs will come to have good or bad value by virtue of their associations with activities in these units, but the development of such associations requires no new mechanisms; they result from the modification of coupling coefficients situated between SCM, TCM, or control module units and those units representing direction primitives; to summarize:

Proposition 9. There is a special class of primitives (*direction primitives*) that, by virtue of their influence on learning (commitment and associative), can be thought of as having positive or negative meaning to the nervous system.

The particular primitives employed depend on the particular application. Some direction primitives for the numbers world simulation were mentioned in the introduction. Presumably such things as hunger, pain, and perhaps joy, companionship, and the like play the roles of direction primitives for people. (Here, of course, it would be nearly impossible to distinguish what is truly a primitive from what has been associated with a primitive.)

The influence of activity in the direction primitives on the commitment process in SCMs and TCMs has already been described. Activities in direction primitives are also responsible for the development of the coupling coefficients of the control module (i.e., all r_{ij} for which j represents a unit of the control module). Roughly, if the unit i has been "followed" (recall the definition given in Section 9.6) by the unit j, and if there is net positive activity in the set of direction primitives, then r_{ij}^{+} is incremented upward and r_{ij}^{-} incremented downward. If net activity in the direction primitives is negatie, then movement of the coupling coefficients is in the opposite direction.

Proposition 10. In networks of control features (features that derive from control primitives) associative learning is determined by activity in direction primitives rather than by observed correlation. Positive activity reinforces currently active associations; negative activity reinforces the negative of currently active associations.

The actual equations are such that the r_{ij}s remain always between 0 and 1, so that the local decision function, Eq. (9-1), still makes sense. The control module chooses a next move by looking ahead some fixed number of iterations (which number may itself be a control primitive) and evaluating the expected positive or negative consequences of a given sequence. "Looking ahead" means choosing potential paths by the TCM procedure, governed by the r_{ij}s (i may refer to a unit in any of the three types of modules). "Evaluating" means taking a certain average of the associations with direction primitives over the units in a particular path.

Proposition 11. Networks of control features activate new features by a combination of filling-in (guided by connectivities) and a bias towards features strongly associated with positive activity in direction primitives.

The details are not important. What should be emphasized is the position that *declarative* and *procedural* knowledge have essentially the same representation.[3] The structure envisioned for temporal and spatial coding of the environment applies to the problem of organizing an appropriate direction for the network as a

[3] For a good discussion of this issue and an example of a model based on the opposite proposition, see Anderson (1976).

whole. Thus units are successively committed to the representation of more and more complex actions, and the choice of new actions is driven by coefficients containing information about the significance of pairwise activities in these units.

My approach has demanded a strict separation of the notions of temporal and spatial associations, a binary notion of feature, and a rigid definition of temporal ordering. The advantage is that the logic of the processing is largely accessible. We can be reasonably sure that certain stability problems are avoided and that the architecture truly lends itself to a hierarchy, that "more is better." But the approach is severely restrictive. There are few environments that respect these conditions of regularity. The approach taken here is based on the expectation that the right generalizations will come from models sufficiently idealized to permit thorough analysis and simulation.

ACKNOWLEDGMENT

Research supported in part by the National Science Foundation through Grant MCS 76-07203 and by the Air Force Office of Scientific Research through Grant 78-3514.

REFERENCES

Anderson, J. R. *Language, memory, and thought*. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1976.

Barlow, H. B. Single units and sensation: A neuron doctrine for perceptual psychology? *Perception*, 1972, *1*, 371–394.

Bienenstock, E. L. A theory of development of neuronal selectivity. Unpublished doctoral dissertation, Div. of Appl. Mathematics, Brown University, June, 1980.

Cooper, L. N., Liberman, F., & Oja, E. A theory for the acquisition and loss of neuron specificity in visual cortex. *Biological Cybernetics*, 1979, *33*, 9–28.

Geman, S. A method of averaging for random differential equations with applications to stability and stochastic approximations. In A. T. Bharucha-Reid (Ed.), *Approximate solution of random equations*. Amsterdam: Elsevier-North Holland, 1979.

Hayes-Roth, B. Evolution of cognitive structure and processes. *Psychological Review*, 1977, *84*, 260–278.

Kohonen, T. *Associative memory, a system theoretic approach*. Berlin: Springer-Verlag, 1977.

Lashley, K. S. In search of the engram. *Society of Experimental Biology Symposium* (No. 4): *Physiological mechanisms in animal behavior*. Cambridge, England: Cambridge University Press, 1950.

Livingston, R. B. Brian circuitry relating to complex behavior. In G. C. Quarton, P. Melnechuk, & F. O. Schmitt (Eds.), *The Neurosciences*. New York: Rockefeller University Press, 1967.

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