PARALLEL DISTRIBUTED PROCESSING

Explorations in the Microstructure of Cognition

Volume 1: Foundations

David E. Rumelhart James L. McClelland
and the PDP Research Group

Chisato Asanuma
Francis H. C. Crick
Jeffrey L. Elman
Geoffrey E. Hinton
Michael I. Jordan
Alan H. Kawamoto
Paul W. Munro
Donald A. Norman
Daniel E. Rabin
Terrence J. Sejnowski
Paul Smolensky
Gregory O. Stone
Ronald J. Williams
David Zipser

Institute for Cognitive Science
University of California, San Diego

A Bradford Book

The MIT Press
Cambridge, Massachusetts
London, England

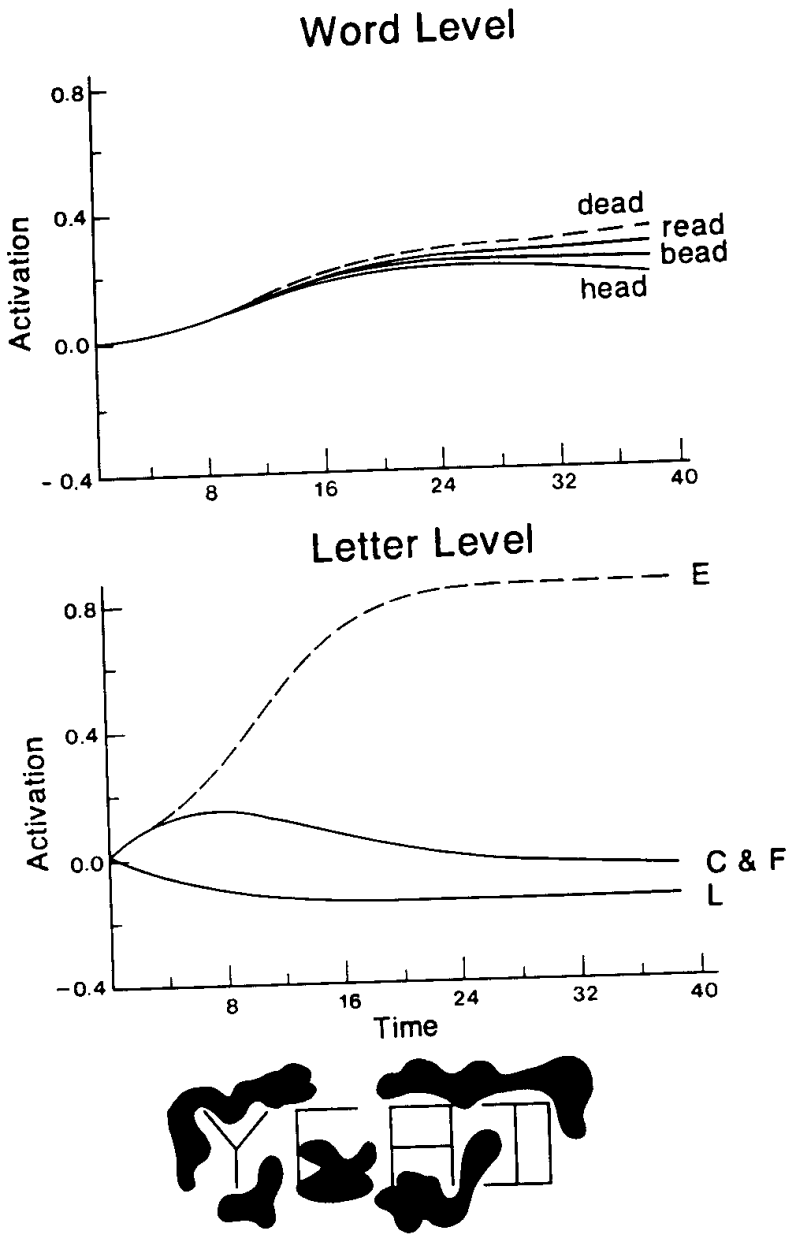

FIGURE 9. An example of a nonword display that might be presented to the interactive activation model of word recognition and the response of selected units at the letter and word levels. The letter units illustrated are detectors for letters in the second input position.

based on a set of rules representing human competence—at least in some domains.

Retrieving Information From Memory

Content addressability. One very prominent feature of human memory is that it is content addressable. It seems fairly clear that we

can access information in memory based on nearly any attribute of the representation we are trying to retrieve.

Of course, some cues are much better than others. An attribute which is shared by a very large number of things we know about is not a very effective retrieval cue, since it does not accurately pick out a particular memory representation. But, several such cues, in conjunction, can do the job. Thus, if we ask a friend who goes out with several women, "Who was that woman I saw you with?", he may not know which one we mean—but if we specify something else about her—say the color of her hair, what she was wearing (in so far as he remembers this at all), where we saw him with her—he will likely be able to hit upon the right one.

It is, of course, possible to implement some kind of content addressability of memory on a standard computer in a variety of different ways. One way is to search sequentially, examining each memory in the system to find the memory or the set of memories which has the particular content specified in the cue. An alternative, somewhat more efficient, scheme involves some form of indexing—keeping a list, for every content a memory might have, of which memories have that content.

Such an indexing scheme can be made to work with error-free probes, but it will break down if there is an error in the specification of the retrieval cue. There are possible ways of recovering from such errors, but they lead to the kind of combinatorial explosions which plague this kind of computer implementation.

But suppose that we imagine that each memory is represented by a unit which has mutually excitatory interactions with units standing for each of its properties. Then, whenever any property of the memory became active, the memory would tend to be activated, and whenever the memory was activated, all of its contents would tend to become activated. Such a scheme would automatically produce content addressability for us. Though it would not be immune to errors, it would not be devastated by an error in the probe if the remaining properties specified the correct memory.

As described thus far, whenever a property that is a part of a number of different memories is activated, it will tend to activate all of the memories it is in. To keep these other activities from swamping the "correct" memory unit, we simply need to add initial inhibitory connections among the memory units. An additional desirable feature would be mutually inhibitory interactions among mutually incompatible property units. For example, a person cannot both be single and married at the same time, so the units for different marital states would be mutually inhibitory.

McClelland (1981) developed a simulation model that illustrates how a system with these properties would act as a content addressable memory. The model is obviously oversimplified, but it illustrates many of the characteristics of the more complex models that will be considered in later chapters.

Consider the information represented in Figure 10, which lists a number of people we might meet if we went to live in an unsavory neighborhood, and some of their hypothetical characteristics. A subset

The Jets and The Sharks

Name	Gang	Age	Edu	Mar	Occupation
Art	Jets	40's	J.H.	Sing.	Pusher
Al	Jets	30's	J.H.	Mar.	Burglar
Sam	Jets	20's	COL.	Sing.	Bookie
Clyde	Jets	40's	J.H.	Sing.	Bookie
Mike	Jets	30's	J.H.	Sing.	Bookie
Jim	Jets	20's	J.H.	Div.	Burglar
Greg	Jets	20's	H.S.	Mar.	Pusher
John	Jets	20's	J.H.	Mar.	Burglar
Doug	Jets	30's	H.S.	Sing.	Bookie
Lance	Jets	20's	J.H.	Mar.	Burglar
George	Jets	20's	J.H.	Div.	Burglar
Pete	Jets	20's	H.S.	Sing.	Bookie
Fred	Jets	20's	H.S.	Sing.	Pusher
Gene	Jets	20's	COL.	Sing.	Pusher
Ralph	Jets	30's	J.H.	Sing.	Pusher
Phil	Sharks	30's	COL.	Mar.	Pusher
Ike	Sharks	30's	J.H.	Sing.	Bookie
Nick	Sharks	30's	H.S.	Sing.	Pusher
Don	Sharks	30's	COL.	Mar.	Burglar
Ned	Sharks	30's	COL.	Mar.	Bookie
Karl	Sharks	40's	H.S.	Mar.	Bookie
Ken	Sharks	20's	H.S.	Sing.	Burglar
Earl	Sharks	40's	H.S.	Mar.	Burglar
Rick	Sharks	30's	H.S.	Div.	Burglar
Ol	Sharks	30's	COL.	Mar.	Pusher
Neal	Sharks	30's	H.S.	Sing.	Bookie
Dave	Sharks	30's	H.S.	Div.	Pusher

FIGURE 10. Characteristics of a number of individuals belonging to two gangs, the Jets and the Sharks. (From "Retrieving General and Specific Knowledge From Stored Knowledge of Specifics" by J. L. McClelland, 1981, *Proceedings of the Third Annual Conference of the Cognitive Science Society*, Berkeley, CA. Copyright 1981 by J. L. McClelland Reprinted by permission.)

of the units needed to represent this information is shown in Figure 11. In this network, there is an "instance unit" for each of the characters described in Figure 10, and that unit is linked by mutually excitatory connections to all of the units for the fellow's properties. Note that we have included property units for the names of the characters, as well as units for their other properties.

Now, suppose we wish to retrieve the properties of a particular individual, say Lance. And suppose that we know Lance's name. Then we can probe the network by activating Lance's name unit, and we can see what pattern of activation arises as a result. Assuming that we know of no one else named Lance, we can expect the Lance name unit to be hooked up only to the instance unit for Lance. This will in turn activate the property units for Lance, thereby creating the pattern of

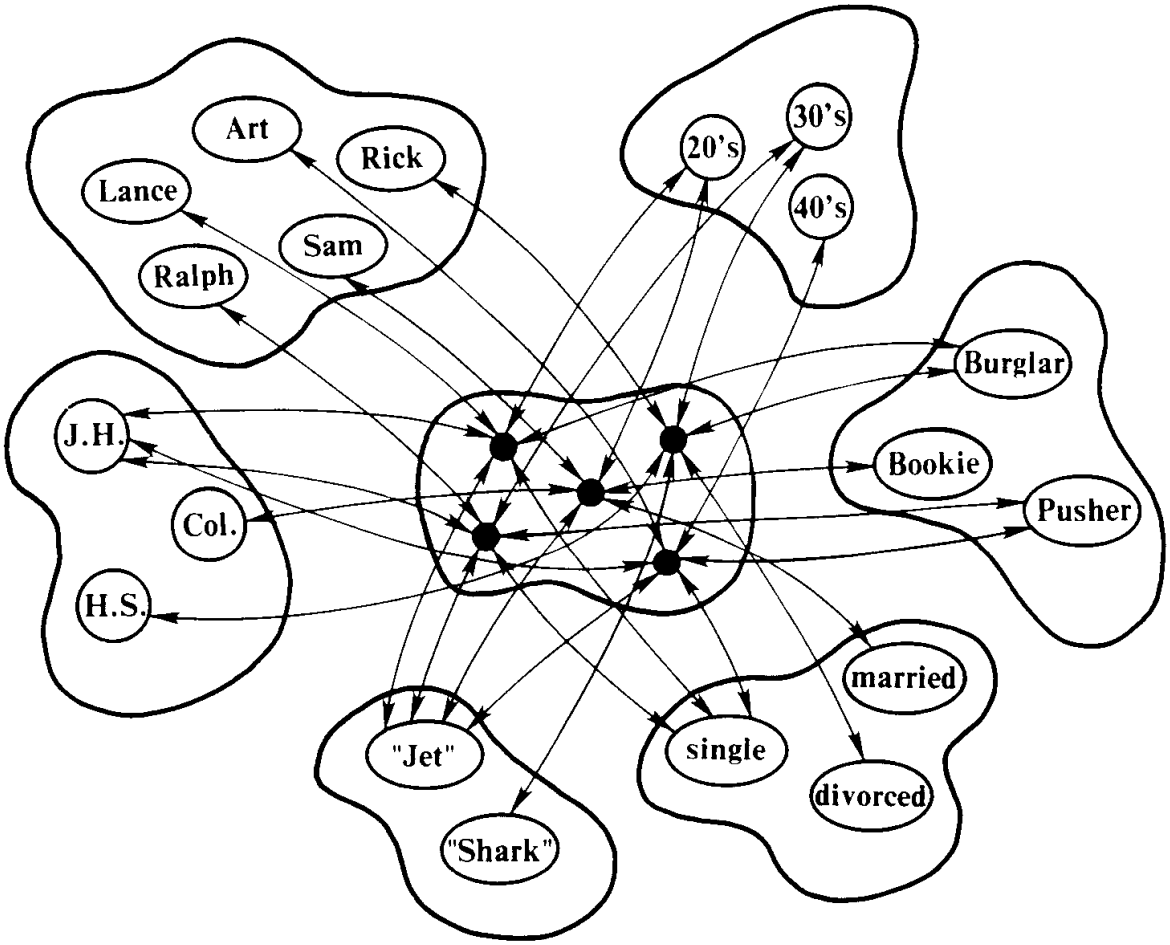

FIGURE 11. Some of the units and interconnections needed to represent the individuals shown in Figure 10. The units connected with double-headed arrows are mutually excitatory. All the units within the same cloud are mutually inhibitory. (From "Retrieving General and Specific Knowledge From Stored Knowledge of Specifics" by J. L. McClelland, 1981, *Proceedings of the Third Annual Conference of the Cognitive Science Society*, Berkeley, CA. Copyright 1981 by J. L. McClelland. Reprinted by permission.)

activation corresponding to Lance. In effect, we have retrieved a representation of Lance. More will happen than just what we have described so far, but for the moment let us stop here.

Of course, sometimes we may wish to retrieve a name, given other information. In this case, we might start with some of Lance's properties, effectively asking the system, say "Who do you know who is a Shark and in his 20s?" by activating the Shark and 20s units. In this case it turns out that there is a single individual, Ken, who fits the description. So, when we activate these two properties, we will activate the instance unit for Ken, and this in turn will activate his name unit, and fill in his other properties as well.

Graceful degradation. A few of the desirable properties of this kind of model are visible from considering what happens as we vary the set of features we use to probe the memory in an attempt to retrieve a particular individual's name. Any set of features which is sufficient to uniquely characterize a particular item will activate the instance node for that item more strongly than any other instance node. A probe which contains misleading features will most strongly activate the node that it matches best. This will clearly be a poorer cue than one which contains no misleading information—but it will still be sufficient to activate the "right answer" more strongly than any other, as long as the introduction of misleading information does not make the probe closer to some other item. In general, though the degree of activation of a particular instance node and of the corresponding name nodes varies in this model as a function of the exact content of the probe, errors in the probe will not be fatal unless they make the probe point to the wrong memory. This kind of model's handling of incomplete or partial probes also requires no special error-recovery scheme to work—it is a natural by-product of the nature of the retrieval mechanism that it is capable of graceful degradation.

These aspects of the behavior of the Jets and Sharks model deserve more detailed consideration than the present space allows. One reason we do not go into them is that we view this model as a stepping stone in the development of other models, such as the models using more distributed representations, that occur in other parts of this book. We do, however, have more to say about this simple model, for like some of the other models we have already examined, this model exhibits some useful properties which emerge from the interactions of the processing units.

Default assignment. It probably will have occurred to the reader that in many of the situations we have been examining, there will be other

activations occurring which may influence the pattern of activation which is retrieved. So, in the case where we retrieved the properties of Lance, those properties, once they become active, can begin to activate the units for other individuals with those same properties. The memory unit for Lance will be in competition with these units and will tend to keep their activation down, but to the extent that they do become active, they will tend to activate their own properties and therefore fill them in. In this way, the model can fill in properties of individuals based on what it knows about other, similar instances.

To illustrate how this might work we have simulated the case in which we do not know that Lance is a Burglar as opposed to a Bookie or a Pusher. It turns out that there are a group of individuals in the set who are very similar to Lance in many respects. When Lance's properties become activated, these other units become partially activated, and they start activating their properties. Since they all share the same "occupation," they work together to fill in that property for Lance. Of course, there is no reason why this should necessarily be the right answer, but generally speaking, the more similar two things are in respects that we know about, the more likely they are to be similar in respects that we do not, and the model implements this heuristic.

Spontaneous generalization. The model we have been describing has another valuable property as well—it tends to retrieve what is common to those memories which match a retrieval cue which is too general to capture any one memory. Thus, for example, we could probe the system by activating the unit corresponding to membership in the Jets. This unit will partially activate all the instances of the Jets, thereby causing each to send activations to its properties. In this way the model can retrieve the typical values that the members of the Jets have on each dimension—even though there is no one Jet that has these typical values. In the example, 9 of 15 Jets are single, 9 of 15 are in their 20s, and 9 of 15 have only a Junior High School education; when we probe by activating the Jet unit, all three of these properties dominate. The Jets are evenly divided between the three occupations, so each of these units becomes partially activated. Each has a different name, so that each name unit is very weakly activated, nearly cancelling each other out.

In the example just given of spontaneous generalization, it would not be unreasonable to suppose that someone might have explicitly stored a generalization about the members of a gang. The account just given would be an alternative to "explicit storage" of the generalization. It has two advantages, though, over such an account. First, it does not require any special generalization formation mechanism. Second, it can provide us with generalizations on unanticipated lines, on demand.

Thus, if we want to know, for example, what people in their 20s with a junior high school education are like, we can probe the model by activating these two units. Since all such people are Jets and Burglars, these two units are strongly activated by the model in this case; two of them are divorced and two are married, so both of these units are partially activated.[1]

The sort of model we are considering, then, is considerably more than a content addressable memory. In addition, it performs default assignment, and it can spontaneously retrieve a general concept of the individuals that match any specifiable probe. These properties must be explicitly implemented as complicated computational extensions of other models of knowledge retrieval, but in PDP models they are natural by-products of the retrieval process itself.

REPRESENTATION AND LEARNING IN PDP MODELS

In the Jets and Sharks model, we can speak of the model's *active representation* at a particular time, and associate this with the pattern of activation over the units in the system. We can also ask: What is the stored knowledge that gives rise to that pattern of activation? In considering this question, we see immediately an important difference between PDP models and other models of cognitive processes. In most models, knowledge is stored as a static copy of a pattern. Retrieval amounts to finding the pattern in long-term memory and copying it into a buffer or working memory. There is no real difference between the stored representation in long-term memory and the active representation in working memory. In PDP models, though, this is not the case. In these models, the patterns themselves are not stored. Rather, what is stored is the *connection strengths* between units that allow these patterns to be re-created. In the Jets and Sharks model, there is an instance unit assigned to each individual, but that unit does not contain a copy of the representation of that individual. Instead, it is simply the case that the connections between it and the other units in the system are such that activation of the unit will cause the pattern for the individual to be reinstated on the property units.

[1] In this and all other cases, there is a tendency for the pattern of activation to be influenced by partially activated, near neighbors, which do not quite match the probe. Thus, in this case, there is a Jet Al, who is a Married Burglar. The unit for Al gets slightly activated, giving Married a slight edge over Divorced in the simulation.