

# Mind Design II

Philosophy  
Psychology  
Artificial Intelligence

Revised and enlarged edition

edited by  
John Haugeland

A Bradford Book  
The MIT Press  
Cambridge, Massachusetts  
London, England

# Mind Design II

Philosophy  
Psychology  
Artificial Intelligence

Revised and enlarged edition

edited by  
John Haugeland

A Bradford Book  
The MIT Press  
Cambridge, Massachusetts  
London, England

*for Barbara and John III*

Second printing, 1997

© 1997 Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

Book design and typesetting by John Haugeland. Body text set in Adobe Garamond 11.5 on 13; titles set in Zapf Humanist 601 BT. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Mind design II / edited by John Haugeland. — 2nd ed., rev. and enlarged.

p. cm.

"A Bradford book."

Includes bibliographical references.

ISBN 0-262-08259-4 (hc : alk. paper). — ISBN 0-262-58153-1 (pb : alk. paper)

1. Artificial intelligence. 2. Cognitive psychology.

I. Haugeland, John, 1945–

Q335.5.M492 1997

006.3—dc21

96-45188

CIP

that the preliminary training need only be approximately right. A final round of training can be used after assembly to learn the interfaces among the modules.

### 2.3 The generalization problem

One final aspect of learning that has been looked at is the nature of generalization. It is clear that the most important aspect of networks is not that they learn a set of mappings but that they learn the function implicit in the exemplars under study in such a way that they respond properly to cases not yet observed. Although there are many examples of successful generalization (e.g., the learning of spelling-to-phoneme mappings in Sejnowski and Rosenberg's NETtalk, 1987), there are a number of cases in which the networks do not generalize correctly (see Denker et al. 1987). One simple way to understand this is to note that for most problems there are enough degrees of freedom in the network that there are a large number of genuinely different solutions to the problems—each of which constitutes a different way of generalizing to unseen patterns. Clearly not all of these can be correct.

Weigend and I have proposed an hypothesis that shows some promise in promoting better generalization (Weigend and Rumelhart 1991). The basic idea is this: the problem of generalization is essentially the induction problem. Given a set of observations, what is the appropriate principle that applies to all cases? Note that the network at any point in time can be viewed as a specification of an inductive hypothesis. Our proposal is that we follow a version of Occam's razor and select the *simplest, most robust* network that is consistent with the observations made. The assumption of robustness is simply an embodiment of a kind of continuity assumption that small variations in the input pattern should have little effect on the output or on the performance of the system. The simplicity assumption is simply to choose—of all networks that correctly account for the input data—the net with the fewest hidden units, the fewest connections, the most symmetries among the weights, and so on. We have formalized this procedure and modified the backpropagation learning procedure so that it prefers simple, robust networks, and, all things being equal, will select those networks. In many cases it turns out that these are just the networks that do the best job generalizing.

## Connectionist Modeling: Neural Computation / Mental Connections

9

Paul Smolensky  
1989

In the past few years the approach to cognitive science and artificial intelligence known as *connectionist modeling* has dramatically increased its influence. Connectionist systems are large networks of extremely simple computational units, massively interconnected and running in parallel. Each unit or processor has a numerical *activation value* which it communicates to other processors along connections of varying strength; the activation value of each processor constantly changes in response to the activity of the processors to which it is connected. The values of some of the units form the input to the system, and the values of other units form the output; the connections between the units determine how input is transformed to output. In connectionist systems, knowledge is encoded not in symbolic structures but rather in the pattern of numerical strengths of the connections between units.

The goal of connectionist research is to model both lower-level perceptual processes and such higher-level processes as object recognition, problem solving, planning, and language understanding. The rapidly growing collection of connectionist systems includes models of the following cognitive phenomena:

- speech perception,
- visual recognition of figures in the "origami world",
- development of specialized feature detectors,
- amnesia,
- language parsing and generation,
- aphasia,
- discovering binary encodings,
- dynamic programming of massively parallel networks,
- acquisition of English past tense morphophonology from examples,

- tic-tac-toe,
- inference about rooms, and
- qualitative problem solving in simple electric circuits.

One crucial question is whether the computational power of connectionist systems is sufficient for the construction of truly intelligent systems. Explorations addressing this question form the bulk of the contributions to the connectionist literature; many can be found in the proceedings of the International Joint Conference on AI, the annual meetings of the American Association for AI, and the Cognitive Science Society over the past several years. The connectionist systems referred to in the previous paragraph can be found in the collections in Hinton and Anderson (1981); Feldman (1985); Rumelhart, McClelland, and the PDP Research Group (1986); McClelland, Rumelhart, and the PDP Research Group (1986); and the bibliography by Feldman, Ballard, Brown, and Dell (1985). In the present paper I will not address the issue of computational power, except to point out that connectionist research has been strongly encouraged by successful formal models of the details of human cognitive performance, and strongly motivated by the conviction that the pursuit of the principles of neural computation will eventually lead to architectures of great computational power.

In addition to the question of whether the connectionist approach to AI *can* work, there is the question: What exactly would it mean if the approach *did* work? There are fundamental questions about the connectionist approach that are not yet clearly understood despite their importance. What is the relation between connectionist systems and the brain? How does the connectionist approach to modeling higher-level cognitive processes relate to the symbolic approach that has traditionally *defined* AI and cognitive science? Can connectionist models contribute to our understanding of the nature of the symbol processing characterizing the mind and its relation to the neural processing characterizing the brain? These are the questions I address in this paper. In the process of addressing these questions it will become clear that the answers are important not only in their own right, but also as contributions to the determination of whether the connectionist approach has sufficient power.

## 1 Levels of analysis: neural and mental structures

We begin with the questions: How do accounts of intelligence relate to neural and mental structures? What are the roles of the neural and the symbolic levels of analysis? We first consider the answers from the traditional symbolic approach to AI, and then from a connectionist alternative.

### 1.1 The symbolic paradigm

We start with the mental structures of "folk psychology": goals, beliefs, concepts, and so forth (figure 9.1). In the symbolic approach, these mentalist concepts are formalized in terms of a "language of thought", as Fodor (1975) calls it; this language is supposed to provide a literal formalization of folk psychology. The rules for operating on this language are essentially Boole's (1854/1961) "laws of thought". These symbolic structures are supported by a *physical symbol system* (a physical computing device for manipulating symbols), which in turn is supported by lower implementation levels in a computing device. The idea is that, eventually, if we were to get low enough down in the human physical symbol system, we would see something like neurons. In other words, on this account we just have to figure out how to relate neural structures to the low implementation levels of a physical symbol system, and then we understand the relation between neural structures and mental structures. If it were the case that increasingly lower levels

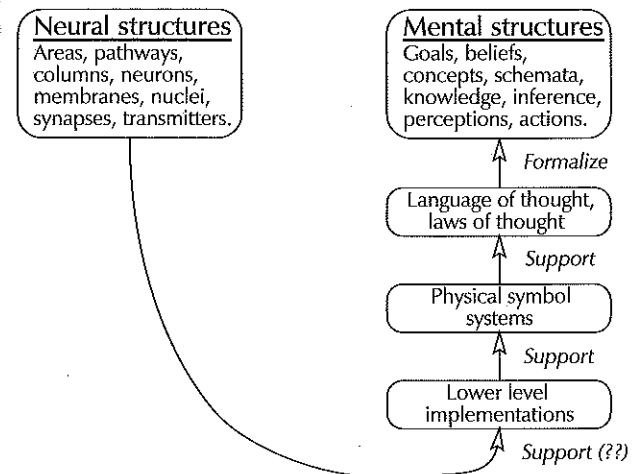


Figure 9.1: Neural and mental structures in the symbolic paradigm

of computers looked more and more like neural systems this would be a promising approach; unfortunately, insights into the design and implementation of physical symbol systems have so far shed virtually no light on how the brain works.

To understand the connectionist alternative more clearly, it is helpful to articulate a number of the properties of the symbolic approach. Allen Newell formulated this paradigm best in his *physical symbol system hypothesis*:

The necessary and sufficient condition for a physical system to exhibit general intelligent action is that it be a physical symbol system. (1980, p. 170).

"General intelligent action" means rational behavior (p. 171); "rationality" means that, when an agent has a certain goal and the knowledge that a certain action will lead to that goal, then the agent selects that action (Newell 1982); and physical symbol systems are physically realized universal computers.

What all this means in the practice of symbolic AI is that goals, beliefs, knowledge, and so on are all formalized as symbolic structures (for instance, Lisp lists) that are built of symbols (such as Lisp atoms) that are each semantically interpretable in terms of the ordinary concepts we use to conceptualize the domain. Thus in a medical expert system, we expect to find structures like (IF FEVER THEN (HYPOTHEZIZE INFECTION)). These symbolic structures are operated on by symbol-manipulation procedures composed of primitive operations like concatenating lists and extracting elements from lists. According to the symbolic paradigm, it is in terms of such operations that we are to understand cognitive processes.

It is important to note that in the symbolic paradigm, levels of cognition are analogized to levels of computer systems. The symbolic level that implements knowledge structures is alleged to be exact and complete. That means that lower levels are *unnecessary* for the accurate description of cognition in terms of the semantically interpretable elements. This relegates the neural question to simply: How does the nervous system happen to physically implement a physical symbol system? The answer to this question does not matter as far as symbol-level AI systems are concerned.

In this paradigm—which Hofstadter (1985) has called "the Boolean dream"—there are a number of inadequacies. These can be seen from a number of perspectives, which can only be caricatured here:

- From the perspective of neuroscience, the problem with the symbolic paradigm is quite simply, as I have already indicated, that it has provided precious little insight into the computational organization of the brain.
- From the perspective of modeling human performance, symbolic models, such as Newell and Simon's General Problem Solver (1972), do a good job on a coarse level; but the fine structure of cognition seems to be more naturally described by nonsymbolic models. In word recognition, for example, it is natural to think about activation levels of perceptual units.
- In AI, the trouble with the Boolean dream is that symbolic rules and the logic used to manipulate them tend to produce rigid and brittle systems.

## 1.2 The subsymbolic paradigm

The alternative to the symbolic paradigm that I want to present is what I call the *subs symbolic paradigm* (figure 9.2). In this paradigm, there is an intermediate level of structure between the neural and symbolic levels. This new *subs symbolic level* is supposed to be closer to each of the neural and symbolic levels than they are to each other. When cognition is described at the subsymbolic level, the description is that of a connectionist system.

The subsymbolic level is an attempt to formalize, *at some level of abstraction*, the kind of processing occurring in the nervous system. Many of the details of neural structure and function are absent from the subsymbolic level, and the level of description is higher than the

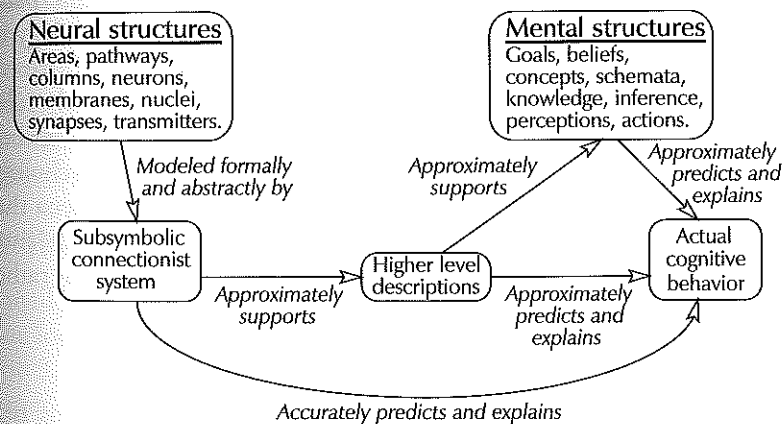


Figure 9.2: Neural and mental structures in the subsymbolic paradigm.

neural level. The precise relationship between the neural and subsymbolic levels is still a fairly wide-open research question; but it seems quite clear that connectionist systems are much closer to neural systems than are symbolic systems.

The relation between the subsymbolic and symbolic descriptions of cognition is illustrated in figure 9.2. If we adopt a higher level of description of what's going on in these subsymbolic systems (and that involves, to a significant degree, approximation) then we get descriptions that are approximately like symbolic accounts—the sort that traditional AI constructs. While the subsymbolic paradigm is content to give approximate accounts of things like goals and beliefs, it is not prepared to compromise on actual performance. Behind the accounts of folk psychology and symbolic AI, there lie real data on human intelligent performance; and the claim is that subsymbolic systems can provide accurate accounts of that data.

Note that the subsymbolic paradigm gives an essentially different role to the neural part of the story: Neural structures provide the basis (in some suitably abstract sense) of the formalism that gives the precise description of intelligence, whereas mental structures enter only into approximate descriptions.

In the remainder of the essay, I will elaborate on the nature of the subsymbolic level, and on the higher-level descriptions of subsymbolic systems that approximate symbolic accounts. I want to indicate how formalizing cognition by abstracting from neural structures—rather than with symbolic formalizations of mental structures—provides new and exciting views of knowledge, memory, concepts, and learning.

Figure 9.2 illustrates an important part of the subsymbolic paradigm: that levels of cognition should *not* be thought of by analogy to levels of computer systems, all stacked underneath the “mental” part of the diagram. Just as Newtonian concepts provide approximately valid descriptions of physical phenomena that are more accurately described with quantum concepts, so the symbolic concepts of folk psychology provide approximately valid descriptions of cognitive phenomena that are more accurately described with subsymbolic concepts. Mental structures are like higher-level descriptions of a *physical* system, rather than higher-level descriptions of a *computer* system.

### 1.3 Semantic interpretation

Perhaps the most fundamental contrast between the paradigms pertains to semantic interpretation of the formal models. In the symbolic

approach, symbols (atoms) are used to denote the semantically interpretable entities (concepts); these same symbols are the objects governed by symbol manipulations in the rules that define the system. The entities which are semantically interpretable are *also* the entities governed by the formal laws that define the system. In the subsymbolic paradigm, this is no longer true. The semantically interpretable entities are *patterns of activation* over large number of units in the system, whereas the entities manipulated by formal rules are the individual activations of cells in the network. The rules take the form of activation-passing rules, which are essentially different in character from symbol-manipulation rules.

Now, what I'm talking about here is the particular kind of connectionist system in which what I just said is true: concepts are represented by patterns of activity, rather than by the activations of individual elements in the network. (In the latter case, we would have a collapse of just the same kind that we have in symbolic paradigm.) So the subsymbolic paradigm involves connectionist systems using so-called *distributed representations*, as opposed to local representations. (The *PDP* books by Rumelhart, et al., and McClelland, et al., consider *distributed* connectionist systems; *local* connectionist systems are considered in Feldman and Ballard 1982, and Feldman, Ballard, Brown, and Dell 1985.)

Thus, in the subsymbolic paradigm the formal system description is at a lower level than the level of semantic interpretation; the level of denotation is higher than the level of manipulation. There is a fundamental two-layer structure to the subsymbolic paradigm, in contrast to the symbolic approach. The higher semantic level is not necessarily precisely formalizable, and the lower level is not “merely implementation” of a complete higher-level formalism. Both levels are essential: the lower level is essential for defining what the system *is* (in terms of activation passing), and the higher level is essential for understanding what the system *means* (in terms of the problem domain).

## 2 The subsymbolic level

I shall now characterize the subsymbolic level in more detail. Cognition looks quite different at this level than at the symbolic level. In section 3, where higher-level descriptions of connectionist systems will be considered, we will see some of the characteristics of the symbolic level emerging.



## 2.1 Subsymbolic computation

At the fundamental level in subsymbolic systems we have a collection of dynamical variables. There are two kinds of variables: an activation value for each of the units and a connection strength for each of the links. Typically, both kinds of variables are continuous. The rules that define these systems are activation-passing rules and connection-strength-modification rules. Typically these are expressed as differential equations (although they are simulated with finite difference equations). The differential equations are typically not stochastic, but stochastic versions will enter briefly later.

The computational role of these two kinds of equations are as follows. The activation-passing rules are in fact inference rules—not logical inference rules, but statistical inference rules. And the connection-strength-modification rules are memory-storage and learning procedures. These points will be expanded shortly.

Because the fundamental system is a dynamical system with continuously evolving variables, the subsymbolic paradigm constitutes a radical departure from the symbolic paradigm; the claim, in effect, is that *cognition should be thought of taking place in dynamical systems and not in digital computers*. This is a natural outcome of the neurally-inspired (rather than mentally-inspired) conception of computation.

The relation between the subsymbolic formalism and psychological processing is in part determined by the time constants that enter into the differential equations governing activation and connection-strength modification. The time required for significant change in activation levels is on the order of 100 milliseconds; the time it takes for a connection strength to change appreciably is much longer (say, on the order of a minute). Thus, for times less than about 100 milliseconds, what we're talking about is a single equilibration or "settling" of the network; all the knowledge embedded in the connections is used in parallel. On this time scale, we have parallel computation. When we go beyond this, to cognitive processes that go on for several seconds (such as problem solving and extended reasoning), then we're talking about multiple settlings of the network, and serial computation. This is the part of cognition for which serial symbolic descriptions, such as Newell and Simon's General Problem Solver, provide a fairly good description of the coarse structure. The claim of the subsymbolic paradigm is that the symbolic description of such processing is an approximate description of the global behavior of a lot of parallel computation. Finally, if we go to still longer time scales (on the order

of a minute), then we have adaptation of the network to the situation it finds itself in.

Let me summarize the contrasts between the symbolic and subsymbolic approaches, viewed at the fundamental level. In the subsymbolic paradigm we have fundamental laws that are differential equations, not symbol-manipulation procedures. The systems we are talking about are dynamical systems, not von Neumann machines. The mathematical category in which these formalisms live is the continuous category, not the discrete category; so we have a different kind of mathematics coming into play. The differences are dramatically illustrated in the way memory is modeled in the two formalisms. In a von Neumann machine, memory storage is a primitive operation (you give a location and a content, and it gets stored), memory retrieval is likewise primitive. In subsymbolic systems, by contrast, these processes are quite involved—they're not primitive operations at all. When a memory is retrieved, it is "addressed" by its *contents*: a fragment of a previously-instantiated activation pattern is put into one part of the network (by another part of the network), and the connections fill out the remainder of that previously-present pattern. This is a much more involved process than a simple "memory fetch". Memories are stored in subsymbolic systems by adjusting connection strengths such that the retrieval process will actually work—and this is no simple matter.

## 2.2 Subsymbolic inference and the statistical connection

At the fundamental level of the subsymbolic formalism, we have moved from thinking about cognition in terms of discrete processes to thinking in terms of continuous processes. This means that different mathematical concepts apply. One manifestation of this, in computational terms, is the claim that inference should be construed not in the logical sense but rather in the statistical sense—at least at the fundamental level of the system. (We will see below that, *at higher levels*, certain subsymbolic systems do perform logical inference.)

I have encapsulated this idea in what I've called the *statistical connection*:

The strength of the connection between two units is a measure of the statistical relation between their activity.

The origins of this principle are easily seen. The relationship between statistics and connections was represented in neuroscience by Hebb's (1949) principle: a synapse between two neurons is strengthened when



both are active simultaneously. In psychology, this relation appeared in the notion of "strength of association" between concepts, an important precursor of connectionist ideas (although, since this involved statistical associations between *concepts*, it was not itself a subsymbolic notion).

From a physical point of view, the statistical connection is basically a tautology, since if two units are strongly connected, then when one is active the other is likely to be too. But from a computational point of view, the statistical connection has rather profound implications for AI and for symbolic computation. Activation passing is now to be thought of as statistical inference. Each connection represents a *soft constraint*; and the knowledge contained in the system is the set of all such constraints. If two units have an inhibitory connection, then the network has the knowledge that when one is active the other ought not be; but that is a soft constraint that can easily be overridden by countermanding excitatory connections to that same unit (if those excitatory connections come from units that are sufficiently active). The important point is that soft constraints, any one of which can be overridden by the others, *have no implications singly*; they only have implications collectively. That's why the natural process for using this kind of knowledge is *relaxation*, in which the network uses all the connections at once, and tries to settle into a state that balances all the constraints against each other. This is to be contrasted with *hard constraints*, such as rules of the form "If A, then B", which can be used individually, one at a time, to make inferences serially. The claim is that using soft constraints avoids the brittleness that hard constraints tend to produce in AI. (It is interesting to note that advocates of logic in AI have, for some time now, been trying to evade the brittleness of hard constraints by developing logics, such as non-monotonic logics, in which all of the rules are essentially used *together* to make inferences, and not separately; see, for example, *Artificial Intelligence*, 1980.)

To summarize: In the symbolic paradigm, constraints are typically hard, inference is logical, and processing can therefore be serial. (One can try to parallelize it, but the most natural approach is serial inference.) In the subsymbolic paradigm, constraints are soft, inference is statistical, and therefore it is most natural to use parallel implementations of inference.

### 3 Higher-level descriptions

Having characterized the subsymbolic paradigm at the fundamental, subsymbolic level, I would now like to turn to higher-level descriptions of these connectionist systems. As was stated above, according to the subsymbolic paradigm, serial symbolic descriptions of cognitive processing are approximate descriptions of the higher-level properties of connectionist computations. I will only be able to sketch this part of the story—pointing to published work for further details. The main point is that interesting relations *do* exist between the higher-level properties of connectionist systems and mental structures, as they have been formalized symbolically. The view of mental structures that emerges is strikingly different from that of the symbolic paradigm.

#### 3.1 The best-fit principle

That crucial principle of the subsymbolic level, the statistical connection, can be reformulated at a higher level as what I call the *best-fit principle*:

Given an input, a connectionist system outputs a set of inferences that, as a whole, give a best fit to the input, in a statistical sense defined by the statistical knowledge stored in the system's connections.

In this vague form, this principle may be generally true of connectionist systems. But it is exactly true in a precise sense, at least in an idealized limit, for a certain class of systems that I have studied in what I call *harmony theory* (Smolensky 1983, 1984a, 1984b, 1986a, 1986b, 1986c; Riley and Smolensky 1984).

To render the best-fit principle precise, it is necessary to provide precise definitions of "inferences", "best fit", and "statistical knowledge stored in the system's connections". This is done in harmony theory, where the central object is the "harmony function"  $H$ , which measures, for any possible set of inferences, the goodness-of-fit to the input with respect to the soft constraints stored in the connection strengths. The set of inferences with the largest value of  $H$  (that is, with the highest harmony) is the best set of inferences, with respect to a well-defined statistical problem.

Harmony theory basically offers three things. It gives a mathematically precise characterization of a very general statistical inference problem that covers a great number of connectionist computations. It tells how that problem can be solved using a connectionist network

with a certain set of connections. And it provides a procedure by which the network can learn the correct connections with experience. I will comment briefly on each of these three elements, to give the flavor of the form that the best-fit principle takes, and to prepare the way for the remaining remarks on higher-level properties of connectionist computation.

Harmony theory analyzes systems that confront the following statistical inference task: If we give the system some features of an environmental state, it should infer values for unknown features. An example I will consider in the next section concerns reasoning about a simple electric circuit. Given a change in the value of some circuit feature (say, an increase in the value of some resistor), then what happens to the other, "unknown" features—the currents and voltages, say? This general task is what I call *the completion task*.

In response to a completion problem, the system is supposed to give the maximum-likelihood set of inferred values, with respect to a probability distribution maintained internal to the system as a model of the environment. In other words, the system maintains a probability distribution that represents the likelihoods of various events' occurring in the environment; and it should give as its output the maximum-likelihood set of values for the unknowns.

What model of the environment—what probability distribution—is the system supposed to use? Here harmony theory adopts a principle commonly used in statistical inference: The system should use the probability distribution with minimal informational content that is consistent with the statistical constraints that the system observes holding in the environment.

Having specified the inference problem in this way, we can now draw some conclusions. The first result says that the minimal information distribution can actually be computed as follows. The probability of a set of inferred values  $\mathbf{x}$  is proportional to the exponential of a particular function:

$$\text{prob}(\mathbf{x}) \propto e^{\sum_{\alpha} \lambda_{\alpha} f_{\alpha}(\mathbf{x})}$$

This function has one parameter  $\lambda_{\alpha}$  for each statistical constraint  $\alpha$  observed in the environment. (The function  $f_{\alpha}$  has value 1 when constraint  $\alpha$  is satisfied, and 0 otherwise). It turns out that the maximum-likelihood completions that the system is supposed to give as answers to questions can be computed from a simpler distribution that has a

quadratic function in the exponent, but which uses auxiliary variables to achieve the simplification:

$$\text{prob}(\mathbf{x}, \mathbf{y}) \propto e^{H(\mathbf{x}, \mathbf{y})}$$

This quadratic function  $H$  measures the internal consistency of a set of inferred values, with respect to these constraint parameters  $\lambda_{\alpha}$ ; I called it  $H$  because it turns out to play the mathematical role of the Hamiltonian of a statistical-mechanical system. That's where the name "harmony" comes from:  $H$  measures the internal consistency of states of the system.

I call this first result the *competence theorem* because it explicitly characterizes how the system ideally ought to behave. The next result, the *realizability theorem*, describes how to instantiate this competence in a performance system—an actual computing device—the behavior of which obeys the competence theorem in suitably ideal circumstances, but which in real circumstances exhibits performance that deviates from the ideal competence. By creating one computing element for each of the given and to-be-inferred variables in the set  $\mathbf{x}$ , and one for each of the auxiliary variables in the set  $\mathbf{y}$ , and using the parameters  $\lambda_{\alpha}$  to determine connection strengths, a connectionist network can be built which can compute the maximum-likelihood completions by a stochastic relaxation method. The units in this harmony network are stochastic processors—the differential equations defining the system are stochastic. There is a system parameter called the *computational temperature* that governs the degree of randomness in the units' behavior: it starts out high, at the beginning of the computation when there is a lot of randomness in the network, but then is lowered during computation until eventually the system "freezes" into an answer. In the idealized limit, where the system has unlimited relaxation time, the network converges with probability 1 to the correct answer, as characterized by the competence theorem. (The stochastic relaxation process is *simulated annealing*, as in the Boltzmann machine; see Hinton and Sejnowski 1983. For the historical and logical relations between harmony theory and the Boltzmann machine, see Rumelhart, et al. 1986, p. 148, and Smolensky 1986a.)

The third result is a learnability theorem. It says that, through a statistical sampling of the environment, the values of the parameters  $\lambda_{\alpha}$  (that is, the connection strengths) required by the competence theorem can be computed by relaxation. That is, the parameters start off with some initial values ("genetically" selected or randomly assigned)

which are then gradually tuned through experience to become the correct ones for the given environment.

### 3.2 Productions, sequential processing, and logical inference

A simple harmony model of expert intuition in qualitative physics was described in Riley and Smolensky (1984) and Smolensky (1986a, 1986c). The model answers questions like: "What happens to the voltages in this circuit if I increase this resistor?" Higher-level descriptions of this subsymbolic problem-solving system illustrate several interesting points.

It is possible to identify *macro-decisions* during the system's solution of a problem; these are each the result of many individual micro-decisions by the units of the system, and each amounts to a large-scale commitment to a portion of the solution. These macro-decisions are approximately like the firing of production rules. In fact, these "productions" "fire" at different times, in essentially the same order as in a symbolic forward-chaining inference system. One can measure the total amount of order in the system, and see that there is a qualitative change in the system when the first micro-decisions are made—the system changes from a disordered phase to an ordered one.

A corollary of the way this network embodies the problem-domain constraints, and the general theorems of harmony theory, is that the system, when given a well-posed problem and unlimited relaxation time, will always give the correct answer. Thus, under that idealization, the *competence* of the system is described by *hard* constraints: Ohm's law, Kirchhoff's laws, and so on. It is as if the system had those laws written down inside it. However, as in all subsymbolic systems, the *performance* of the system is achieved by satisfying a large set of *soft* constraints. What this means is that if we go outside of the ideal conditions under which hard constraints seem to be obeyed, the illusion that the system has hard constraints inside it is quickly dispelled. The system can violate Ohm's law if it has to; but if it doesn't have to violate the law, it won't. Thus, *outside the idealized domain of well-posed problems and unlimited processing time, the system gives sensible performance*. It isn't brittle in the way that symbolic inference systems are. If the system is given an ill-posed problem, it satisfies as many constraints as possible. If it is given inconsistent information, it doesn't fall flat, and deduce just anything. If it is given insufficient information, it doesn't just sit there and deduce nothing. Given limited processing time, the

performance degrades gracefully as well. Thus, the competence/performance distinction can be addressed in a sensible way.

Continuing the theme of physical analogies instead of computer analogies, we might think of this as like a "quantum" system that appears to be "Newtonian" under certain conditions. A system that has, at the micro-level, soft constraints, satisfied in parallel, *appears* at the macro-level, under the right conditions, to have hard constraints, satisfied serially. But it doesn't *really*; and if you go outside the "Newtonian" domain, you see that it has really been a "quantum" system all along.

### 3.3 The dynamics of activation patterns

In the subsymbolic paradigm, semantic interpretation occurs at the higher level of patterns of activity, not at the lower level of individual nodes. Thus an important question about the higher level is: How do the semantically-interpretable entities *combine*?

In the symbolic paradigm, the semantically-interpretable entities are symbols, which combine by some form of *concatenation*. In the subsymbolic paradigm, the semantically-interpretable entities are activation patterns, and these combine by *superposition*: activation patterns superimpose upon each other, the way that wave-like structures always do in physical systems. This difference is another manifestation of moving the formalization from the discrete to the continuous (indeed the linear) category.

Using the mathematics of the superposition operation, it is possible to describe connectionist systems at the higher, semantic level. If the connectionist system is purely linear (so that the activity of each unit is precisely a weighted sum of the activities of the units giving it input), it can easily be proved that the higher-level description obeys formal laws of just the same sort as the lower level: the subsymbolic and symbolic levels are *isomorphic*. Linear connectionist systems are, however, of limited computational power; and most interesting connectionist systems are nonlinear. However, nearly all are *quasi-linear*—that is, each unit *combines* its inputs linearly, even though the effect of this combination on the unit's activity is nonlinear. Further, the problem-specific *knowledge* in such systems is in the combination weights (that is, the *linear part* of the dynamical equations); and, in learning systems, it is generally only these linear weights that adapt. For these reasons, even though the higher level is not isomorphic to the lower level in nonlinear systems, there are senses in which the higher level *approx-*

*imately* obeys formal laws similar to the lower level. (For the details, see Smolensky 1986b.)

The conclusion here is a rather different one from that of the preceding subsection, where we saw how there are senses in which higher-level characterizations of certain subsymbolic systems approximate productions, serial processing, and logical inference. What we see now is that there are also senses in which the laws approximately describing cognition at the semantic level are *activation-passing laws*—like those at the subsymbolic level, but operating between “units” with individual semantics. These semantic-level descriptions of mental processing (which include *local* connectionist models) have been of considerable value in cognitive psychology (see, for example, McClelland and Rumelhart 1981; Rumelhart and McClelland 1982; Dell 1985). We can now see how these “spreading-activation” accounts of mental processing relate to subsymbolic accounts.

### 3.4 Schemata

One of the most important symbolic concepts is that of the *schema* (Rumelhart 1980). This concept goes back at least to Kant (1787/1929) as a description of mental concepts and mental categories. Schemata appear in many AI systems in the forms of frames, scripts, or similar structures; they are prepackaged bundles of information that support inference in stereotyped situations.

I will very briefly summarize work on schemata in connectionist systems reported in Rumelhart, Smolensky, McClelland and Hinton (1986; see also Feldman 1981, and Smolensky 1986a, 1986c). This work addressed the case of schemata for rooms. Subjects were asked to describe some imagined rooms using a set of 40 features like: has-ceiling, has-window, contains-toilet, and so on. Statistics computed from these data were used to construct a network containing one node for each feature, and containing connections computed from the statistical data by using a particular form of the statistical connection.

This resulting network can do inference of the kind that can be performed by symbolic systems with schemata for various types of rooms. For example, the network can be told that some room contains a ceiling and an oven, and then be given the question: What else is likely to be in the room? The system settles down into a final state, and the inferences contained in that final state are that the room contains a coffee cup but no fireplace, a coffee pot but no computer.

The inference process in this system is simply one of greedily maximizing harmony. To describe the inferences of this system on a higher level, we can examine the global states of the system in terms of their harmony values. How internally consistent are the various states in the space? It is a 40-dimensional state space, but various 2-dimensional subspaces can be selected and the harmony values there can be graphically displayed. The harmony landscape has various peaks; looking at the features of the state corresponding to one of the peaks, we find that it corresponds to a prototypical bathroom; others correspond to a prototypical office, and so on, for all the kinds of rooms subjects were asked to describe. There are no *units* in this system for bathrooms or offices: there are just lower-level descriptors. The prototypical bathroom is a pattern of activation, and the system's recognition of its prototypicality is reflected in the harmony peak for that pattern. It is a consistent, “harmonious” combination of features: better than neighboring points like one representing a bathroom without a bathtub, which has distinctly lower harmony.

During inference, this system climbs directly uphill on the harmony landscape. When the system state is in the vicinity of the harmony peak representing the prototypical bathroom, the inferences it makes are governed by the shape of the harmony landscape there. This shape is like a “schema” that governs inferences about bathrooms. (In fact, harmony theory was created to give a connectionist formalization of the notion of schema; see Smolensky 1986a, 1986c.) Looking closely at the harmony landscape we can see that the terrain around the “bathroom” peak has many of the properties of a bathroom schema: variables and constants, default values, schemata embedded inside of schemata, and even cross-variable dependencies. The system behaves as though it had schemata for bathrooms, offices, and so on, even though they are not “really there” at the fundamental level. These schemata are strictly properties of a higher-level description. They are informal, approximate descriptions—one might even say they are merely metaphorical descriptions—of an inference process too subtle to admit such high-level descriptions with great precision. Even though these schemata may not be the sort of object on which to base a formal model, nonetheless they *are* useful descriptions—which may, in the end, be all that can really be said about schemata anyway.



## 4 Conclusion

The view of symbolic structures that emerges from viewing them as entities of high-level descriptions of dynamical systems is quite different from the view coming from the symbolic paradigm. "Rules" are not symbolic formulae, but the cooperative result of many smaller soft constraints. Macro-inference is not a process of firing a symbolic production; rather it is a process of qualitative state change in a dynamical system, such as a phase transition. Schemata are not large symbolic data structures but rather the potentially quite intricate shapes of harmony maxima. Similarly, categories turn out to be attractors in dynamical systems: states that "suck in" to a common place many nearby states, like peaks of harmony functions. Categorization is not the execution of a symbolic algorithm but the continuous evolution of the dynamical system—the evolution that drives states into the attractors, to maximal harmony. Learning is not the construction and editing of formulae, but the gradual adjustment of connection strengths with experience, with the effect of slowly shifting harmony landscapes, adapting old and creating new concepts, categories, schemata.

The heterogenous assortment of high-level mental structures that have been embraced in this paper suggests that the symbolic level lacks formal unity. This is just what one expects of approximate higher-level descriptions, which, capturing different aspects of global properties, can have quite different characters. The unity underlying cognition is to be found not at the symbolic level, but rather at the subsymbolic level, where a few principles in a single formal framework lead to a rich variety of global behaviors.

If connectionist models are interpreted within what I have defined as the subsymbolic paradigm, we can start to see how mental structures can emerge from neural structures. By seeing mental entities as higher-level structures implemented in connectionist systems, we get a new, more complex and subtle view of what these mental structures really are. Perhaps subsymbolic systems can achieve a truly rich mental life.

## On the Nature of Theories: A Neurocomputational Perspective

10

Paul M. Churchland

1989

### 1 The classical view of theories

Not long ago, we all knew what a theory was: it was a set of sentences or propositions, expressible in the first-order predicate calculus. And we had what seemed to be excellent reasons for that view. Surely any theory had to be *statable*. And after it had been fully stated, as a set of sentences, what residue remained? Furthermore, the sentential view made systematic sense of how theories could perform the primary business of theories, namely, prediction, explanation, and intertheoretic reduction. It was basically a matter of first-order deduction from the sentences of the theory conjoined with relevant premises about the domain at hand.

Equally important, the sentential view promised an account of the nature of learning, and of rationality. Required was a set of formal rules to dictate appropriate changes or updates in the overall set of believed sentences as a function of new beliefs supplied by observation. Of course there was substantial disagreement about which rules were appropriate. Inductivists, falsificationists, hypothetico-deductivists, and Bayesian subjectivists each proposed a different account of them. But the general approach seemed clearly correct. Rationality would be captured as the proper set of formal rules emerged from logical investigation.

Finally, if theories are just sentences, then the ultimate virtue of a theory is truth. And it was widely expected that an adequate account of rational methodology would reveal why humans must tend, in the long run, toward theories that are true.

Hardly anyone will now deny that there are serious problems with every element of the preceding picture, difficulties we shall discuss below. Yet the majority of the profession is not yet willing to regard them as fatal. I profess myself among the minority that does so regard