Journal of Experimental Psychology: General
1985, Vol. 114, No. 2, 159–188

Distributed Memory and the Representation of General and Specific Information

James L. McClelland and David E. Rumelhart
University of California, San Diego

We describe a distributed model of information processing and memory and apply it to the representation of general and specific information. The model consists of a large number of simple processing elements which send excitatory and inhibitory signals to each other via modifiable connections. Information processing is thought of as the process whereby patterns of activation are formed over the units in the model through their excitatory and inhibitory interactions. The memory trace of a processing event is the change or increment to the strengths of the interconnections that results from the processing event. The traces of separate events are superimposed on each other in the values of the connection strengths that result from the entire set of traces stored in the memory. The model is applied to a number of findings related to the question of whether we store abstract representations or an enumeration of specific experiences in memory. The model simulates the results of a number of important experiments which have been taken as evidence for the enumeration of specific experiences. At the same time, it shows how the functional equivalent of abstract representations—prototypes, logogens, and even rules—can emerge from the superposition of traces of specific experiences, when the conditions are right for this to happen. In essence, the model captures the structure present in a set of input patterns; thus, it behaves as though it had learned prototypes or rules, to the extent that the structure of the environment it has learned about can be captured by describing it in terms of these abstractions.

In the late 1960s and early 1970s a number of experimenters, using a variety of different tasks, demonstrated that subjects could learn through experience with exemplars of a category to respond better—more accurately, or more rapidly—to the prototype than to any of the particular exemplars. The seminal demonstration of this basic point comes from the work of Posner and Keele (1968, 1970). Using a categorization task, they found that there were some conditions in which subjects categorized the prototype of a category more accurately than the particular exemplars of the category that they had previously seen. This work, and many other related experiments, supported the development of the view that memory by its basic nature somehow abstracts the central tendency of a set of disparate experiences, and gives relatively little weight to the specific experiences that gave rise to these abstractions.

Recently, however, some have come to question this "abstractive" point of view, for two reasons. First, specific events and experiences clearly play a prominent role in memory and learning. Experimental demonstrations of the importance of specific stimulus events even in tasks which have been thought to involve abstraction of a concept or rule are now legion. Responses in categorization tasks (Brooks, 1978; Medin & Shaffer, 1978), perceptual identification tasks (Jacoby, 1983a,

Preparation of this article was supported in part by a grant from the Systems Development Foundation and in part by a National Science Foundation Grant BNS-79-24062. The first author is a recipient of a Career Development Award from the National Institute of Mental Health (5-K01-MH00385).

This article was originally presented at a conference organized by Lee Brooks and Larry Jacoby on "The Priority of the Specific." We would like to thank the organizers, as well as several of the participants, particularly Doug Medin and Rich Shiffrin, for stimulating discussion and for empirical input to the development of this article.

Requests for reprints should be sent to James L. McClelland, Department of Psychology, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213 or to David E. Rumelhart, Institute for Cognitive Science, C-015, University of California—San Diego, La Jolla, California 92093.

1983b; Whittlesea, 1983), and pronunciation tasks (Glushko, 1979) all seem to be quite sensitive to the congruity between particular training stimuli and particular test stimuli, in ways which most abstraction models would not expect.

At the same time, a number of models have been proposed in which behavior which has often been characterized as *rule-based* or *concept-based* is attributed to a process that makes use of stored traces of specific events or specific exemplars of the concepts or rules. According to this class of models, the apparently rule-based or concept-based behavior emerges from what might be called a conspiracy of individual memory traces or from a sampling of one from the set of such traces. Models of this class include the Medin and Shaffer (1978) context model, Hintzman's (1983) multiple trace model, and Whittlesea's (1983) episode model. This trend is also exemplified by our interactive activation model of word perception (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1981, 1982), and an extension of the interactive activation model to generalization from exemplars (McClelland, 1981).

One feature of some of these exemplar-based models troubles us. Many of them are internally inconsistent with respect to the issue of abstraction. Thus, though our word perception model assumes that linguistic rules emerge from a conspiracy of partial activations of detectors for particular words, thereby eliminating the need for abstraction of rules, the assumption that there is a single detector for each word implicitly assumes that there is an abstraction process that lumps each occurrence of the same word into the same single detector unit. Thus, the model has its abstraction and creates it too, though at slightly different levels.

One logically coherent response to this inconsistency is to simply say that each word or other representational object is itself a conspiracy of the entire ensemble of memory traces of the different individual experiences we have had with that unit. We will call this view the *enumeration of specific experiences* view. It is exemplified most clearly by Jacoby (1983a, 1983b), Hintzman (1983), and Whittlesea (1983).

As the papers just mentioned demonstrate, enumeration of specific experiences can work quite well as an account of quite a number of empirical findings. However, there still seems to be one drawback. Such models seem to require an unlimited amount of storage capacity, as well as mechanisms for searching an almost unlimited mass of data. This is especially true when we consider that the primitives out of which we normally assume one experience is built are themselves abstractions. For example, a word is a sequence of letters, or a sentence is a sequence of words. Are we to believe that all of these abstractions are mere notational conveniences for the theorist, and that every event is stored as an extremely rich (obviously structured) representation of the event, with no abstraction?

In this article, we consider an alternative conceptualization: a distributed, superpositional approach to memory. This view is similar to the separate enumeration of experiences view in some respects, but not in all. On both views, memory consists of traces resulting from specific experiences; and on both views, generalizations emerge from the superposition of these specific memory traces. Our model differs, though, from the enumeration of specific experiences in assuming that the superposition of traces occurs at the time of storage. We do not keep each trace in a separate place, but rather we superimpose them so that what the memory contains is a composite.

Our theme will be to show that distributed models provide a way to resolve the abstraction–representation of specifics dilemma. With a distributed model, the superposition of traces automatically results in abstraction though it can still preserve to some extent the idiosyncrasies of specific events and experiences, or of specific recurring subclasses of events and experiences.

We will begin by introducing a specific version of a distributed model of memory. We will show how it works and describe some of its basic properties. We will show how our model can account for several recent findings (Salasoo, Shiffrin, & Feustel, 1985; Whittlesea, 1983), on the effects of specific experiences on later performance, and the conditions under which functional equivalents of abstract representations such as prototypes or logogens emerge. The discussion considers generalizations of the approach to the semantic-episodic distinction and the acquisition of linguistic rule systems, and considers reasons for preferring a distributed-superpositional memory over other models.

Previous, related models. Before we get down to work, some important credits are in order. Our distributed model draws heavily from the work of Anderson (e.g., 1977, 1983; Anderson, Silverstein, Ritz, & Jones, 1977; Knapp & Anderson, 1984) and Hinton (1981a). We have adopted and synthesized what we found to be the most useful aspects of their distinct but related models, preserving (we hope) the basic spirit of both. We view our model as an exemplar of a class of existing models whose exploration Hinton, Anderson, Kohonen (e.g., Kohonen, 1977; Kohonen, Oja, & Lehtio, 1981), and others have pioneered. A useful review of prior work in this area can be obtained from Anderson and Hinton (1981) and other articles in the volume edited by Hinton and Anderson (1981). Some points similar to some of these we will be making have recently been covered in the papers of Murdock (1982) and Eich (1982), though the distributed representations we use are different in important ways from the representations used by these other authors.

Our distributed model is not a complete theory of human information processing and memory. It is a model of the internal structure of some components of information processing, in particular those concerned with the retrieval and use of prior experience. The model does not specify in and of itself how these acts of retrieval and use are planned, sequenced, and organized into coherent patterns of behavior.

A Distributed Model of Memory

General Properties

Our model adheres to the following general assumptions, some of which are shared with several other distributed models of processing and memory.

Simple, highly interconnected units. The processing system consists of a collection of simple processing units, each interconnected with many other units. The units take on activation values, and communicate with other units by sending signals modulated by weights associated with the connections between the units. Sometimes, we may think of the units as corresponding to particular representational primitives, but they need not. For example, even what we might consider to be a primitive feature of something, like having a particular color, might be a pattern of activation over a collection of units.

Modular structure. We assume that the units are organized into modules. Each module receives inputs from other modules, the units within the module are richly interconnected with each other, and they send outputs to other modules. Figure 1 illustrates the internal structure of a very simple module, and Figure 2 illustrates some hypothetical interconnections between a number of modules. Both figures grossly underrepresent our view of the numbers of units per module and the number of modules. We would imagine that there would be thousands to millions of units per module and many hundreds or perhaps many thousands of partially redundant modules in anything close to a complete memory system.

The state of each module represents a synthesis of the states of all of the modules it receives inputs from. Some of the inputs will be from relatively more sensory modules, closer to the sensory end-organs of one modality or another. Others will come from relatively more abstract modules, which themselves receive inputs from and send outputs to other modules placed at the abstract end of several different modalities. Thus, each module combines a number of different sources of information.

Mental state as pattern of activation. In a distributed memory system, a mental state is a pattern of activation over the units in some subset of the modules. The patterns in the different modules capture different aspects of the content of the mental states in a partially overlapping fashion. Alternative mental states are simply alternative patterns of activation over the modules. Information processing is

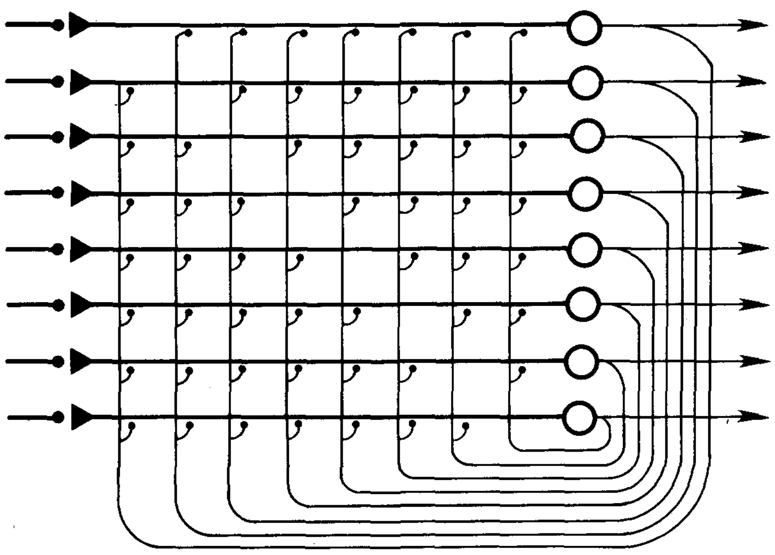

Figure 1. A simple information processing module, consisting of a small ensemble of eight processing units. [Each unit receives inputs from other modules (indicated by the single input impinging on the input line of the node from the left; this can stand for a number of converging input signals from several nodes outside the module) and sends outputs to other modules (indicated by the output line proceeding to the right from each unit). Each unit also has a modifiable connection to all the other units in the same module, as indicated by the branches of the output lines that loop back onto the input lines leading into each unit. All connections, which may be positive or negative, are represented by dots.]

the process of evolution in time of mental states.

Units play specific roles within patterns. A pattern of activation only counts as the same as another if the same units are involved.

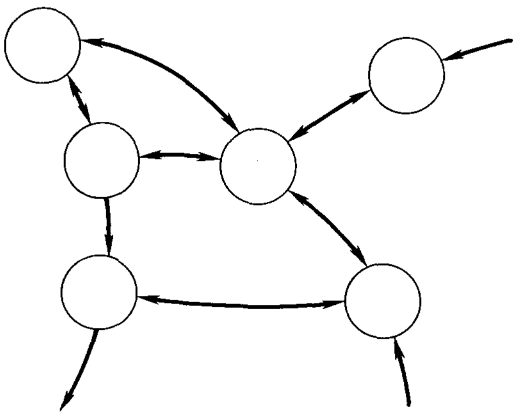

Figure 2. An illustrative diagram showing several modules and interconnections among them. (Arrows between modules simply indicate that some of the nodes in one module send inputs to some of the nodes in the other. The exact number and organization of modules is of course unknown; the figure is simply intended to be suggestive.)

The reason for this is that the knowledge built into the system for recreating the patterns is built into the set of interconnections among the units, as we will explain later. For a pattern to access the right knowledge it must arise on the appropriate units. In this sense, the units play specific roles in the patterns. Obviously, a system of this sort is useless without sophisticated perceptual processing mechanisms at the interface between memory and the outside world, so that similar input patterns arising at different locations in the world can be mapped into the same set of units internally. Such mechanisms are outside the scope of this article (but see Hinton, 1981b; McClelland, 1985).

Memory traces as changes in the weights. Patterns of activation come and go, leaving traces behind when they have passed. What are the traces? They are changes in the strengths or *weights* of the connections between the units in the modules.

This view of the nature of the memory trace clearly sets these kinds of models apart from traditional models of memory in which some copy of the "active" pattern is generally thought of as being stored directly. Instead of this, what is actually stored in our model is changes in the connection strengths. These changes are derived from the presented pattern, and are arranged in such a way that, when a part of a known pattern is presented for processing, the interconnection strengths cause the rest of the pattern to be reinstated. Thus, although the memory trace is not a copy of the learned pattern, it is something from which a replica of that pattern can be recreated. As we already said, each memory trace is distributed over many different connections, and each connection participates in many different memory traces. The traces of different mental states are therefore superimposed in the same set of weights. Surprisingly enough, as we will see in several examples, the connections between the units in a single module can store the information needed to complete many different familiar patterns.

Retrieval as reinstatement of prior pattern of activation. Retrieval amounts to partial reinstatement of a mental state, using a cue which is a fragment of the original state. For any given module, we can see the cues as originating from outside of it. Some cues could arise ultimately from sensory input. Others would arise from the results of previous retrieval operations fed back to the memory system under the control of a search or retrieval plan. It would be premature to speculate on how such schemes would be implemented in this kind of a model, but it is clear that they must exist.

Detailed Assumptions

In the rest of our presentation, we will be focusing on operations that take place within a single module. This obviously oversimplifies the behavior of a complete memory system because the modules are assumed to be in continuous interaction. The simplification is justified, however, in that it allows us to focus on some of the basic properties of distributed memory that are visible even without these interactions with other modules.

Let us look, therefore, at the internal structure of one very simple module, as shown in Figure 1. Again, our image is that in a real system there would be much larger numbers of units. We have restricted our analysis to small numbers simply to illustrate basic principles as clearly as possible; this also helps to keep the running time of simulations in bounds.

Activation values. The units take on activation values which range from -1 to $+1$. Zero represents in this case a neutral resting value, toward which the activations of the units tend to decay.

Inputs, outputs, and internal connections. Each unit receives input from other modules and sends output to other modules. For the present, we assume that the inputs from other modules occur at connections whose weights are fixed. In the simulations, we treat the input from outside the module as a fixed pattern, ignoring (for simplicity) the fact that the input pattern evolves in time and might be affected by feedback from the module under study. Although the input to each unit might arise from a combination of sources in other modules, we can lump the external input to each unit into a single real valued number representing the combined effects of all components of the external input. In addition to extra-modular connections, each unit is connected to all other units in the module via a weighted connection. The weights on these connections are modifiable, as described later. The weights can take on any real values, positive, negative, or 0. There is no connection from a unit onto itself.

The processing cycle. Processing within a module takes place as follows. Time is divided into discrete ticks. An input pattern is presented at some point in time over some or all of the input lines to the module and is then left on for several ticks, until the pattern of activation it produces settles down and stops changing.

Each tick is divided into two phases. In the first phase, each unit determines its net input, based on the external input to the unit and activations of all of the units at the end of the preceding tick modulated by the weight coefficients which determine the strength and direction of each unit's effect on every other.

For mathematical precision, consider two units in our module, and call one of them unit i, and the other unit j. The input to unit i from unit j, written i_{ij} is just

$$i_{ij} = a_j w_{ij},$$

where a_j is the activation of unit j, and w_{ij} is the weight constant modulating the effect of unit j on unit i. The total input to unit i from all other units internal to the module, i_i, is then just the sum of all of these separate inputs:

$$i_i = \sum_j i_{ij}.$$

Here, j ranges over all units in the module other than i. This sum is then added to the *external* input to the unit, arising from outside the module, to obtain the net input to unit i, n_i:

$$n_i = i_i + e_i,$$

where e_i is just the lumped external input to unit i.

In the second phase, the activations of the units are updated. If the net input is positive, the activation of the unit is incremented by an amount proportional to the distance left to the ceiling activation level of +1.0. If the net input is negative, the activation is decremented by an amount proportional to the distance left to the floor activation level of −1.0. There is also a decay factor which tends to pull the activation of the unit back toward the resting level of 0.

Mathematically, we can express these assumptions as follows: For unit i, if $n_i > 0$,

$$\dot{a}_i = En_i(1 - a_i) - Da_i.$$

If $n_i \leq 0$,

$$\dot{a}_i = En_i[a_i - (-1)] - Da_i.$$

In these equations, E and D are global parameters which apply to all units, and set the rates of excitation and decay, respectively. The term a_i is the activation of unit i at the end of the previous cycle, and $\dot{a}_i$ is the change in a_i; that is, it is the amount added to (or, if negative, subtracted from) the old value a_i to determine its new value for the next cycle.

Given a fixed set of inputs to a particular unit, its activation level will be driven up or down in response until the activation reaches the point where the incremental effects of the input are balanced by the decay. In practice, of course, the situation is complicated by the fact that as each units' activation is changing it alters the input to the others. Thus, it is necessary to run the simulation to see how the system will behave for any given set of inputs and any given set of weights. In all the simulations reported here, the model is allowed to run for 50 cycles, which is considerably more than enough for it to achieve a stable pattern of activation over all the units.

Memory traces. The memory trace of a particular pattern of activation is a set of changes in the entire set of weights in the module. We call the whole set of changes an *increment* to the weights. After a stable pattern of activation is achieved, weight adjustment takes place. This is thought of as occurring simultaneously for all of the connections in the module.

The Delta rule. The rule that determines the size and direction (up or down) of the change at each connection is the crux of the model. The idea is often difficult to grasp on first reading, but once it is understood it seems very simple, and it directly captures the goal of facilitating the completion of the pattern, given some part of the pattern as a retrieval or completion cue.

To allow each part of a pattern to reconstruct the rest of the pattern, we simply want to set up the internal connections among the units in the module so that when part of the pattern is presented, activating some of the units in the module, the internal connections will lead the active units to tend to reproduce the rest. To do this, we want to make the internal input to each unit have the same effect on the unit that the external input has on the unit. That is, given a particular pattern to be stored, we want to find a set of connections such that the internal input to each unit from all of the other units matches the external input to that unit. The connection change procedure we will describe has the effect of moving the weights of all the connections in the direction of achieving this goal.

The first step in weight adjustment is to see how well the module is already doing. If the network is already matching the external input to each unit with the internal input from the other units, the weights do not need to be changed. To get an index of how well the network is already doing at matching its excitatory input, we assume that each unit i computes the difference Δ_i between its external input and the net internal input to the unit from the other units in the module:

$$\Delta_i = e_i - i_i.$$

In determining the activation value of the unit, we added the external input together with the internal input. Now, in adjusting the weights, we are taking the difference between these two terms. This implies that the unit must be able to aggregate all inputs for purposes of determining its activation, but it must be able to distinguish between external and internal inputs for purposes of adjusting its weights.

Let us consider the term Δ_i for a moment. If it is positive, the internal input is not activating the unit enough to match the external input to the unit. If negative, it is activating the unit too much. If zero, everything is fine and we do not want to change anything. Thus, Δ_i determines the magnitude and direction of the overall change that needs to be made in the internal input to unit i. To achieve this overall effect, the individual weights are then adjusted according to the following formula:

$$\dot{w}_{ij} = S\Delta_i a_j.$$

Thei parameter S is just a global strength parameter which regulates the overall magnitude of the adjustments of the weights; $\dot{w}_{ij}$ is the change in the weight to i from j.

We call this weight modification rule the *delta rule.* It has all the intended consequences; that is, it tends to drive the weights in the direction of the right values to make the internal inputs to a unit match the external inputs. For example, consider the case in which Δ_i is positive and a_j is positive. In this case, the value of Δ_i tells us that unit i is not receiving enough excitatory input, and the value of a_j tells us that unit j has positive activation. In this case, the delta rule will increase the weight from j to i. The result will be that the next time unit j has a positive activation, its excitatory effect on unit i will be increased, thereby reducing Δ_i.

Similar reasoning applies to cases where Δ_i is negative, a_j is negative, or both are negative. Of course, when either Δ_i or a_j is 0, w_{ij} is not changed. In the first case, there is no error to compensate for; in the second case, a change in the weight will have no effect the next time unit j has the same activation value.

What the delta rule can and cannot do. The delta rule is a continuous variant of the perceptron convergence procedure (Rosenblatt, 1962), and has been independently invented many times (see Sutton & Barto, 1981, for a discussion). Its popularity is based on the fact that it is an error-correcting rule, unlike the Hebb rule used until recently by Anderson (1977; Anderson et al., 1977). A number of interesting theorems have been proven about this rule (Kohonen, 1977; Stone, 1985). Basically, the important result is that, for a set of patterns which we present repeatedly to a module, if there is a set of weights which will allow the system to reduce Δ to 0 for each unit in each pattern, this rule will find it through repeated exposure to all of the members of the set of patterns.

It is important to note that the existence of a set of weights that will allow Δ to be reduced to 0 is not guaranteed, but depends on the structure inherent in the set of patterns which the model is given to learn. To be perfectly learnable by our model, the patterns must conform to the following *linear predictability constraint:*

> Over the entire set of patterns, the external input to each unit must be predictable from a linear combination of the activations of every other unit.

This is an important constraint, for there are many sets of patterns that violate it. However, it is necessary to distinguish between the patterns used inside the model, and the stimulus patterns to which human observers might be exposed in experiments, as described by psychologists. For our model to work, it is important for patterns to be assigned to stimuli in a way that will allow them to be learned.

A crucial issue, then, is the exact manner in which the stimulus patterns are encoded. As a rule of thumb, an encoding which treats each dimension or aspect of a stimulus separately is unlikely to be sufficient; what is required is a *context sensitive* encoding, such that the representation of each aspect is colored by the other aspects. For a full discussion

Table 1
Behavior of an 8-Unit Distributed Memory Module

Case	Input or response for each unit							
Pattern 1								
The Pattern:	+	−	+	−	+	+	−	−
Response to Pattern before learning	+.5	−.5	+.5	−.5	+.5	+.5	−.5	−.5
Response to Pattern after 10 learning trials	+.7	−.7	+.7	−.7	+.7	+.7	−.7	−.7
Test Input (Incomplete version of Pattern)	+	−	+	−				
Response	+.6	−.6	+.6	−.6	+.4	+.4	−.4	−.4
Test Input (Distortion of Pattern)	+	−	+	−	+	+	−	+*
Response	+.6	−.6	+.6	−.6	+.6	+.6	−.6	+.1
Pattern 2								
The Pattern:	+	+	−	−	−	+	−	+
Response to Pattern with weights learned for Pattern 1	+.5	+.5	−.5	−.5	−.5	+.5	−.5	+.5
Response to Pattern after 10 learning trials	+.7	+.7	−.7	−.7	−.7	+.7	−.7	+.7
Retest of response to Pattern 1	+.7	−.7	+.7	−.7	+.7	+.7	−.7	−.7

of this issue, see Hinton, McClelland, and Rumelhart (in press).

Decay in the increments to the weights. We assume that each trace or increment undergoes a decay process though the rate of decay of the increments is assumed to be much slower than the rate of decay of patterns of activation. Following a number of theorists (e.g., Wickelgren, 1979), we imagine that traces at first decay rapidly, but then the remaining portion becomes more and more resistant to further decay. Whether it ever reaches a point where it is no longer decaying at all we do not know. The basic effect of this assumption is that individual inputs exert large short-term effects on the weights, but after they decay the residual effect is considerably smaller. The fact that each increment has its own temporal history increases the complexity of computer simulations enormously. In all of the particular cases to be examined, we will therefore specify simplified assumptions to keep the simulations tractable.

Illustrative Examples

In this section, we describe a few illustrative examples to give the reader a feel for how we use the model, and to illustrate key aspects of its behavior.

Storage and retrieval of several patterns in a single memory module. First, we consider the storage and retrieval of two patterns in a single module of 8 units. Our basic aim is to show how several distinct patterns of activation can all be stored in the same set of weights, by what Lashley (1950) called a kind of algebraic summation, and not interfere with each other.

Before the first presentation of either pattern, we start out with all the weights set to 0. The first pattern is given at the top of Table 1. It is an arrangement of +1 and −1 inputs to the eight units in the module. (In Table 1, the 1s are suppressed in the inputs for clarity). When we present the first pattern to this module, the resulting activation values simply reflect the effects of the inputs themselves because none of the units are yet influencing any of the others.

Then, we teach the module this pattern by presenting it to the module 10 times. Each time, after the pattern of activation has had plenty of time to settle down, we adjust the weights. The next time we present the complete pattern after the 10 learning trials, the module's response is enhanced, compared with the earlier situation. That is, the activation values are increased in magnitude, owing to the combined effects of the external and internal inputs to each of the units. If we present an incomplete part of the pattern, the module can complete it; if we distort the pattern, the module tends to drive the activation back in the direction it thinks it ought to have. Of course, the magnitudes of these effects depend on parameters; but the basic nature of the effects is independent of these details.

Figure 3 shows the weights our learning procedure has assigned. Actual numerical values have been suppressed to emphasize the basic pattern of excitatory and inhibitory influences. In this example, all the numerical values are identical. The pattern of + and − signs simply gives the pattern of pairwise correlations of the elements. This is as it should be to allow pattern enhancement, completion, and noise elimination. Units which have the same activation in the pattern have positive weights, so that when one is activated it will tend to activate the other, and when one is inhibited it will tend to inhibit the other. Units which have different activations in the pattern have negative weights, so that when one is activated it will inhibit the other and vice versa.

What happens when we present a new pattern, dissimilar to the first? This is illustrated in the lower portion of Table 1. At first, the network responds to it just as though it knew nothing at all: The activations simply reflect the direct effects of the input, as they would in a module with all 0 weights. The reason is simply that the effects of the weights already in the network cancel each other out. This is a result of the fact that the two patterns are maximally dissimilar from each other. If the patterns had been more similar, there would not have been this complete cancellation of effects.

Now we learn the new pattern, presenting it 10 times and adjusting the weights each time. The resulting weights (Figure 3) represent the sum of the weights for Patterns 1 and 2. The response to the new pattern is enhanced, as shown in Table 1. The response to the old, previously learned pattern is not affected. The module will now show enhancement, completion, and noise elimination for both patterns though these properties are not illustrated in Table 1.

	Pattern 1								Pattern 2															
	+	−	+	−	+	+	−	−	+	+	−	−	−	+	−	+								
	Weights for Pattern 1								Weights for Pattern 2								Composite Weights for Both Patterns							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
1		−	+	−	+	+	−	−		+	−	−	−	+	−	+				−−		++	−−	
2	−		−	+	−	−	+	+	+		−	−	−	+	−	+			−−		−−			++
3	+	−		−	+	+	−	−	−	−		+	+	−	+	−		−−			++			−−
4	−	+	−		−	−	+	+	−	−	+		+	−	+	−	−−					−−	++	
5	+	−	+	−		+	−	−	−	−	+	+		−	+	−		−−	++					−−
6	+	−	+	−	+		−	−	+	+	−	−	−		−	+	++			−−			−−	
7	−	+	−	+	−	−		+	−	−	+	+	+	−		−	−−			++		−−		
8	−	+	−	+	−	−	+		+	+	−	−	−	+	−			++	−−		−−			

Figure 3. Weights acquired in learning Pattern 1 and Pattern 2 separately, and the composite weights resulting from learning both. (The weight in a given cell reflects the strength of the connection from the corresponding column unit to the corresponding row unit. Only the sign and relative magnitude of the weights are indicated. A blank indicates a weight of 0; + and − signify positive and negative, with a double symbol, ++ or −−, representing a value twice as large as a single symbol, + or −.

Thus, we see that more than one pattern can coexist in the same set of weights. There is an effect of storing multiple patterns, of course. When only one pattern is stored, the whole pattern (or at least, a pale copy of it) can be retrieved by driving the activation of any single unit in the appropriate direction. As more patterns are stored, larger subpatterns are generally needed to specify the pattern to be retrieved uniquely.

Learning a Prototype From Exemplars

In the preceeding section, we considered the learning of particular patterns and showed that the delta rule was capable of learning multiple patterns, in the same set of connections. In this section, we consider what happens when distributed models using the delta rule are presented with an ensemble of patterns that have some common structure. The examples described in this section illustrate how the delta rule can be used to extract the structure from an ensemble of inputs, and throw away random variability.

Let us consider the following hypothetical situation. A little boy sees many different dogs, each only once, and each with a different name. All the dogs are a little different from each other, but in general there is a pattern which represents the typical dog: each one is just a different distortion of this prototype. (We are not claiming that the dogs in the world have no more structure than this; we make this assumption for purposes of illustration only.) For now we will assume that the names of the dogs are all completely different. Given this experience, we would expect that the boy would learn the prototype of the category, even without ever seeing any particular dog which matches the prototype directly (Posner & Keele, 1968, 1970; Anderson, 1977, applies an earlier version of a distributed model to this case). That is, the prototype will seem as familiar as any of the exemplars, and he will be able to complete the pattern corresponding to the prototype from any part of it. He will not, however, be very likely to remember the names of each of the individual dogs though he may remember the most recent ones.

We model this situation with a module consisting of 24 units. We assume that the presentation of a dog produces a visual pattern of activation over 16 of the units in the hypothetical module (the 9th through 24th, counting from left to right). The name of the dog produces a pattern of activation over the other 8 units (Units 1 to 8, counting from left to right).

Each visual pattern, by assumption, is a distortion of a single prototype. The prototype used for the simulation simply had a random series of $+1$ and -1 values. Each distortion of the prototype was made by probabilistically flipping the sign of randomly selected elements of the prototype pattern. For each new distorted pattern, each element has an independent chance of being flipped, with a probability of .2. Each name pattern was simply a random sequence of $+1$s and -1s for the eight name units. Each encounter with a new dog is modeled as a presentation of a new name pattern with a new distortion of the prototype visual pattern. Fifty different trials were run, each with a new name pattern–visual pattern pair.

For each presentation, the pattern of activation is allowed to stabilize, and then the weights are adjusted as before. The increment to the weights is then allowed to decay considerably before the next input is presented. For simplicity, we assume that before the next pattern is presented, the last increment decays to a fixed small proportion of its initial value, and thereafter undergoes no further decay.

What does the module learn? The module acquires a set of weights which is continually buffeted about by the latest dog exemplar, but which captures the prototype dog quite well. Waiting for the last increment to decay to the fixed residual yields the weights shown in Figure 4.

These weights capture the correlations among the values in the prototype dog pattern quite well. The lack of exact uniformity is due to the more recent distortions presented, whose effects have not been corrected by subsequent distortions. This is one way in which the model gives priority to specific exemplars, especially recent ones. The effects of recent exemplars are particularly strong, of course, before they have had a chance to decay. The module can complete the prototype quite well, and it will respond more strongly to the prototype than to any distortion of it. It has, however, learned no particular relation between this prototype and any name pattern, because a totally different random association was presented on each trial. If the pattern of activation on the name units had been the same in every case (say, each dog was just called *dog*), or even in just a reasonable fraction of the cases, then the module would have been able to retrieve this shared name pattern from the prototype of the visual pattern and the prototype pattern from the name.

Multiple, nonorthogonal prototypes. In the preceeding simulation we have seen how the distributed model acts as a sort of signal averager, finding the central tendency of a set of related patterns. In and of itself this is an important property of the model, but the importance of this property increases when we realize that the model can average several different patterns in the same composite memory trace. Thus, several different prototypes can be stored in the same set of weights. This is important, because it means that the model does not fall into the trap of needing to decide which category to put a pattern in before knowing which prototype to average it with. The acquisition of the different prototypes proceeds without any sort of explicit

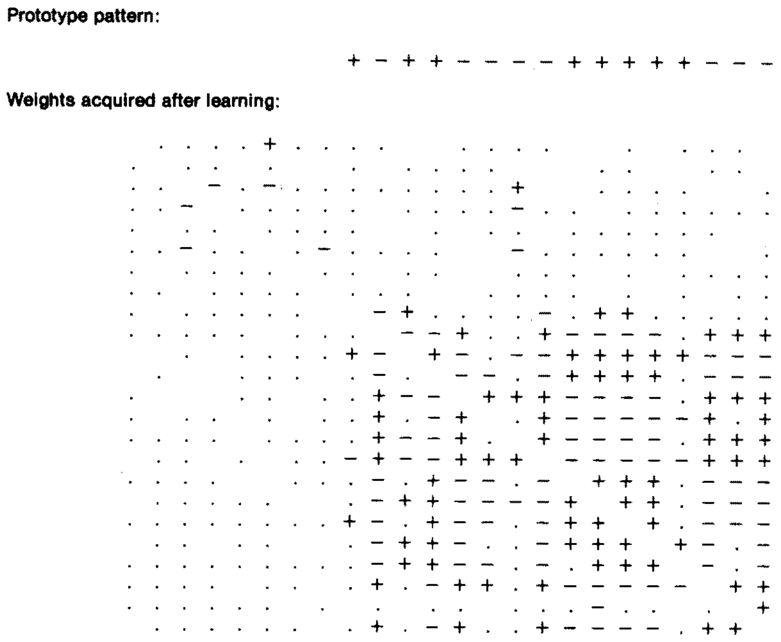

Figure 4. Weights acquired in learning from distorted exemplars of a prototype. (The prototype pattern is shown above the weight matrix. Blank entries correspond to weights with absolute values less than .01; dots correspond to absolute values less than .06; pluses or minuses are used for weights with larger absolute values.)

categorization. If the patterns are sufficiently dissimilar, there is no interference among them at all. Increasing similarity leads to increased confusability during learning, but eventually the delta rule finds a set of connection strengths that minimizes the confusability of similar patterns.

To illustrate these points, we created a simulation analog of the following hypothetical situation. Let us say that our little boy sees, in the course of his daily experience, different dogs, different cats, and different bagels. First, let's consider the case in which each experience with a dog, a cat, or a bagel is accompanied by someone saying *dog, cat,* or *bagel,* as appropriate.

The simulation analog of this situation involved forming three *visual* prototype patterns of 16 elements, two of them (the one for dog and the one for cat) somewhat similar to each other (r = .5), and the third (for the bagel) orthogonal to both of the other two. Paired with each visual pattern was a name pattern of eight elements. Each name pattern was orthogonal to both of the others. Thus, the prototype visual pattern for cat and the prototype visual pattern for dog were similar to each other, but their names were not related.

Stimulus presentations involved presentations of distorted exemplars of the name–visual pattern pairs to a module of 24 elements like the one used in the previous simulation. This time, both the name pattern and the visual pattern were distorted, with each element having its sign flipped with an independent probability of .1 on each presentation. Fifty different distortions of each name–visual pattern pair were presented in groups of three consisting of one distortion of the dog pair, one distortion of the cat pair, and one distortion of the bagel pair. Weight adjustment occurred after each presentation, with decay to a fixed residual before each new presentation.

At the end of training, the module was tested by presenting each name pattern and observing the resulting pattern of activation over the visual nodes, and by presenting each visual pattern and observing the pattern of activation over the name nodes. The results are shown in Table 2. In each case, the model

Table 2
Results of Tests After Learning the Dog, Cat, and Bagel Patterns

	Input or response for each unit																							
Case	Name units								Visual pattern units															
Pattern for dog prototype	+	−	+	−	+	−	+	−	+	−	+	+	−	−	−	−	+	+	+	+	+	−	−	−
Response to dog name									+3	−4	+4	+4	−4	−4	−4	−4	+4	+4	+4	+3	+4	−4	−4	−3
Response to dog visual pattern	+5	−4	+4	−5	+5	−4	+4	−4																
Pattern for cat prototype	+	+	−	−	+	+	−	−	+	−	+	+	−	−	−	−	+	−	+	−	+	+	−	+
Response to cat name									+4	−3	+4	+4	−4	−3	−3	−4	+4	−4	+4	−4	+4	+4	−4	+4
Response to cat visual pattern	+5	+4	−4	−5	+4	+4	−4	−4																
Pattern for bagel prototype	+	−	−	+	+	−	−	+	+	+	−	+	−	+	+	−	+	−	−	+	+	+	+	−
Response to bagal name									+3	+4	−4	+4	−4	+4	+4	−4	+4	−4	−4	+4	+3	+4	+4	−4
Response to bagel visual pattern	+4	−4	−4	+4	+4	−4	−4	+4																

Note. Decimal points have been suppressed for clarity; thus, an entry of +4 represents an activation value of +.4.

reproduces the correct completion for the probe, and there is no apparent contamination of the cat pattern by the dog pattern, even though the visual patterns are similar to each other.

In general, pattern completion is a matter of degree. One useful measure of pattern completion is the dot product of the pattern of activation over the units with the pattern of external inputs to the units. Because we treat the external inputs as +1s and −1s, and because the activation of each node can only range from +1 to −1, the largest possible value the dot product can have is 1.0. We will use this measure explicitly later when considering some simulations of experimental results. For getting an impression of the degree of pattern reinstatement in the present cases, it is sufficient to note that when the sign of all of the elements is correct, as it is in all of the completions in Table 2, the average magnitude of the activations of the units corresponds to the dot product.

In a case like the present one, in which some of the patterns known to the model are correlated, the values of the connection strengths that the model produces do not necessarily have a simple interpretation. Though their sign always corresponds to the sign of the correlation between the activations of the two units, their magnitude is not a simple reflection of the magnitude of their correlation, but is influenced by the degree to which the model is relying on this particular correlation to predict the activation of one node from the others. Thus, in a case where two nodes (call them *i* and *j*) are perfectly correlated, the strength of the connection from *i* to *j* will depend on the number of other nodes whose activations are correlated with *j*. If *i* is the only node correlated with *j*, it will have to do all the work of predicting *j*, so the weight will be very strong; on the other hand, if many nodes besides *i* are correlated with *j*, then the work of predicting *j* will be spread around, and the weight between *i* and *j* will be considerably smaller. The weight matrix acquired as a result of learning the dog, cat, and bagel patterns (Figure 5) reflects these effects. For example, across the set of three prototypes, Units 1 and 5 are perfectly correlated, as are Units 2 and 6. Yet the connection from 2 to 5 is stronger than the connection from 1 to 4 (these connections are *d in Figure 5). The reason for the difference is that 2 is one of only three units which correlate perfectly with 5, whereas Unit 1 is one of seven units which correlate perfectly with 4. (In Figure 5, the weights do not reflect these contrasts perfectly in every case, because the noise introduced into the learning happens by chance to alter some of the correlations present in the prototype patterns. Averaged over time, though, the weights will conform to their expected values.)

Thus far we have seen that several prototypes, not necessarily orthogonal, can be stored in the same module without difficulty. It is true, though we do not illustrate it, that the model has more trouble with the cat and dog visual patterns earlier on in training, before learning has essentially reached asymptotic levels as it has by the end of 50 cycles through the full set of patterns. And, of course, even at the end of learning, if we present as a probe a part of the visual pattern, if it does not differentiate between the dog and the cat, the model will produce a blended response. Both these aspects of the model seem generally consistent with what we should expect from human subjects.

Category learning without labels. An important further fact about the model is that it can learn several different visual patterns, even without the benefit of distinct identifying name patterns during learning. To demonstrate this we repeated the previous simulation, simply replacing the name patterns with 0s. The model still learns about the internal structure of the visual patterns, so that, after 50 cycles through the stimuli, any unique subpart of any one of the patterns is sufficient to reinstate to the rest of the corresponding

Prototypes:																								
Dog:	+	−	+	−	+	−	+	−	+	−	+	+	−	−	−	−	+	+	+	+	+	−	−	−
Cat:	+	+	−	−	+	+	−	−	+	−	+	+	−	−	−	−	+	−	+	−	+	+	−	+
Bagel:	+	−	−	+	+	−	−	+	+	+	−	+	−	+	+	−	+	−	−	+	+	+	+	−
Weights:																								
		−1	−1		+2•	−1		−1	+3			+5	−2	−1		−3	+2	−1			+2			−1
			−2		−1	+4•			+1	−1				−1	−1	+1	−1	−1		−3	−1		−3	+3
	−1	−2		−1			+5				+1		+1	−1	−1		−1	+4		+1		−5		−1
			−1		−1		−1	+2	−1	+3	−3			+2	+3			−1	−2		−1	+1	+3	
	+3	−1	−1					+1		−1		+3	−3			−2	+2			+1	+3	+1	−1	−1
	−1	+5	−3				−2		−1	−1		−1		−1			−1	−2	+1	−5	−1	+2	−1	+5
	−1		+3	−1	−1	−2					+1			−1		+1		+4			−1	−2		
	−1	−1	−1	+2			−1		−1	+3	−2	−1		+3	+3		−1	−1	−3	+1	−1	+3	+3	
	+3		−1	−1	+2	−1	−1					+3	−2			−3	+2		+1		+3	+1		−1
	−1	−1	−1	+3		−1		+3			−3	−1		+3	+3	+1	−1	−2	−2	+1	−1		+2	
	+1	+1	+1	−3				−2	+1	−3			−1	−3	−3			+1	+3	−1		−1	−2	+2
	+2	−1	−1	−1	+3		−1		+3	−1	+1		−2	−1		−2	+2	−1			+2	+1	−1	
	−2	+1	+1		−2	+1	+1	+1	−3			−3			+1	+2	−3			−1	−2	−1	+1	
		−1		+2	−1			+2	−1	+2	−2		+1		+2	+1	−1		−3		−1		+2	
	−1	−1	−1	+2		−1	−1	+2		+2	−3			+3.		+1			−3		−1		+3	
	−2	+1	+1	+1	−3		+1	+1	−2	+1	−1	−3	+3				−2	+1	−1	−1	−3	−1	+1	
	+3		−1		+2		−1		+2			+3	−3			−3		−1	+1		+3	+1	−1	−1
			+5	−1	−1	−1	+4	−1					+1		−3	+1			+1		−1	−5	−1	−1
		+1		−2		+1	+1	−3	+1	−2	+2	+1	−1	−3	−3	−1		+1		−2			−3	+1
	+1	−5	+2	+1	+1	−5	+3				−2	+1	−1	+1			+1	+1	−2		+1	−2	+1	−3
	+2	−1	−1		+3	−1			+2		+1	+2	−2			−3	+2		+1					
		+3	−4	+1		+3	−4	+1	+1		−1			+1	+1		+1	−5	−1	−3			+1	+2
				+2	−1		−1		−1	+2	−3			+2	+2			−1	−3			+1		
	−1	+3	−2	−1	−1	+5	−3			−1	+3		+1	−1			−1		+1	−2		+3	−1	

Figure 5. Weights acquired in learning the three prototype patterns shown. (Blanks in the matrix of weights correspond to weights with absolute values less than or equal to .05. Otherwise the actual value of the weight is about .05 times the value shown; thus +5 stands for a weight of +.25. The gap in the horizontal and vertical dimensions is used to separate the name field from the visual pattern field.)

Table 3
Results of Tests After Learning The Dog, Cat, and Bagel Patterns Without Names

Case	Input or response for each visual unit															
Dog visual pattern	+	−	+	+	−	−	−	−	+	+	+	+	+	−	−	−
Probe									+	+	+	+				
Response	+3	−3	+3	+3	−3	−4	−3	−3	+6	+5	+6	+5	+3	−2	−3	−2
Cat visual pattern	+	−	+	+	−	−	−	−	+	−	+	−	+	+	−	+
Probe									+	−	+	−				
Response	+3	−3	+3	+3	−3	−3	−3	−3	+6	−5	+6	−5	+3	+2	−3	+2
Bagel visual pattern	+	+	−	+	−	+	+	−	+	−	−	+	+	+	+	−
Probe									+	−	−	+				
Response	+2	+3	−4	+3	−3	+3	+3	−3	+6	−6	−6	+6	+3	+3	+3	−3

pattern correctly. This aspect of the model's behavior is illustrated in Table 3. Thus, we have a model that can, in effect, acquire a number of distinct categories, simply through a process of incrementing connection strengths in response to each new stimulus presentation. Noise, in the form of distortions in the patterns, is filtered out. The model does not require a name or other guide to distinguish the patterns belonging to different categories.

Coexistence of the prototype and repeated exemplars. One aspect of our discussion up to this point may have been slightly misleading. We may have given the impression that the model is simply a prototype extraction device. It is more than this, however; it is a device that captures whatever structure is present in a set of patterns. When the set of patterns has a prototype structure, the model will act as though it is extracting prototypes; but when it has a different structure, the model will do its best to accommodate this as well. For example, the model permits the coexistence of representations of prototypes with representations of particular, repeated exemplars.

Consider the following situation. Let us say that our little boy knows a dog next door named Rover and a dog at his grandma's house named Fido. And let's say that the little boy goes to the park from time to time and sees dogs, each of which his father tells him is a dog.

The simulation-analog of this involved three different eight-element name patterns, one for Rover, one for Fido, and one for Dog. The visual pattern for Rover was a particular randomly generated distortion of the dog prototype pattern, as was the visual pattern for Fido. For the dogs seen in the park, each one was simply a new random distortion of the prototype. The probability of flipping the sign of each element was again .2. The learning regime was otherwise the same as in the dog–cat–bagel example.

At the end of 50 learning cycles, the model was able to retrieve the visual pattern corresponding to either repeated exemplar (see Table 4) given the associated name as input. When given the Dog name pattern as input, it retrieves the prototype visual pattern for dog. It can also retrieve the appropriate name from each of the three visual patterns. This is true, even though the visual pattern for Rover differs from the visual pattern for dog by only a single element. Because of the special importance of this particular element, the weights from this element to the units that distinguish Rover's name pattern from the prototype name pattern are quite strong. Given part of a visual pattern, the model will complete it; if the part corresponds to the prototype, then that is what is completed, but if it corresponds to one of the repeated exemplars, that exemplar is completed. The model, then, knows both the prototype and the repeated exemplars quite well. Several other sets of prototypes and their repeated exemplars could also be stored in the same module, as long as its capacity is not exceeded; given large numbers of units per module, a lot of different patterns can be stored.

Let us summarize the observations we

Table 4
Results of Tests with Prototype and Specific Exemplar Patterns

	Input or response for each unit																							
Case	Name units								Visual pattern units															
Pattern for dog prototype	+	−	+	−	+	−	+	−	+	−	+	+	−	−	−	−	+	+	+	+	+	−	−	−
Response to prototype name									+4	−5	+3	+3	−4	−3	−3	−3	+3	+3	+4	+3	+4	−3	−4	−4
Response to prototype visual pattern	+5	−4	+4	−4	+5	−4	+4	−4																
Pattern for "Fido" exemplar	+	−	−	−	+	−	−	−	+	−	(−)	+	−	−	−	−	+	+	+	+	+	(+)	−	−
Response to Fido name									+4	−4	−4	+4	−4	−4	−4	−4	+4	+4	+4	+4	+4	+4	−4	−4
Response to Fido visual pattern	+5	−5	−3	−5	+4	−5	−3	−5																
Pattern for "Rover" exemplar	+	−	−	+	+	+	−	+	+	(+)	+	+	−	−	−	−	+	+	+	+	+	−	−	−
Response to Rover name									+4	+5	+4	+4	−4	−4	−4	−4	+4	+4	+4	+4	+4	−4	−4	−4
Response to Rover visual pattern	+4	−4	−2	+4	+4	+4	−2	+4																

have made in these several illustrative simulations. First, our distributed model is capable of storing not just one but a number of different patterns. It can pull the central tendency of a number of different patterns out of the noisy inputs; it can create the functional equivalent of perceptual categories with or without the benefit of labels; and it can allow representations of repeated exemplars to coexist with the representation of the prototype of the categories they exemplify in the same composite memory trace. The model is not simply a categorizer or a prototyping device; rather, it captures the structure inherent in a set of patterns, whether it be characterizable by description in terms of prototypes or not.

The ability to retrieve accurate completions of similar patterns is a property of the model which depends on the use of the delta learning rule. This allows both the storage of different prototypes that are not completely orthogonal and the coexistence of prototype representations and repeated exemplars.

Simulations of Experimental Results

Up to this point, we have discussed our distributed model in general terms and have outlined how it can accommodate both abstraction and representation of specific information in the same network. We will now consider, in the next two sections, how well the model does in accounting for some recent evidence about the details of the influence of specific experiences on performance.

Repetition and Familiarity Effects

When we perceive an item—say a word, for example—this experience has effects on our later performance. If the word is presented again, within a reasonable interval of time, the prior presentation makes it possible for us to recognize the word more quickly, or from a briefer presentation.

Traditionally, this effect has been interpreted in terms of units that represent the presented items in memory. In the case of word perception, these units are called *word detectors* or *logogens,* and a model of repetition effects for words has been constructed around the logogen concept (Morton, 1979). The idea is that the threshold for the logogen is reduced every time it *fires* (that is, every time the word is recognized), thereby making it easier to fire the logogen at a later time. There is supposed to be a decay of this priming effect, with time, so that eventually the effect of the first presentation wears off.

This traditional interpretation has come under serious question of late, for a number of reasons. Perhaps paramount among the reasons is the fact that the exact relation between the specific context in which the priming event occurs and the context in which the test event occurs makes a huge difference (Jacoby, 1983a, 1983b). Generally speaking, nearly any change in the stimulus—from spoken to printed, from male speaker to female speaker, and so forth—tends to reduce the magnitude of the priming effect.

These facts might easily be taken to support the enumeration of specific experiences view, in which the logogen is replaced by the entire ensemble of experiences with the word, with each experience capturing aspects of the specific context in which it occurred. Such a view has been championed most strongly by Jacoby (1983a, 1983b).

Our distributed model offers an alternative interpretation. We see the traces laid down by the processing of each input as contributing to the composite, superimposed memory representation. Each time a stimulus is processed, it gives rise to a slightly different memory trace: either because the item itself is different or because it occurs in a different context that conditions its representation. The logogen is replaced by the set of specific traces, but the traces are not kept separate. Each trace contributes to the composite, but the characteristics of particular experiences tend nevertheless to be preserved, at least until they are overridden by cancelling characteristics of other traces. And the traces of one stimulus pattern can coexist with the traces of other stimuli, within the same composite memory trace.

It should be noted that we are not faulting either the logogen model or models based on the enumeration of specific experiences for their physiological implausibility here, because these models are generally not stated in physiological terms, and their authors might reasonably argue that nothing in their models precludes distributed storage at a physiological level. What we are suggesting is that a model which proposes explicitly distributed, superpositional storage can account for the kinds of findings that logogen models have been proposed to account for, as well as other findings which strain the utility of the concept of the logogen as a psychological construct. In the discussion section we will consider ways in which our distributed model differs from enumeration models as well.

To illustrate the distributed model's account of repetition priming effects, we carried out the following simulation experiment. We made up a set of eight random vectors, each 24 elements long, each one to be thought of as the prototype of a different recurring stimulus pattern. Through a series of 10 training cycles using the set of eight vectors, we constructed a composite memory trace. During training, the model did not actually see the prototypes, however. On each training presentation it saw a new random distortion of one of the eight prototypes. In each of the distortions, each of the 24 elements had its value flipped with a probability of .1. Weights were adjusted after every presentation, and then allowed to decay to a fixed residual before the presentation of the next pattern.

The composite memory trace formed as a result of the experience just described plays the same role in our model that the set of logogens or detectors play in a model like Morton's or, indeed, the interactive activation model of word preception. That is, the trace contains information which allows the model to enhance perception of familiar patterns, relative to unfamiliar ones. We demonstrate this by comparing the activations resulting from the processing of subsequent presentations of new distortions of our eight familiar patterns with other random patterns with which the model is not familiar. The pattern of activation that is the model's response to the input is stronger, and grows to a particular level more quickly, if the stimulus is a new distortion of an old pattern than if it is a new pattern. We already observed this general enhanced response to exact repetitions of familiar patterns in our first example (see Table 1). Figure 6 illustrates that the effect also applies to new distortions of old patterns, as compared with new patterns, and illustrates how the activation process proceeds over successive time cycles of processing.

Pattern activation and response strength. The measure of activation shown in the Figure 6 is the dot product of the pattern of activation over the units of the module times the stim-

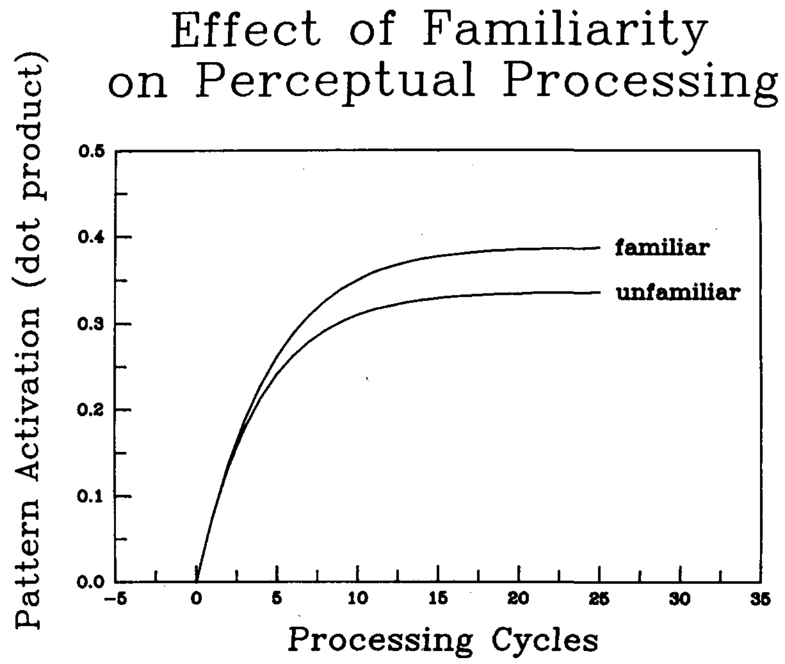

Figure 6. Growth of the pattern of activation for new distortions of familiar and unfamiliar patterns. (The measure of the strength of the pattern of activation is the dot product of the response pattern with the input vector. See text for an explanation.)

ulus pattern itself, normalized for the number n of elements in the pattern: For the pattern j we call this expression α_j. The expression α_j represents the degree to which the actual pattern of activation on the units captures the input pattern. It is an approximate analog to the activation of an individual unit in models which allocate a single unit to each whole pattern.

To relate these pattern activations to response probabilities, we must assume that mechanisms exist for translating patterns of activation into overt responses measurable by an experimenter. We will assume that these mechanisms obey the principles stated by McClelland and Rumelhart (1981) in the interactive activation model of word perception, simply replacing the activations of particular units with the α measure of pattern activation.

In the interactive activation model, the probability of choosing the response appropriate to a particular unit was based on an exponential transform of a time average of the activation of the unit. This quantity, called the *strength* of the particular response, was divided by the total strength of all alternatives (including itself) to find the response probability (Luce, 1963). One complication arises because of the fact that it is not in general possible to specify exactly what the set of alternative responses might be for the denominator. For this reason, the strengths of other responses are represented by a constant C (which stands for the competition). Thus, the expression for probability of choosing the response appropriate to pattern j is just $p(r_j) = e^{k\bar{\alpha}_j}/(C + e^{k\bar{\alpha}_j})$, where $\bar{\alpha}_j$ represents the time average of α_j, and k is a scaling constant.

These assumptions finesse an important issue, namely the mechanism by which a pattern of activation give rise to a particular response. A detailed discussion of this issue will appear in Rumelhart and McClelland (in press). For now, we wish only to capture basic properties any actual response selection mechanism must have: It must be sensitive to the input pattern, and it must approximate other basic aspects of response selection behavior captured by the Luce (1963) choice model.

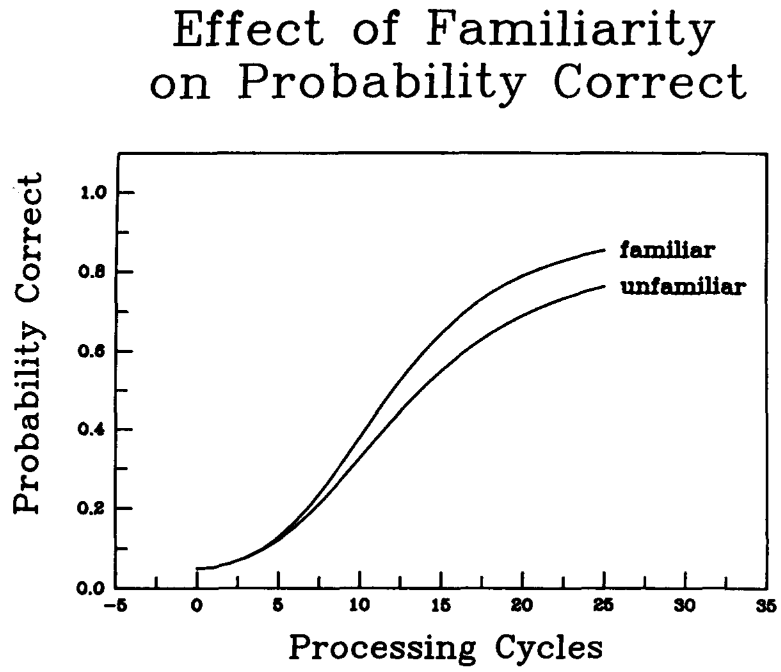

Figure 7. Simulated growth of response accuracy over the units in a 24-unit module, as a function of processing cycles, for new distortions of previously learned patterns compared with new distortions of patterns not previously learned.

Effects of experimental variables on time-accuracy curves. Applying the assumptions just described, we can calculate probability of correct response as a function of processing cycles for familiar and unfamiliar patterns. The result, for a particular choice of scaling parameters, is shown in Figure 7. If we assume performance in a perceptual identification task is based on the height of the curve at the point where processing is cut off by masking (McClelland & Rumelhart, 1981), then familiarity would lead to greater accuracy of perceptual identification at a given exposure duration. In a reaction time task, if the response is emitted when its probability reaches a particular threshold activation value (McClelland, 1979), then familiarity would lead to speeded responses. Thus, the model is consistent with the ubiquitous influence of familiarity both on response accuracy and speed, in spite of the fact that it has no detectors for familiar stimuli.

But what about priming and the role of congruity between the prime event and the test event? To examine this issue, we carried out a second experiment. Following learning of eight patterns as in the previous experiment, new distortions of half of the random vectors previously learned by the model were presented as primes. For each of these primes, the pattern of activation was allowed to stabilize, and changes in the strengths of the connections in the model were then made. We then tested the model's response to (a) the same four distortions; (b) four new distortions of the same patterns; and (c) distortions of the four previously learned patterns that had not been presented as primes. There was no decay in the weights over the course of the priming experiment; if decay had been included, its main effect would have been to reduce the magnitude of the priming effects.

The results of the experiment are shown in Figure 8. The response of the model is greatest for the patterns preceded by identical primes, intermediate for patterns preceded by similar primes, and weakest for patterns not preceded by any related prime.

Our model, then, appears to provide an account, not only for the basic existence of priming effects, but also for the graded nature of priming effects as a function of congruity

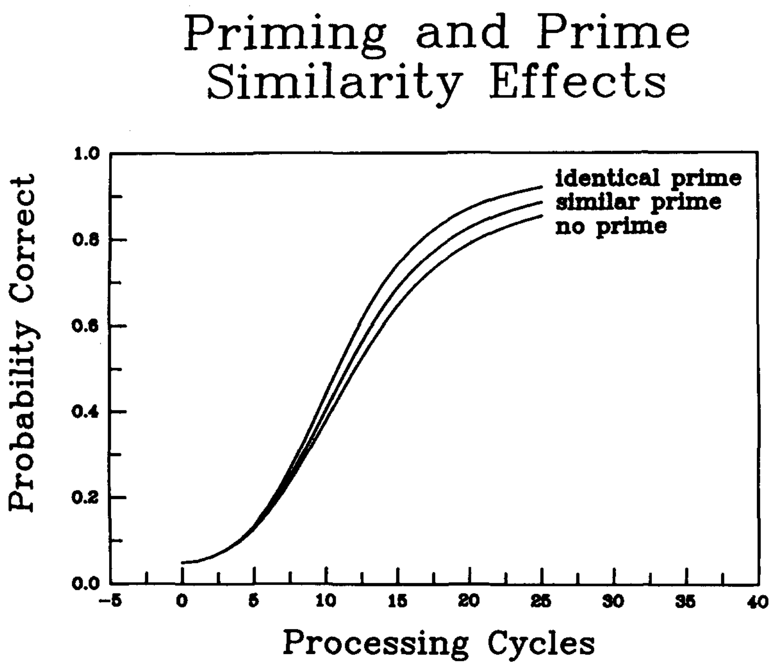

Figure 8. Response probability as a function of exposure time for patterns preceded by identical primes, similar primes, or no related prime.

between prime event and test event. It avoids the problem of multiplication of context-specific detectors which logogen theories fall prey to, while at the same time avoiding enumeration of specific experiences. Congruity effects are captured in the composite memory trace.

The model also has another advantage over the logogen view. It accounts for repetition priming effects for unfamiliar as well as familiar stimuli. When a pattern is presented for the first time, a trace is produced just as it would be for stimuli that had previously been presented. The result is that, on a second presentation of the same pattern, or a new distortion of it, processing is facilitated. The functional equivalent of a logogen begins to be established from the very first presentation.

To illustrate the repetition priming of unfamiliar patterns and to compare the results with the repetition priming we have already observed for familiar patterns, we carried out a third experiment. This time, after learning eight patterns as before, a priming session was run in which new distortions of four of the familiar patterns and distortions of four new patterns were presented. Then, in the test phase, 16 stimuli were presented: New distortions of the primed, familiar patterns; new distortions of the unprimed, familiar patterns; new distortions of the primed, previously unfamiliar patterns; and finally, new distortions of four patterns that were neither primed nor familiar. The results are shown in Figure 9. What we find is that long-term familiarity and recent priming have approximately additive effects on the asymptotes of the time-accuracy curves. The time to reach any given activation level shows a mild interaction, with priming having slightly more of an effect for unfamiliar than for familiar stimuli.

These results are consistent with the bulk of the findings concerning the effects of preexperimental familiarity and repetition in the recent series of experiments by Feustel, Shiffrin, and Salasoo (1983) and Salasoo et al. (1985). They found that preexperimental familiarity of an item (word vs. nonword) and prior exposure had this very kind of interactive effect on exposure time required for accurate identification of all the letters of a string, at least when words and nonwords

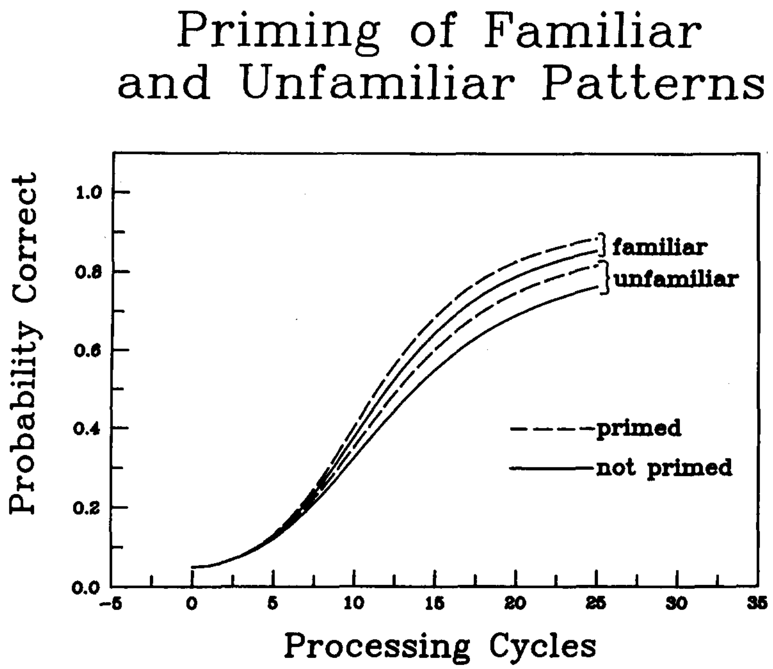

Figure 9. Response to new distortions of primed, familiar patterns, unprimed, familiar patterns, primed, unfamiliar patterns, and unprimed, unfamiliar patterns.

were mixed together in the same lists of materials.

A further aspect of the results reported by Salasoo, Shiffrin, and Feustel is also consistent with our approach. In one of their experiments, they examined threshold for accurate identification as a function of number of prior presentations, for both words and pseudowords. Although thresholds were initially elevated for pseudowords, relative to words, there was a rather rapid convergence of the thresholds over repeated presentations, with the point of convergence coming at about the same place on the curve for two different versions of their perceptual identification task. (Salasoo et al., 1985, Figure 7.) Our model, likewise, shows this kind of convergence effect, as illustrated in Figure 10.

The Feustel et al. (1983) and Salasoo et al. (1985) experiments provide very rich and detailed data that go beyond the points we have extracted from them here. We do not claim to have provided a detailed account of all aspects of their data. However, we simply wish to note that the general form of their basic findings is consistent with a model of the distributed type. In particular, we see no reason to assume that the process by which unfamiliar patterns become familiar involves the formation of an abstract, logogenlike unit separate from the episodic traces responsible for repetition priming effects.

There is one finding by Salasoo et al. (1985) that appears to support the view that there is some special process of unit formation that is distinct from the priming of old units. This is the fact that after a year between training and testing, performance with pseudowords used during training is indistinguishable from performance with words, but performance with words used during training shows no residual benefit compared with words not previously used. The data certainly support the view that training experience made the pseudowords into lasting perceptual units, at the same time that is produced transitory priming of existing units. We have not attempted to account for this finding in detail, but we doubt that it is inconsistent with a distributed model. In support of this, we offer one reason why repetition effects might seem to persist longer for pseudowords rather than for words in the Salasoo et al. experiment. For pseudowords, a strong asso-

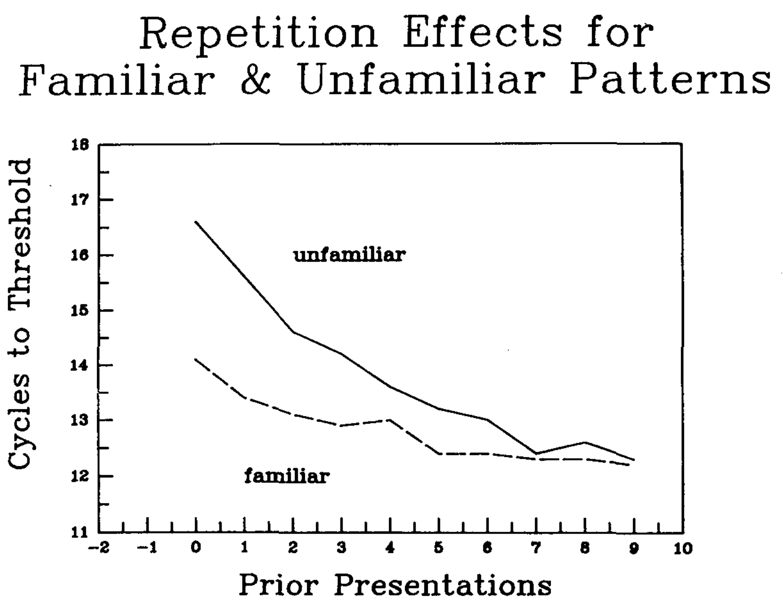

Figure 10. Time to reach a fixed-accuracy criterion (60% correct) for previously familiar and unfamiliar patterns, as a function of repetitions.

ciation would be built up between the item and the learning context during initial training. Such associations would be formed for words, but because these stimuli have been experienced many times before and have already been well learned, smaller increments in connection strength are formed for these stimuli during training, and thus the strength of the association between the item and the learning context would be less. If this view is correct, we would expect to see a disadvantage for pseudowords relative to words if the testing were carried out in a situation which did not reinstate the mental state associated with the original learning experience, because for these stimuli much of what was learned would be tied to the specific learning context: such a prediction would appear to differentiate our account from any view which postulated the formation of an abstract, context-independent logogen as the basis for the absence of a pseudoword decrement effect.

Representation of General and Specific Information

In the previous section, we cast our distributed model as an alternative to the view that familiar patterns are represented in memory either by separate detectors or by an enumeration of specific experiences. In this section, we show that the model provides alternatives to both abstraction and enumeration models of learning from exemplars of prototypes.

Abstraction models were originally motivated by the finding that subjects occasionally appeared to have learned better how to categorize the prototype of a set of distorted exemplars than the specific exemplars they experienced during learning (Posner & Keele, 1968). However, pure abstraction models have never fared very well, because there is nearly always evidence of some superiority of the particular training stimuli over other stimuli equally far removed from the prototype. A favored model, then, is one in which there is both abstraction and memory for particular training stimuli.

Recently, proponents of models involving only enumeration of specific experiences have noted that such models can account for the basic fact that abstraction models are primarily designed to account for—enhanced response to the prototype, relative to particular previously seen exemplars, under some conditions—as well as failures to obtain such effects under other conditions (Hintzman, 1983; Medin & Shaffer, 1978). In evaluating distributed models, it is important to see if they can do as well. Anderson (1977) has

Table 5
Schematic Description of Stimulus Sets Used in Simulations of Whittlesea's Experiments

	Stimulus set						
Prototype	Ia	Ib	IIa	IIb	IIc	III	V
PPPPP	APPPP	BPPPP	ABPPP	ACPPP	APCPP	ABCPP	CCCCC
	PAPPP	PBPPP	PABPP	PACPP	PAPCP	PABCP	CBCBC
	PPAPP	PPBPP	PPABP	PPACP	PPAPC	PPABC	BCACB
	PPPAP	PPPBP	PPPAB	PPPAC	CPPAP	CPPAB	ABCBA
	PPPPA	PPPPB	BPPPA	CPPPA	PCPPA	BCPPA	CACAC

Note. The actual stimuli used can be filled in by replacing P with +−+−; A with ++−−; B with +−−+; and C with ++++. The model is not sensitive to the fact the same subpattern was used in each of the five slots.

made important steps in this direction, and Knapp and Anderson (1984) have shown how their distributed model can account for many of the details of the Posner–Keele experiments. Recently, however, two sets of findings have been put forward which appear to strongly favor the enumeration of specific experiences view, at least relative to pure abstraction models. It is important, therefore, to see how well our distributed model can do in accounting for these kinds of effects.

The first set of findings comes from a set of studies by Whittlesea (1983). In a large number of studies, Whittlesea demonstrated a role for specific exemplars in guiding performance on a perceptual identification task. We wanted to see whether our model would demonstrate a similar sensitivity to specific exemplars. We also wanted to see whether our model would account for the conditions under which such effects are not obtained.

Whittlesea used letter strings as stimuli. The learning experiences subjects received involved simply looking at the stimuli one at a time on a visual display and writing down the sequence of letters presented. Subjects were subsequently tested for the effect of this training on their ability to identify letter strings bearing various relations to the training stimuli and to the prototypes from which the training stimuli were derived. The test was a perceptual identification task; the subject was simply required to try to identify the letters from a brief flash.

The stimuli Whittlesea used were all distortions of one of two prototype letter strings. Table 5 illustrates the essential properties of the sets of training and test stimuli he used. The stimuli in Set Ia were each one step away from the prototype. The Ib items were also one step from the prototype and one step from one of the Ia distortions. The Set IIa stimuli were each two steps from the prototype, and one step from a particular Ia distortion. The Set IIb items were also two steps from the prototype, and each was one step from one of the IIa distortions. The Set IIc distortions were two steps from the prototype also, and each was two steps from the closest IIa distortion. Over the set of five IIc distortions, the A and B subpatterns each occurred once in each position, as they did in the case of the IIa distortions. The distortions in Set III were three steps from the prototype, and one step from the closest member of Set IIa. The distortions in Set V were each five steps from the prototype.

Whittlesea ran seven experiments using different combinations of training and test stimuli. We carried out simulation analogs of all of these experiments, plus one additional experiment that Whittlesea did not run. The main difference between the simulation experiments and Whittlesea's actual experiments was that he used two different prototypes in each experiment, whereas we only used one.

The simulation used a simple 20-unit module. The set of 20 units was divided into five submodules, one for each letter in Whittlesea's letter strings. The prototype pattern and the different distortions used can be derived from the information provided in Table 5.

Each simulation experiment began with null connections between the units. The training phase involved presenting the set or sets of training stimuli analogous to those Whittlesea used, for the same number of presentations. To avoid idiosyncratic effects of particular orders of training stimuli, each experiment was run six times, each with a different random order of training stimuli. On each trial, activations were allowed to settle down through 50 processing cycles, and then connection strengths were adjusted. There was no decay of the increments to the weights over the course of an experiment.

In the test phase, the model was tested with the sets of test items analogous to the sets Whittlesea used. As a precaution against effects of prior test items on performance, we simply turned off the adjustment of weights during the test phase.

A summary of the training and test stimuli used in each of the experiments, of Whittlesea's findings, and of the simulation results are shown in Table 6. The numbers represent relative amounts of enhancement in performance as a result of the training experience, relative to a pretest baseline. For Whittlesea's data, this is the per letter increase in letter identification probability between a pre- and posttest. For the simulation, it is the increase in the size of the dot product for a pretest with null weights and a posttest after training. For comparability to the data, the dot product difference scores have been doubled. This is simply a scaling operation to facilitate qualitative comparison of experimental and simulation results.

A comparison of the experimental and simulation results shows that wherever there is a within-experiment difference in Whittlesea's data, the simulation produced a difference in the same direction. (Between experiment comparisons are not considered because of subject and material differences which renders such differences unreliable.) The next several paragraphs review some of the major findings in detail.

Some of the comparisons bring out the importance of congruity between particular test and training experiences. Experiments 1, 2, and 3 show that when distance of test stimuli from the prototype is controlled, similarity to particular training exemplars makes a difference both for the human subject and in the model. In Experiment 1, the relevant contrast was between Ia and Ib items. In Experiment 2, it was between IIa and IIc items. Experiment 3 shows that the subjects and the model both show a gradient in performance with increasing distance of the test items from the nearest old exemplar.

Experiments 4, 4′, and 5 examine the status of the prototype and other test stimuli closer to the prototype than any stimuli actually shown during training. In Experiment 4, the training stimuli were fairly far away from the prototype, and there were only five different training stimuli (the members of the IIa set). In this case, controlling for distance from the nearest training stimuli, test stimuli closer to the prototype showed more enhancement than those farther away (Ia vs. III comparison). However, the actual training stimuli nevertheless had an advantage over both other sets of test stimuli, including those that were closer to the prototype than the training stimuli themselves (IIa vs. Ia comparison).

In Experiment 4′ (not run by Whittlesea) the same number of training stimuli were used as in Experiment 4, but these were

Table 6
Summary of Perceptual Identification Experiments With Experimental and Simulation Results

Whittlesea's experiment	Training stimulus set(s)	Test stimulus sets	Experimental results	Simulation results
1	Ia	Ia	.27	.24
		Ib	.16	.15
		V	.03	−.05
2	IIa	IIa	.30	.29
		IIc	.15	.12
		V	.03	−.08
3	IIa	IIa	.21	.29
		IIb	.16	.14
		IIc	.10	.12
4	IIa	P	—	.24
		Ia	.19	.21
		IIa	.23	.29
		III	.15	.15
4′	Ia	P	—	.28
		Ia	—	.24
		IIa	—	.12
5	IIa, b, c	P	—	.25
		Ia	.16	.21
		IIa	.16	.18
		III	.10	.09
6	III	Ia	.16	.14
		IIa	.16	.19
		III	.19	.30
7	IIa	IIa	.24	.29
		IIc	.13	.12
		III	.17	.15

closer to the prototype. The result is that the simulation shows an advantage for the prototype over the old exemplars. The specific training stimuli used even in this experiment do influence performance, however, as Whittlesea's first experiment (which used the same training set) shows (Ia–Ib contrast). This effect holds both for the subjects and for the simulation. The pattern of results is similar to the findings of Posner and Keele (1968), in the condition where subjects learned six exemplars which were rather close to the prototype. In this condition, their subjects' categorization performance was most accurate for the prototype, but more accurate for old than for new distortions, just as in this simulation experiment.

In Experiment 5, Whittlesea demonstrated that a slight advantage for stimuli closer to the prototype than the training stimuli would emerge, even with high-level distortions, when a large number of different distortions were used once each in training, instead of a smaller number of distortions presented three times each. The effect was rather small in Whittlesea's case (falling in the third decimal place in the per letter enhancement effect measure) but other experiments have produced similar results, and so does the simulation. In fact, because the prototype was tested in the simulation, we were able to demonstrate a monotonic drop in performance with distance from the prototype in this experiment.

Experiments 6 and 7 examine in different ways the relative influence of similarity to the prototype and similarity to the set of training exemplars, using small numbers of training exemplars rather far from the prototype. Both in the data and in the model, similarity to particular training stimuli is more important than similarity to the prototype, given the sets of training stimuli used in these experiments.

Taken together with other findings, Whittlesea's results show clearly that similarity of test items to particular stored exemplars is of paramount importance in predicting perceptual performance. Other experiments show the relevance of these same factors in other tasks, such as recognition memory, classification learning, and so forth. It is interesting to note that performance does not honor the specific exemplars so strongly when the training items are closer to the prototype. Under such conditions, performance is superior on the prototype or stimuli closer to the prototype than the training stimuli. Even when the training stimuli are rather distant from the prototype, they produce a benefit for stimuli closer to the prototype, if there are a large number of distinct training stimuli each shown only once. Thus, the dominance of specific training experiences is honored only when the training experiences are few and far between. Otherwise, an apparent advantage for the prototype, though with some residual benefit for particular training stimuli, is the result.

The congruity of the results of these simulations with experimental findings underscores the applicability of distributed models to the question of the nature of the representation of general and specific information. In fact, we were somewhat surprised by the ability of the model to account for Whittlesea's results, given the fact that we did not rely on context-sensitive encoding of the letter string stimuli. That is, the distributed representation we assigned to each letter was independent of the other letters in the string. However, a context sensitive encoding would prove necessary to capture a larger ensemble of stimuli.

Whether a context-sensitive encoding would produce the same or slightly different results depends on the exact encoding. The exact degree of overlap of the patterns of activation produced by different distortions of the same prototype determines the extent to which the model will tend to favor the prototype relative to particular old exemplars. The degree of overlap, in turn, depends on the specific assumptions made about the encoding of the stimuli. However, the general form of the results of the simulation would be unchanged: When all the distortions are close to the prototype, or when there is a very large number of different distortions, the central tendency will produce the strongest response; but when the distortions are fewer, and farther from the prototype, the training exemplars themselves will produce the strongest activations. What the encoding would effect is the similarity metric.

In this regard, it is worth mentioning another finding that appears to challenge our distributed account of what is learned through repeated experiences with exemplars. This is the finding of Medin and Schwanenflugel (1981). Their experiment compared ease of learning of two different sets of stimuli in a categorization task. One set of stimuli could be categorized by a linear combination of weights assigned to particular values on each of four dimensions considered independently. The other set of stimuli could not be categorized in this way; and yet, the experiment clearly demonstrated that linear separability was not necessary for categorization learning. In one experiment, linearly separable stimuli were less easily learned than a set of stimuli that were not linearly separable but had a higher degree of intraexemplar similiarity within categories.

At first glance, it may seem that Medin and Schwanenflugel's experiment is devastating to our distributed approach, because our distributed model can only learn linear combinations of weights. However, whether a linear combination of weights can suffice in the Medin and Schwanenflugel experiments depends on how patterns of activation are assigned to stimuli. If each stimulus dimension is encoded separately in the representation of the stimulus, then the Medin and Schwanenflugel stimuli cannot be learned by our model. But if each stimulus dimension is encoded in a context sensitive way, then the patterns of activation associated with the different stimuli become linearly separable again.

One way of achieving context sensitivity is via separate enumeration of traces. But it is well known that there are other ways as well. Several different kinds of context-sensitive encodings which do not require separate enumeration of traces, or the allocation of separate nodes to individual experiences are considered in Hinton (1981a), Hinton, McClelland, and Rumelhart (in press), and Rumelhart and McClelland (in press).

It should be noted that the motivation for context-sensitive encoding in the use of distributed representations is captured by but by no means limited to the kinds of observations reported in the experiment by Medin and Schwanenflugel. The trouble is that the assignment of particular context-sensitive encodings to stimuli is at present rather ad hoc: There are too many different possible ways it can be done to know which way is right. What is needed is a principled way of assigning distributed representations to patterns of activation. The problem is a severe one, but really it is no different from the problem that all models face, concerning the assignment of representations to stimuli. What we can say for sure at this point is that context-sensitive encoding is necessary, for distributed models or for any other kind.

Discussion

Until very recently, the exploration of distributed models was restricted to a few workers, mostly coming from fields other than cognitive psychology. Although in some cases, particularly in the work of Anderson (1977; Anderson et al., 1977; Knapp & Anderson, 1984), some implications of these models for our understanding of memory and learning have been pointed out, they have only begun to be applied by researchers primarily concerned with understanding cognitive processes per se. The present article, along with those of Murdock (1982) and Eich (1982), represents what we hope will be the beginning of a more serious examination of these kinds of models by cognitive psychologists. For they provide, we believe, important alternatives to traditional conceptions of representation and memory.

We have tried to illustrate this point here by showing how the distributed approach circumvents the dilemma of specific trace models. Distributed memories abstract even while they preserve the details of recent, or frequently repeated, experiences. Abstraction and preservation of information about specific stimuli are simply different reflections of the operation of the same basic learning mechanism.

The basic points we have been making can of course be generalized in several different directions. Here we will mention two: The relation between episodic and semantic memory (Tulving, 1972) and the representations underlying the use of language.

With regard to episodic and semantic memory, our distributed model leads naturally to the suggestion that semantic memory may be just the residue of the superposition of episodic traces. Consider, for example, representation of a proposition encountered in several different contexts, and assume for the moment that the context and content are represented in separate parts of the same module. Over repeated experience with the same proposition in different contexts, the proposition will remain in the interconnections of the units in the proposition submodule, but the particular associations to particular contexts will wash out. However, material that is only encountered in one particular context will tend to be somewhat contextually bound. So we may not be able to retrieve what we learn in one context when we need it in other situations. Other authors (e.g., Anderson & Ross, 1980) have recently argued against a distinction between episodic and semantic memory, pointing out interactions between traditionally episodic and semantic memory tasks. Such findings are generally consistent with the view we have taken here.

Distributed models also influence our thinking about how human behavior might come to exhibit the kind of regularity that often leads linguists to postulate systems of rules. We have recently developed a distributed model of a system that can learn the past tense system of English, given as inputs pairs of patterns, corresponding to the phonological structure of the present and past tense forms of actual English verbs (Rumelhart & McClelland, in press). Given plausible assumptions about the learning experiences to which a child is exposed, the model provides a fairly accurate account of the time course of acquisition of the past tense (Brown, 1973; Ervin, 1964; Kuczaj, 1977).

In general distributed models appear to provide alternatives to a variety of different kinds of models that postulate abstract, summary representations such as prototypes, logogens, semantic memory representations, or even linguistic rules.

Why Prefer a Distributed Model?

The fact that distributed models provide alternatives to other sorts of accounts is important, but the fact that they are sometimes linked rather closely to the physiology often makes them seem irrelevant to the basic enterprise of cognitive psychology. It may be conceded that distributed models describe the *physiological substrate* of memory better than other models, but why should we assume that they help us to characterize human information processing at a more abstract level of description? There are two parts to the answer to this question. First, though distributed models may be approximated by other models, on close inspection they differ from them in ways that should have testable consequences. If tests of these consequences turn out to favor distributed models—and there are indications that in certain cases they will—it would seem plausible to argue that distributed models provide an importantly different description of cognition, even if it does take the phenomena somewhat closer to the physiological level of analysis. Second, distributed models alter our thinking about a number of aspects of cognition at the same time. They give us a whole new constellation of assumptions about the structure of cognitive processes. They can change the way we think about the learning process, for example, and can even help shed some light on why and how human behavior comes to be as regular (as bound by rules and concepts) as it seems to be. In this section we consider these two points in turn.

A different level, or a different description? Are distributed models at a different level of analysis than cognitive models, or do they provide a different description of cognition? We think the answer is some of both. Here we focus primarily on underscoring the differences between distributed and other models.

Consider, first, the class of models which state that concepts are represented by prototypes. Distributed models approximate prototype models, and under some conditions their predictions converge, but under other conditions their predictions diverge. In particular, distributed models account both for conditions under which the prototype dominates and conditions under which particular exemplars dominate performance. Thus, they clearly have an advantage over such models, and should be preferred as accounts of empirical phenomena.

Perhaps distributed models are to be preferred over some cognitive level models, but one might argue that they are not to be preferred to the correct cognitive level model. For example, in most of the simulations discussed in this article, the predictions of enumeration models are not different from the predictions of our distributed model. Perhaps we should see our distributed model as representing a physiologically plausible implementation of enumeration models.

Even here, there are differences, however. Though both models superimpose traces of different experiences, distributed models do so at the time of storage, while enumeration models do so at the time of retrieval. But there is no evidence to support the separate storage assumption of enumeration models. Indeed, most such models assume that performance is always based on a superimposition of the specific experiences. Now, our distributed model could be rejected if convincing evidence of separate storage could be provided, for example, by some kind of experiment in which a way was found to separate the effects of different memory experiences. But the trend in a number of recent approaches to memory has been to emphasize the ubiquity of interactions between memory traces. Distributed models are essentially constructed around the assumption that memory traces interact by virtue of the nature of the manner in which they are stored, and they provide an explanation for these interactions. Enumeration models, on the other hand, simply assume interactions occur and postulate separate storage without providing any evidence that storage is in fact separate.

There is another difference between our distributed model and the enumeration models, at least existing ones. Our distributed model assumes that learning is an *error-correcting* process, whereas enumeration models do not. This difference leads to empirical consequences which put great strain on existing enumeration models. In existing enumeration models, what is stored in memory is simply a copy of features of the stimulus event, independent of the prior knowledge already stored in the memory system. But there are a number of indications that what is learned depends on the current state of knowledge. For example, the fact that learning is better after distributed practice appears to suggest that more learning occurs on later learning trials, if subjects have had a chance to forget what they learned on the first trial. We would expect such effects to occur in an error-correcting model such as ours.

The main point of the foregoing discussion has been to emphasize that our distributed model is not simply a plausible physiological implementation of existing models of cognitive processes. Rather, the model is an alternative to most, if not all, existing models, as we have tried to emphasize by pointing out differences between our distributed model and other models which have been proposed. Of course this does not mean that our distributed model will not turn out to be an exact notational variant of some particular other model. What it does mean is that our distributed model must be treated as an alternative to—rather than simply an implementation of—existing models of learning and memory.

Interdependence of theoretical assumptions. There is another reason for taking distributed models seriously as psychological models. Even in cases where our distributed model may not be testably distinct from existing models, it does provide an entire constellation of assumptions which go together as a package. In this regard, it is interesting to contrast a distributed model with a model such as John Anderson's ACT* model (J. R. Anderson, 1983). One difference between the models is that in ACT* it is productions rather than connection strengths that serve as the basis of learning and memory. This difference leads to other differences: in our model, learning occurs through connection strength modulation, whereas in ACT* learning occurs through the creation, differentiation, and generalization of productions. At a process level the models look very different, whether or not they make different empirical predictions. Learning in our distributed model is an automatic consequence of processing based on information locally available to each unit whose connections are changing; in ACT*, learning requires an overseer that detects cases in which a production has been misapplied, or in which two productions with similar conditions both fit the same input, to trigger the differentiation and generalization processes as appropriate.

Similar contrasts exist between our distributed model and other models; in general, our model differs from most abstractive models (that is, those that postulate the formation of abstract rules or other abstract representations) in doing away with complex acquisition mechanisms in favor of a very simple connection strength modulation scheme. Indeed, to us, much of the appeal of distributed models is that they do not already have to be intelligent in order to learn, like some models do. Doubtless, sophisticated hypothesis testing models of learning such as those which have grown out of the early concept identification work of Bruner, Goodnow, and Austin (1956) or out of the artificial intelligence learning tradition established by Winston (1975) have their place, but for many phenomena, particularly those that do not seem to require explicit hypothesis formation and testing, the kind of learning mechanism incorporated in our distributed model may be more appropriate.

Two final reasons for preferring a distributed representation are that it leads us to understand some of the reasons why human behavior tends to exhibit such strong regularities. Some of the regularity is due to the structure of the world, of course, but much of it is a result of the way in which our cultures structure it; certainly the regularity of languages is a fact about the way humans communicate that psychological theory can be asked to explain. Distributed models provide some insight both into why it is beneficial for behavior to be regular, and how it comes to be that way.

It is beneficial for behavior to be regular, because regularity allows us to economize on the size of the networks that must be devoted to processing in a particular environment. If all experiences were completely random and unrelated to each other, a distributed model would buy us very little—in fact it would cost us a bit—relative to separate enumeration of experiences. An illuminating analysis of this situation is given by Willshaw (1981). Where a distributed model pays off, though, is in the fact that it can capture generalizations economically, given that there are generalizations. Enumeration models lack this feature. There are of course limits on how much can be stored in a distributed memory system, but the fact that it can abstract extends those limits far beyond the capacity of any system relying on the separate enumeration of experiences, whenever abstraction is warranted by the ensemble of inputs.

We have just explained how distributed models can help us understand why it is a good thing for behavior to exhibit regularity, but we have not yet indicated how they help us understand how it comes to be regular. But it is easy to see how distributed models tend to impose regularity. When a new pattern is presented, the model will impose regularity by dealing with it as it has learned to deal with similar patterns in the past; the model automatically generalizes. In our analysis of past tense learning (Rumelhart & McClelland, in press), it is just this property of distributed models which leads them to produce the kinds of over-regularizations we see in language development; the same property, operating in all of the members of a culture at the same time, will tend to produce regularizations in the entire language.

Conclusion

The distributed approach is in its infancy, and we do not wish to convey the impression that we have solved all the problems of learning and memory simply by invoking it. Considerable effort is needed on several fronts. We will mention four that seem of paramount importance: (a) Distributed models must be integrated with models of the overall organization of information processing, and their relation to models of extended retrieval processes and other temporally extended mental activities must be made clear. (b) Models must be formulated which adequately capture the structural relations of the components of complex stimuli. Existing models do not do this in a sufficiently flexible and open-ended way to capture arbitrarily complex propositional structures. (c) Ways must be found to take the assignment of patterns of activation to stimuli out of the hands of the modeler, and place them in the structure of the model itself. (d) Further analysis is required to determine which of the assumptions of our particular distributed model are essential and which are unimportant details. The second and third of these problems are under intensive study. Some developments along these lines are reported in a number of recent papers (Ackley, Hinton, & Sejnowski, 1985; McClelland, 1985; Rumelhart & Zipser, 1985).

Although much remains to be done, we hope we have demonstrated that distributed models provide distinct, conceptually attractive alternatives to models involving the explicit formation of abstractions or the enumeration of specific experiences. Just how far distributed models can take us toward an understanding of learning and memory remains to be seen.

References

Ackley, D., Hinton, G. E., & Sejnowski, T. J. (1985). Boltzmann machines: Constraint satisfaction networks that learn. *Cognitive Science, 9,* 147–169.

Anderson, J. A. (1977). Neural models with cognitive implications. In D. LaBerge & S. J. Samuels (Eds.), *Basic processes in reading: Perception and comprehension.* Hillsdale, NJ: Erlbaum.

Anderson, J. A. (1983). Cognitive and psychological computation with neural models. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13,* 799–815.

Anderson, J. A., & Hinton, G. E. (1981). Models of information processing in the brain. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum.

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

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard.

Anderson, J. R., & Ross, B. H. (1980). Evidence against a semantic-episodic distinction. *Journal of Experimental Psychology: Human Learning and Memory, 6,* 441–465.

Brooks, L. R. (1978). Nonanalytic concept formation and memory for instances. In E. Rosch & B. B. Lloyd (Eds.), *Cognition and categorization.* Hillsdale, NJ: Erlbaum.

Brown, R. (1973). *A first language.* Cambridge, MA: Harvard University Press.

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A study of thinking.* New York: Wiley.

Eich, J. M. (1982). A composite holgraphic associative retrieval model. *Psychological Review, 89,* 627–661.

Ervin, S. (1964). Imitation and structural change in children's language. In E. Lenneberg (Ed.), *New directions in the study of language.* Cambridge, MA: MIT Press.

Feustel, T. C., Shiffrin, R. M., & Salasoo, A. (1983). Episodic and lexical contributions to the repetition effect in word identification. *Journal of Experimental Psychology: General, 112,* 309–346.

Glushko, R. J. (1979). The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance, 5,* 674–691.

Hinton, G. E. (1981a). Implementing semantic networks in parallel hardware. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum.

Hinton, G. E. (1981b). A parallel computation that assigns canonical object-based frames of reference. *Proceedings of the Seventh International Joint Conference in Artificial Intelligence* (pp. 683-685). Vancouver, British Columbia, Canada.

Hinton, G. E., & Anderson, J. A. (Eds.). (1981). *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (in press). Distributed representations. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume I: Foundations.* Cambridge, MA; Bradford Books.

Hintzman, D. (1983). *Schema abstraction in a multiple trace memory model.* Paper presented at conference on "The priority of the specific." Elora, Ontario, Canada.

Jacoby, L. L. (1983a). Perceptual enhancement: Persistent effects of an experience. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9,* 21–38.

Jacoby, L. L. (1983b). Remembering the data: Analyzing interaction processes in reading. *Journal of Verbal Learning and Verbal Behavior, 22,* 485–508.

Knapp, A., & Anderson, J. A. (1984). A signal averaging model for concept formation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 10,* 616–637.

Kohonen, T. (1977). *Associative memory: A system-theoretical approach.* Berlin: Springer-Verlag.

Kohonen, T., Oja, E., & Lehtio, P. (1981). Storage and processing of information in distributed associative memory systems. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum.

Kuczaj, S. A., II. (1977). The acquisition of regular and irregular past tense forms. *Journal of Verbal Learning and Verbal Behavior, 16,* 589–600.

Lashley, K. S. (1950). In search of the engram. *Society for Experimental Biology Symposium No. 4: Physiological Mechanisms in Animal Behavior* (pp. 478–505). London: Cambridge University Press.

Luce, R. D. (1963). Detection and recognition. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of Mathematical Psychology: Vol. I.* New York: Wiley.

McClelland, J. L. (1979). On the time-relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86,* 287–330.

McClelland, J. L. (1981). Retrieving general and specific information from stored knowledge of specifics. *Proceedings of the Third Annual Meeting of the Cognitive Science Society* (pp. 170–172). Berkeley, CA.

McClelland, J. L. (1985). Putting knowledge in its place: A framework for programming parallel processing structures on the fly. *Cognitive Science, 9,* 113–146.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of the effect of context in perception, Part I. An account of basic findings. *Psychological Review, 88,* 375–407.

Medin, D., & Schwanenflugel, P. J. (1981). Linear separability in classification learning. *Journal of Experimental Psychology: Human Learning and Memory, 7,* 355–368.

Medin, D. L., & Shaffer, M. M. (1978). Context theory of classification learning. *Psychological Review, 85,* 207–238.

Morton, J. (1979). Facilitation in word recognition: Experiments causing change in the logogen model. In

P. A. Kohlers, M. E. Wrolstal, & H. Bouma (Eds.), *Processing visible language I.* New York: Plenum.

Murdock, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review, 89,* 609-626.

Posner, M. I., & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology, 77,* 353-363.

Posner, M. I., & Keele, S. W. (1970). Retention of abstract ideas. *Journal of Experimental Psychology, 83,* 304-308.

Rosenblatt, F. (1962). *Principles of neurodynamics.* Washington, DC: Spartan.

Rumelhart, D. E., & McClelland, J. L. (1981). Interactive processing through spreading activation. In A. M. Lesgold & C. A. Perfetti (Eds.), *Interactive Processes in Reading.* Hillsdale, NJ: Erlbaum.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of the effect of context in perception Part II. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review, 89,* 60-94.

Rumelhart, D. E., & McClelland, J. L. (in press). On learning the past tenses of English verbs. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume II: Applications.* Cambridge, MA: Bradford Books.

Rumelhart, D. E., & Zipser, D. (1985). Competitive learning. *Cognitive Science, 9,* 75-112.

Salasoo, A., Shiffrin, R. M., & Feustel, T. C. (1985). Building permanent memory codes: Codification and repetition effects in word identification. *Journal of Experimental Psychology: General, 114,* 50-77.

Stone, G. (1985). *An analysis of the delta rule.* Manuscript in preparation.

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review, 88,* 135-170.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), *Organization of Memory.* New York: Academic Press.

Whittlesea, B. W. A. (1983). *Representation and generalization of concepts: The abstractive and episodic perspectives evaluated.* Unpublished doctoral dissertation, MacMaster University.

Wickelgren, W. A. (1979). 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, 86,* 44-60.

Willshaw, D. (1981). Holography, associative memory, and inductive generalization. In G. E. Hinton & J. A. Anderson (Eds.), *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum.

Winston, P. H. (1975). Learning structural descriptions from examples. In P. H. Winston (Ed.), *The psychology of computer vision.* Cambridge, MA: Harvard.

Received May 22, 1984
Revision received October 2, 1984 ■

Improved Reproduction of Photomicrographs in *Behavioral Neuroscience*

Behavioral Neuroscience is pleased to announce new and improved photomicrograph reproduction. Previously, photomicrographs lacked the high resolution needed for detailed study. Beginning in 1985, the photomicrograph will appear twice: once in the text, as usual, and again in a special added signature of better quality, coated paper stock that will yield substantially more detail.
