Connectionism, Artificial Life, and Dynamical Systems: New approaches to old questions

Jeffrey L. Elman

Department of Cognitive Science
University of California, San Diego

Introduction

Periodically in science there arrive on the scene what appear to be dramatically new theoretical frameworks (what the philosopher of science Thomas Kuhn has called “paradigm shifts”). Characteristic of such changes in perspective is the recasting of old problems in new terms. By altering the conceptual vocabulary we use to think about problems, we may discover solutions which were obscured by prior ways of thinking about things. Connectionism, Artificial Life, and Dynamical Systems are all approaches to cognition which are relatively new and have been claimed to represent such paradigm shifts. Just how deep the shifts are remains an open question, but each of these approaches certainly seems to offer novel ways of dealing with basic questions in cognitive science.

While there are significant differences among these three approaches and some complementarity, they also share a great deal in common and there are many researchers who work simultaneously in all three. The goal of this chapter will be first, to trace the historical

roots of these three approaches in order to understand what motivates them; and second, to consider the ways in which they may either offer novel solutions to old problem, or to even to redefine what the problems in cognition are.

Historical context

When researchers first began to think about cognition in computational terms, it seemed natural and reasonable to use the digital computer as a framework for understanding cognition. This led to cognitive models which had a number of characteristics which were shared with digital computers: processing was carried out by discrete operations executed in serial order; the memory component was distinct from the processor; and processor operations could be described in terms of rules of the sort that were found in programming languages.

These assumptions underlay almost all of the most important cognitive theories and frameworks up through the 1970's, as well as a large number of contemporary approaches. These include the Physical Symbol System Hypothesis of Alan Newell and Herbert Simon; the Human Information Processing approach popularized by Peter Lindsay's and Donald Norman's text of the same name; and the Generative Linguistics theory developed by Noam Chomsky.

The metaphor of the brain as a digital computer was enormously important in these theories. Among other things, as the cognitive psychologist has Ulric Neisser pointed out, the computer helped to rationalize the study of cognition itself by demonstrating that it was

possible to study cognition in an explicit, formal manner (as opposed to the behavioral psychologists of the earlier era who argued that internal processes of thought were not proper objects of study).

But as research within this framework progressed, the advances also revealed shortcomings. By the late 1970's, a number of people interested in human cognition began to take a closer look at some of basic assumptions of the current theories. In particular, some people began to worry that the differences between brains and digital computers might be more important than hitherto recognized. In 1981, Geoff Hinton and Jim Anderson put together a collection of papers (*Parallel Associative Models of Associative Memory*) which presented an alternative computational framework for understanding cognitive processes. This collection marked a sort of watershed. Brain-style approaches were hardly new. Psychologists such as Donald Hebb, Frank Rosenblatt, and Oliver Selfridge in the late 1940's and 1950's, mathematicians such as Jack Cowan in the 1960's, and computer scientists such as Teuvo Kohonen in the 1970's (to name but a small number of influential researchers) had made important advances in brain-style computation. But it was not until the early 1980's that connectionist approaches made significant forays into mainstream cognitive psychology. The word-perception model of Dave Rumelhart and Jay McClelland, published in 1981, had a dramatic impact; not only did it present a compelling and comprehensive account of a large body of empirical data, but laid out a conceptual framework for thinking about a number of problems which had seemed not to find ready explanation in the Human Information Processing approach. In 1986, the publication of a two-volume collection, by Rumelhart and

McClelland, and the PDP Research Group, called *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, served to consolidate and flesh out many details of the new approach (variously called PDP, neural networks, or connectionism). Before discussing the sorts of issues which motivated this approach, let us briefly define what we mean by connectionism.

What is connectionism?

The class of models which falls under the connectionist umbrella is large and diverse. But almost all models share certain characteristics.

Processing is carried out by a (usually large) number of (usually very simple) processing elements. These elements, called nodes or units, have a dynamics which is roughly analogous to simple neurons. Each node receives input (which may be excitatory or inhibitory) from some number of other nodes, responds to that input according to a simple activation function, and in turn excites or inhibits other nodes to which it is connected. Details vary across models, but most adhere to this general scheme. One connectionist network is shown in Figure 1. In this network, the task is to take visual input and recognize words—in other words, to read.

There are several key characteristics worth noting. First, the response (or activation) function of the units is often nonlinear, and this nonlinearity has very important consequences for processing. Among other things, the nonlinearity allows the systems to respond under certain circumstances in a discrete, binary-like manner, yielding

crisp categorical behavior. In other circumstances, the system is capable of graded, continuous responses.

Second, what the system “knows” is, to a large extent, captured by the pattern of connections—who talks to whom—as well as the weights associated with each connection (weights serve as multipliers).

Third, rather than using symbolic representations, the vocabulary of connectionist systems consists of patterns of activations across different units. For example, to present a word as a stimulus to a network, we would represent it as a pattern of activations across a set of input units. The exact choice of representation might vary dramatically; at one extreme, a word could be represented by a single, dedicated input unit (thus acting very much like an atomic symbol); at the other extreme, the entire ensemble of input units might participate in the representation, with different words having different patterns of activation across a shared set of units.

Given the importance of the weighted connections in these models, a key question is, Who determines the weights? Put in more traditional terms, Who programs the networks? In early models, the connectivity was done by hand (and this remains the case for what are sometimes called “structured connectionist models). However, of the exciting developments which made connectionism so attractive to many was the development of learning algorithms by which the networks could be “programmed” on its own. In other words, the networks could themselves learn the values for the weights. Moreover, the style of learning was inductive: Networks would be exposed to examples of a target behavior (for example, the appropriate

responses to a set of varied stimuli). Through learning, the network would learn to adjust the weights in small incremental steps in such a way that over time, the network's response accuracy would improve. Ideally, the network would also be able to generalize its performance to novel stimuli, thereby demonstrating that it had learned the underlying function which related outputs to inputs (as opposed to merely memorizing the training examples).

(It should be noted that the type of learning described above—so-called "supervised learning"—is but one of a number of different types of learning that are possible in connectionist networks. Other learning procedures do not involve any prior notion of "correct behavior" at all. The network might learn instead, for example, the correlational structure underlying a set of patterns.) Now let us turn to some of the interesting properties of these networks, as well as the controversies which surround them.

Issues and controversies

Context and top-down processing

Early models of human information processing typically utilized a kind of "template matching" approach in many tasks, especially those for which pattern recognition played an important role. With the advent of the computer metaphor, this approach gave way to the use of rules for analysis. A central feature of such models was the notion of a strict flow of information processing which proceeded in what was

called a “bottom-up” manner (e.g., perceptual features of a stimulus being processed first, yielding a representation which is then passed on to successively “higher” stages of processing). These assumptions were challenged by a number of findings. For example, it was found that subjects’ ability to perceive a single letter (presented very rapidly on a computer screen) was influenced by the context in which it occurred: A letter could be better identified if it appeared in a real word, compared to appearing in isolation or embedded in a non-word letter string. In the domain of speech, it was discovered that the perception of ambiguous sounds was also affected by their context. Thus a sound which was perceptually midway between a “k” and a “g” would be heard by listeners alternatively as a “k” if what followed was “...iss” or as a “g” if followed by “...ift.” This seemed to be an example of top-down processing (since it was assumed that word knowledge came later in processing—so was “higher”—than perceptual processing, which was “lower”). In other types of experiments (for example, in which subjects listened to sentences containing potentially ambiguous words such as “bank”) it appeared that prior context often plays a powerful role in biasing comprehension. This was called a “context effect.”

These results suggested that so-called higher processing (in the above examples, knowledge of the lexicon or contextually established information) might influence supposedly lower processes (such as visual or auditory perception). Indeed, a burgeoning experimental literature soon suggested that the degree of context effects and top-down influences might be very extensive. Furthermore, there was growing evidence that the human cognitive system was able t process

at multiple levels in parallel, rather than being restricted (as was the digital computer) to executing a single instruction at a time.

The word-perception model of Rumelhart and McClelland, published in 1981, was one of the first attempts to account for these sorts of data in a model which was frankly parallel, highly interactive, and departed significantly from the old-style digital framework. Rumelhart and McClelland called this the "interactive activation model."

How to account for regularity? Rules or associations?

One of the basic challenges in cognitive science is how to account for behavior. If human behavior were either entirely random, or limited to a fixed repertoire of actions which could be memorized, then there would be little to explain. What makes the problem so interesting is that behavior is patterned, and is often productive (we generalize these patterns to novel circumstances).

A natural way to account for the patterned nature of cognition is to assume that underlying these behavior is a set of rules. Rules provide a compact and elegant way to account for the abstract and productive nature of behavior. Rules also offer a way to capture system-level properties—that is, there exists the possibility that rules can interact in complex ways.

The problem that can arise, however, is that some human behaviors are often only partially general and productive. A good example of this (and one which has been well-studied and the topic of considerable debate) is the formation of the past tense in English verbs.

The majority of English verbs form the past by adding the suffix “-ed” to the verb stem (e.g., “walk+ed”, “plant+ed”, etc.). This pattern might be captured by positing a rule for the regular past tense formation. At the same time, there are also other verbs whose past tense forms seem idiosyncratic: “go->went”, “sing->sang”, “hit->hit”. One way of dealing such apparently irregular forms is to suppose that they are simply exceptions which must be memorized by rote and are “listed” in some mental dictionary.

One strong piece of evidence in favor of this account was the observation that many children seem to go through several phases when they learn the past tense. Initially, at the stage where they know only a small number of verbs, some children begin by producing past tense forms correctly for both regulars and irregulars. Later, these children begin to make mistakes on the irregulars, treating them as if they were regular (“go->goed”). Ultimately, these children learn which verbs should be treated as regulars—obey the rule—and which are irregular—must be memorized. Thus, performance has a kind of U-shaped to it, starting off relatively good, getting worse, and finally becoming good again. The rule-based account explains this phenomenon by supposing that in early development children are simply memorizing all verbs. At some point, children discover the regularity in past tense formation and start using the rule—only they have not yet learned that the rule is not fully applicable to all verbs, and over-generalize to irregulars. The final stage is achieved when children learn which verbs are regular, and which are irregular.

A difficulty with this account is that although some of the irregulars appear to be truly exceptional, in the sense that they are

unique (e.g., “is->was”, “go->went”), other irregulars clump together in groups (e.g., “sing->sang”, “ring->rang”; or “catch->caught”, “teach->taught”; or “hit->hit”, “cut->cut”). These groups can not only be defined in terms of phonological similarity, but there is evidence that if confronted with a novel word which closely resembles one of the groups and asked to produce a past tense form, native speakers will sometimes produce an irregular: “pling->plang” (presumably, on analogy with “ring->rang”).

In 1986, David Rumelhart and James McClelland published the results of a connectionist simulation in which they trained a network to produce the past tense forms of English present tense verbs. Learning involved gradually changing the weights in the network so that the network’s overall performance improved. Rumelhart and McClelland reported that the network was not only able to master the task (though not perfectly), but that it also exhibited the same U-shape performance found in children. They suggested that this demonstrated that language performance which could be described as rule-governed might in fact not arise from explicit rules.

This paper generated considerable controversy, which continues to the present time. Steven Pinker and Alan Prince wrote a detailed and highly critical response in which they questioned many of the methodological assumptions made by Rumelhart and McClelland in their simulation, and challenged Rumelhart and McClelland’s conclusions. In fact, many of Pinker and Prince’s criticisms are probably correct, and were addressed in subsequent connectionist models of past tense formation.

A great many subsequent simulations have been carried out which correct problems in the Rumelhart and McClelland model. These simulations have in turn generated an on-going debate about new issues. In recent writings, Pinker and Prince have suggested that although perhaps a connectionist-like system is responsible for producing the irregulars, there are qualitative differences in the way regular morphology is processed which can only be explained in terms of rules. This has become known as the "dual mechanism" account. Proponents of a single mechanism approach argue that a network can in fact produce the full range of behaviors which characterize regular and irregular verbs.

Recursion and compositionality

Linguists have long-noted that human languages have a curious property. Consider the grouping of words that is called a Noun Phrase. Noun Phrases are things such as "John," "the old man," "the curious yellow cat," or "the mouse under the chair." Notice in this last example, "the chair" itself is a noun phrase. Thus, Noun Phrases may contain other Noun Phrases—and in fact, there is no principled limit to the degree of such self-embedding (e.g., "the mouse under the chair in the house that Jack built"). Such self-embedding is called "recursion," and refers to the possibility that a category may be defined in terms of itself, even if indirectly.

(Another way of thinking about recursion is in terms of circular definitions. We might define the category Noun Phrase as comprising many things, e.g., "Adj N", "det N", "N PP". This means that possible

Noun Phrases might be an **Adj**ective followed by a **N**oun (“old car”), or a **det**erminer followed by a **N**oun (“the table”), or a **N**oun followed by a **P**repositional **P**hrase (“book on the table”). Suppose Prepositional Phrases, in turn, are defined as made up of a Preposition (“on”) followed by a Noun Phrase (“the table”). This circular definition gives us recursion, since the definition of a Noun Phrase allows the possibility that it might be made up of other things, including Noun Phrases! This may seem strange, but it actually makes a lot of sense in language, because it explains why we can not only say things like “book on the table” but “book on the table in the room”, or “book on the table in the room in the house”, etc. More importantly, for many purposes grammatical rules treat all of these things as the same kind of entity—whey they are: Noun Phrases.)

The notion “compositionality” is closely related, and refers to the structural relationship between different elements which can arise when recursion occurs (as well as in other circumstances). Thus, one might say that the Noun Phrase “...spider on the chair” is composed of a noun which is modified by a prepositional phrase, which in turn is composed of a preposition followed by a noun phrase, etc. The tree diagram in Figure 2 is a typical notation used by linguists to capture such structural relationships.

The Rumelhart and McClelland verb learning simulation discussed above dealt with issues in morphology (e.g., verb inflections), but soon other connectionist simulations were developed which modeled syntactic and semantic phenomena. All of those simulations,

however, involved sentences of pre-specified (and limited) complexity. In 1988, Jerry Fodor and Zenon Pylyshyn wrote a paper in which they called attention to this shortcoming, and argued that the deficiency was not accidental. They claimed that connectionist models were in principle unable to deal with the kind of unbounded recursion or to represent complex structural relationships (constituent structure) in an open-ended manner. Fodor and Pylyshyn argued that since these phenomena are hallmarks of human cognition, connectionism was doomed and that only what they termed "classical" approaches (e.g, those based on the digital computer metaphor of the mind) would work.

The issues raised by Fodor and Pylyshyn have generated a large body of responses from connectionists, as well as further criticisms by proponents of the classical approach. Paul Smolensky provided one possible solution to Fodor and Pylyshyn's arguments. Another response involves using what are called "recurrent networks" (one version of a recurrent network is shown in Figure 3).

In a recurrent network, the internal (or "hidden") units feedback on themselves. This provides the network with a kind of memory. The form of the memory is not like a tape recorder, however; it does not literally record prior inputs. Instead, the network has to learn itself how to encode the inputs in the internal state, such that when the state is fed back it will provide the necessary information to carry out whatever task is being learned.

In 1991, David Servan-Schreiber, Axel Cleeremans, and Jay McClelland demonstrated that such a network could be trained on an artificial language which was generated by something called a Finite State Automaton (FSA; an FSA is the simplest possible digital computer; the most powerful type of computer is a Turing Machine). The recurrent network's task was simply to listen to each incoming symbol, one at a time, and to predict what would come next. In the course of doing this, the network inferred the more abstract structure of the underlying artificial language.

The FSA language, however, lacked the hierarchical structure shown in Figure 2. In another simulation reported in 1991, Jeff Elman showed that simple recurrent networks (Elman, 1990) could also process sentences which contained relative clauses—which involve hierarchical/compositional relations among different sentence elements. Later, Jill Weckerly and Elman reported that these networks also show the same performance asymmetry in processing different types of embedded sentences that humans do. Their networks, like people, find sentences such as (1) more difficult than sentences such as (2).

(1) The mouse that the cat that the dog scared chased ran away.

(2) Do you believe the report that the stuff they put in coke causes cancer is true?

(Both sentences are “hard” in the sense that they have complicated structures and somewhat unnatural. However, both are grammatically legal. The question is why do most people find the first sentence much harder to understand than the second? It can’t be just their structure, since they have nearly identical grammatical structures.)

Understanding exactly how such networks work is an interesting problem in its own right, and there are strong connections between their solutions and the dynamical systems we discuss below.

Other phenomena—and shortcomings

Since the early 1980’s, the connectionist paradigm has grown dramatically, and there are now models which attempt to answer questions in a wide range of areas. A few of these include:

• brain damage—if one “lesions” networks, does this have similar effects to those observed when humans suffer brain damage?

• reading—can networks be taught to read languages such as English, in which the mapping from letter to sound is not straightforwardly captured by a simple set of rules?

• development—can networks model the developmental process which occurs as children grow into adults?

• pattern completion—humans show an uncanny ability to fill-in or reconstruct information which is missing in many patterns (such as partially obscured faces); do networks have similar capabilities?

• philosophy—do connectionist models offer new ways of understanding philosophical concepts such as representation, information, belief, etc.?

Although the explosive interest in connectionism has resulted in what seem like genuinely new ways of dealing with long-standing problems in cognition, there are a number of problems for which connectionist models seem not to offer any direct solution. This is not to say that connectionism is wrong, simply that even if it is the right approach, it is not the whole story. Two other recent approaches may help in this regard: Artificial Life and Dynamical Systems. We turn first to Artificial Life.

Artificial Life

Connectionist models are powerful induction engines. That is, they learn by example, and use the statistics of those examples to drive learning. The attraction of the approach is that although learning is statistically driven, the outcome of that learning process is a system whose knowledge is generalizable to novel instances.

But there are several respects in which connectionist models seem deficient. Furthermore, these deficiencies are similar to those found in almost all artificial intelligence models. Here are three examples of the most striking shortcomings.

Disembodied intelligence. An enormous amount of the cognitive behavior of biological organisms is tightly coupled to the bodies in which the behavior is manifest. The way we think about space (for

example) is highly dependent on properties of our visual system. The way we think about ourselves and the world depends on how we experience it, and our experience is vastly different from that of a fish, or a bird, or a cat. Both traditional AI models and connectionist models tend to ignore the role of bodies; these models are in fact disembodied from the start.

Passive vs. active...the importance of goals. A neural network is a passive thing. Before it learns anything, left to its own devices, it does not do anything very interesting. Even after learning, few connectionist models display any behavior which is internally generated. In general, most connectionist and AI systems are reactive. Or if not, their goals are preprogrammed and determined by an outside agency (their programmer).

Yet even a very primitive biological organism displays goal-directed behavior. A snail, left alone on a table, will wander around in search of food. A baby, left to play in its crib, will spontaneously make noises, move around, and find things to amuse itself. Perhaps more importantly, when an external stimulus does impinge, the baby's reaction—and how much it processes that stimulus—depends on whether the object is interesting (a mother's face is vastly more interesting than a book). Put simply, biological organisms have an agenda.

Social vs. asocial cognition. Almost all AI and connectionist models view cognition as an essentially individual phenomenon: it occurs within the skull. Thus, these models focus on competencies such as chess, problem solving, pattern recognition, etc.

But as Ed Hutchins and many other culturally-oriented cognitive scientists have pointed out, in humans in particular, cognition is a social phenomenon. Placed alone in the Sahara (or even a more hospitable environment), an individual human would not display any of the behaviors we take as characteristically human (building computers, traveling to the stars, creating skyscrapers, etc.). A tremendous amount of our cognitive capacity depends on external artifacts: the physical and social structures we create in order to help us solve problems which could not be solved by one person alone.

Early Artificial Life: *Vehicles*

In 1984, the biologist Valentino Braitenberg published a short monograph called *Vehicles: Experiments in Synthetic Psychology*. The book consisted of 12 short "thought-experiments," in which Braitenberg invited the reader to imagine different primitive vehicles. Each vehicle was simply a block of wood with a pair of wheels in the rear, sensors (where headlights would be), and connections from sensors to the motors which drove each wheel. The exact nature of the sensors and their connection to the motors varied with each vehicle.

Braitenberg then considered how the different vehicles might behave when placed on a surface, possibly with other vehicles, and exposed to a stimulus, such as a light source. Some of the vehicles moved toward the light and then at the last minute, veered away. Others sped aggressively toward it and smashed in to it. Others circled, warily.

The chapters describing the various vehicles bore names such as “Love,” “Hate”, “Values”, “Logic.” And indeed, observed from outside, it was not difficult to imagine these vehicles as animate creatures, motivated by anger, or affection, and even more complex reasoning processes. Of course, the circuits inside were actually quite simple. Braitenberg’s point—and one which was of interest to many connectionists as well—is that simple systems often give rise to complex behaviors. His monograph is a powerful and graphic warning about the “attribution problem”: It is easy to attribute more than is warranted to a mechanism, especially if we already have preconceived notions about what mechanisms must underlie a given behavior.

The hardest kind of intelligence: Staying alive!

In 1987, a workshop (the first of what would become a series) was held at Los Alamos National Laboratory. Researchers from a wide range of disciplines met to exchange views on what was becoming a theme of growing interest in a number of different scientific communities: Artificial Life, or Alife, as it is more popularly known.

Although the methods and specific goals of the different subcommunities varied, there were also a number of perspectives which were shared. One idea was captured in Alife researcher Rik Belew’s comment that “the smartest dumb thing anything can do is to stay alive.” This accorded with ideas that had been developed by MIT roboticist Rodney Brooks. Brooks pointed out that the bulk of evolution had been spent getting organisms to the stage where they had useful sensory and motor systems; phenomena such as tool use,

agriculture, literature, and calculus represent only the most recent few "seconds" in the evolutionary clock. Brooks inferred from this that one should therefore concentrate on the hard job of building systems which have sound basic sensorimotor capacities; the rest, he suggested, would come quickly after that.

Emergentism

Another central insight which underlies much of the work in Alife is the notion of emergentism: many systems have behaviors—"emergent properties"—which result from the collective behavior of the system's components rather than from the action of any single component. Furthermore, these behaviors are often unanticipated (and in the case of artificial systems, unplanned or unprogrammed).

Examples of emergentism abound in nature. Indeed, our very bodies are a compelling example. Our 100 trillion or so cells interact in complex ways to produce coherent activity; no single cell—or even single group of cells—predicts or accounts for the highest level behavior. Social organizations are another example. No matter how autocratic the social structure involved, complex interpersonal dynamics usually give rise to group behaviors which could not have been predicted in advance. Many Alife researchers have come to the conclusion that emergentism is a hallmark of life. Artificial systems which exhibit emergentism (particularly behaviors which in some way resemble those of biological lifeforms) are especially interesting.

One example of what seems like a very simple system that displays interesting emergentism comes from what are called cellular

automata. These are systems which are built out of (usually) two-dimensional grids. At any given point in time, each cell in the grid can assume one of a small number of states; most simply, ON ("alive") or OFF ("dead"). At each tick of the clock, cells may change their state, according to a simple set of rules which usually depend on the states of a cell's eight immediate neighbors. A simple rule set which is the basis for a popular computer game (the "game of life") is the rule of "23/3": If a cell which is already alive has exactly 2 or 3 neighbors which are also alive, it survives to the next cycle; if a cell is not alive but has exactly 3 living neighbors, then it is "born;" in all other cases, a cell dies (or remains dead). If one seeds the initial population of cells with the pattern shown in Figure 4, something very striking happens. Over time, the pattern changes in a way which looks as if it is tumbling and deforming, and in the process glides down and to the right. This is called a "glider."

In addition to providing additional examples of what seem like biological behavior (e.g., some patterns reproduce copies of themselves), cellular automata can be viewed as complex mathematical objects, and their properties have been extensively studied. More recent work by Melanie Mitchell and Randy Beer has also investigated ways in which these systems can solve computational problems.

Evolution

Most artificial systems are built be some external being. Biological systems however, evolve. Furthermore, biological change has both a

random element, in the form of random genetic variations; and also a quasi-directed element, insofar as variants which are better adapted to their environment often produce more offspring, thus altering the genetic makeup of succeeding generations. This insight prompted computer scientist John Holland, in a 1975 monograph called *Adaptation in Natural and Artificial Systems*, to propose what he called the Genetic Algorithm, or GA. The GA was intended to capture some of the characteristics of natural evolution but in artificial systems.

Imagine, for example, that one has a problem which can be described in terms of a set of yes/no questions, and the goal is to find the right set of answers. The solution may be hard if there are interactions between the answers to particular questions; indeed, there may be multiple solutions, depending on how different questions are answered. The GA would model this by constructing an artificial "chromosome;" this is really just a vector of 1's and 0's, each bit position standing for the answer (1=yes, 0=no) to a different question. We begin with a population of randomly constructed chromosomes. Each chromosome represents a possible solution to the problem, so we can evaluate it to see how well it does. This determines the chromosome's "fitness" (on analogy with evolutionary fitness). A new generation is constructed by preferentially replicating those chromosomes which performed better, while at the same time randomly switching a small percentage of the 1's and 0's. (Better results are obtained if one also allows "cross-over" to occur between chromosomes, so that part of one chromosome might be combined with part of another.) The new generation is then tested, and its fitness

is used to determine the composition of a third generation. And so on until a best solution is found.

The GA has much in common with natural evolution. It is especially powerful when there are high-order interactions between the many different sub-parts to a problem. Although the original GA makes simplifying assumptions which are questionable (e.g., there is not the genotype/phenotype distinction found in nature) it is widely used in Alife, sometimes in conjunction with neural networks. The next simulation gives an example of this.

Goals

We noted above that connectionist models have a disturbing passivity which is quite unlike biological organisms. One could construct a network in advance in order to endow it with what looks like internally-generated behaviors, but this is hardly the solution found in nature. Instead, many of the behaviors we think of as goal-directed—such as the search for food, desire to mate and bear offspring, responses to danger—are usually adaptive, and they are the product of the evolution of our species and not taught to us. Thus, it seems more appropriate to model such behaviors using an evolutionary approach such as the GA. One can even do this with neural networks.

Nolfi, Elman, and Parisi demonstrated the evolution of a simple kind of goal-directed behavior in neural networks in the following way. The constructed an artificial 2-dimensional environment consisting of a 10x10 grid of cells. A small number of cells contained food. A population of artificial organisms were then “tested” in this

environment. Each organism was a simple neural network. Two input units provided the animal with information about the smell (direction and strength) of nearby food; input units were connected to a bank of seven hidden units (the animals "brain"), which projected to two motor output units. These two motor units allowed the animal to move forward, turn left or right, or pause. At the outset, each of the 100 animals in the first generation had random weights on the connection strengths, so the animals' behavior was disorganized. In some cases, animals would simply stand still for their whole lifetime. In other cases, animals would march forward until they fell off the edge of the world. This behavior was hardly surprising, given the random nature of the initial connections. Occasionally, an animal would stumble on a food, in which case he would eat it. At the end of an animal's life, it died. However, if by chance it had wandered across some food (most did not), then it was able to give birth to offspring. These offspring were copies of the parent, except in a few cases random weight changes were introduced.

After 50 generations, very different looking individuals had evolved. Placed into a world, an animal would head directly for the nearest food, ingest it, and then proceed to the next food; in short order, the animal gathered up all the food in its world. Viewed from above, the apparently deliberate and very efficient hunting for food certainly looked goal-driven. (But mindful of Braitenberg's vehicles, we must remember that these behaviors were generated by relatively small neural circuits.)

What are the limits?

The Alife approach is still fairly young, and much of the work has a preliminary character. As a corrective to previous modeling approaches, there is no question that the Alife perspective is valuable. The emphasis on emergentism, the role of the environment, the importance of an organism's body, the social nature of intelligence, and the perspective offered by evolution are notions which go well beyond Alife. Further, by trying to understand (as Chris Langton has put it) life, not necessarily as it is, but as it *could be*, we broaden our notion of what counts as intelligent behavior. This expanded definition may in turn give us fresh ways of thinking about the behavior of more traditional (biological) lifeforms. But it is also clear that Alife has a long way to go. As is true of many modeling frameworks, the bridge between the initial "toy" models and more complete and realistic models is a difficult one to cross.

Cognition as a Dynamical System

At the outset, it was pointed out that the digital framework, "mind as computer," has permeated work in cognition until recent times, and that connectionism can be understood at least in part as an alternative which views "mind as brain." The digital framework has also had a profound impact on the way we think about computation and information processing. Much of the formal work in learning theory, for examples, draws heavily on results from computer science.

Interestingly, there is one subset of researchers who have not adopted the digital framework; these are people who study motor

activity. Researchers such as Michael Turvey and Scott Kelso (to name only two prominent scientists from a large and active community) have instead used the tools of dynamical systems to understand how motor activity is planned and executed. This seems natural, given that motor activity has a dynamical quality which is difficult to ignore. For example, when we walk, or run, or ski our limbs move in a rhythmic but complex manner and involve behaviors which change over time (and hence, are dynamic). More recently, scientists from various other domains in cognitive science have also begun to explore the dynamical systems framework as an alternative thinking about cognition in terms of digital computers.

What is a dynamical system? Most simply, it is a system which changes over time according to some lawful rule. Put this way, there is very little which is *not* a dynamical system (including digital computers)! In practice, dynamical systems must also be characterized in terms of some set of components which have states, and the components must somehow belong together. (In other words, my left foot and the Coliseum in Rome do not constitute a natural system—unless perhaps my left foot happens to be kicking stones in the Coliseum.) The goal of dynamical systems is to provide a mathematical formalism which can usefully characterize the kinds of changes which occur in such systems.

There are a number of important constructs which are important in dynamical systems theory. For instance, having identified the parts of a system which are of interest to us (e.g., the position of the jaw, tongue, and lower lip), we can assign numeric values to these entities' current state. We can then use (in this example) a three-

dimensional graph (one axis each for jaw, tongue, and lower lip position) to visualize the way in which all of these components change their state over time. This three-dimensional representation is often called the "state space". If we are interested in a formal characterization how the system changes over time, then this leads us to using differential equations (such equations tell us how things change over time) to capture the way in which the variables' values evolve over time, and in relation to one another. A final example of a construct used by dynamical systems theory is the notion of an "attractor." An attractor is a state toward which, under normal conditions, a dynamical system will tend to move (although it may not actually get there). A child on a playground swing constitutes a dynamical system with an attractor that has the child and swing at rest in the bottom vertical position. The swing may oscillate back and forth if the child is pushed or pumps her legs, but there is an attracting force which draws the child back toward the rest position. The goal of a dynamical systems analysis of this situation would be to describe the behavior of the system using mathematical equations which tell us how the state of the system (e.g., the position of the child at any given moment) changes over time.

The dynamical hypothesis for cognition

Swings and springs and fluids in motion are obvious domains in which a dynamical perspective applies. How might it apply to cognition?

This question was raised at a conference at Indiana University in 1991; the topic of the conference was *Mind as Motion*, which is also the

title of the book (edited by Robert Port and Tim van Gelder) that was subsequently produced from the conference. In their introduction to the book, Port and van Gelder list several reasons why one should think of cognition as a dynamical system. Some of these include the following.

Cognition and time. Cognitive behaviors are not atemporal; they exist and unfold over time. Dynamical models take as their goal the specification of how and a system's states changes occur. Thus, any useful account of cognitive behavior must necessarily explain such temporal changes; and this is precisely what dynamical models take as their goal.

Continuity in state. Although computational models can be formulated which model change as a serial movement from one discrete state to another, natural cognitive systems often change in a continuous manner in which there is never any state which is discretely separable from the next. As we listen to a sentence, for example, our understanding of the words we hear builds up gradually (sometimes, with expectations about an upcoming word even before we hear it, or other times delayed until the word is past). The meaning of the sentence as a whole unfolds gradually, rather than occurring all at once at the end. (Henry James is known for sentences which go on interminably, sometimes over pages; yet a reader need not wait till the very end till she understands the sentence.) It is a natural property of dynamical systems that they model the continuously changing nature of states; indeed, sometimes there never is an "end" state.

Multiple simultaneous interactions. One of the problems which confronted the Human Information Processing framework, described in the earlier section on connectionism, was how to deal with situations in which many things interacted in complex ways. The problem was not only conceptual (how to think about such systems) but also computational: the digital computer carries out only one (or at best, a very few) instructions at one time—very fast, perhaps, but only one at a time. As a system grows in complexity, the interactions between the system's parts can grow exponentially, rapidly outstripping the possibility of modeling behavior using a digital machine. Dynamical systems, on the other hand, allow us to focus (that is, model and formalize) precisely what is otherwise difficult: the many simultaneous interactions in a system which affect its overall behavior.

Self-organization and the emergence of structure. We have already talked about emergentism; self-organization is the ability of a system to develop structure on its own, simply through its own natural behavior. Dynamical systems provide a good framework for understanding how and why such characteristics emerge.

For example, neither the stripes on a zebra and the stripe-like patches of visual cortex (which separately process input from left- and right-eye; these are known as ocular dominance columns) are likely to be programmed in from birth. Instead, these stripes emerge out of initially relatively uniform patterns which, over time, involve dynamic interactions (between substances controlling skin pigmentation in the zebra, and neurons which are stimulated by left and right eyes in the case of visual cortex). The underlying dynamics were first described by the famous mathematician and cryptographer

Alan Turing, and are known as Reaction-Diffusion equations—and they are dynamical equations.

The dynamical hypothesis for cognition is quite new. The body of work relating to motor behavior is the most substantial; far less has been done in realms of higher cognition. One example of how it might be applied to the case of language comes from work in 1995 by Janet Wiles, Paul Rodriguez, and Jeff Elman, who used a dynamical systems analysis to analyze a recurrent neural network that was trained on a counting task.

In this task, a recurrent network was trained on an artificial language in which every "sentence" consisted of some number of *a*'s and *b*'s; grammatical sentences had the form *anbn*. In other words, legal sentences had some number (*n*) of *a*'s, followed by exactly the same number of *b*'s; for example: *aaabbb, ab, aaaaaabbbbbb* are all legal, whereas *abb* and *aaaaabbbb* are not. What makes this language at all of interest is that although it is very simple, it has certain properties which resemble human language, and are known to be hard (specifically, it requires some sort of "stack" or memory device for remembering how many *a*'s there were, in order to known how many *b's* to expect). After training, the network was able to process not only strings it had seen, but to generalize its knowledge to strings that were longer than any encountered during training. Given that the network does not possess a stack of the sort familiar to computers, how did it solve this problem?

The network had only two hidden units, so these two units together define the relevant state for the network. During processing,

each unit's activation is represented by some number between 0 and 1; together, we can treat the two numbers as corresponding to the *x* and *y* axes of a 2-dimensional state space. Thus, over time, the networks internal space will "move" through this *x*-*y* plane. Using dynamical systems analysis, Rodriguez and his colleagues discovered that—like the child on the swing—the network had attractors. Metaphorically (and simplifying a bit), one can think of the dynamics as follows, imagining that instead of a network we are working with our child on the swing: When the network was listening to the initial (*a*) portion of a sentence, it was as if the child were being pulled back; the more *a*'s, the further back the child is drawn. When the *b*'s start coming in, the child is released and allowed to swing, eventually coming to rest. Just how long it takes the swing to return to the rest position depends on how far back the child was pulled. The network worked in a similar fashion. This is an interesting example, because it suggests a way to use dynamical systems to tackle problems (such as processing language) which are central to cognition.

Conclusion

These three approaches are represent attempts to deal with the short-comings of the cognitive models. The connectionist framework is primarily concerned with biological implementation, and with problems in learning and representation. Can an inductive learning procedure discover abstract generalizations, using only examples rather than explicitly formulated instruction? And how do the resulting knowledge structures capture the generalizations? Are there important

differences between the ways in which networks represent generalizations and traditional rule systems?

The focus of Alife is different. Alife rejects the view that cognition consists only of highly developed mental activities such as chess, and emphasizes the intelligence which is required simply to survive. The role of adaptation and the role of evolution as achieving adaptation are valued; and a theme common to much of the work in Alife is the emergence of structure and behaviors which are not designed, but rather the outgrowth of complex interactions.

The dynamical systems approach also is concerned with interaction and emergentism; more generally, it can be viewed as a mathematical framework for understanding the sort of emergentism and the high-order interactions which are found in both connectionist and artificial life models. Dynamical systems also reflects a deeper commitment to the importance of incorporating time into our models.

The three approaches share much in common. They all reflect an increased interest in the ways in which paying closer attention to natural systems (nervous systems; evolution; physics) might elucidate cognition. None of the approaches by itself is probably complete; but taken together, they complement one another in a way which we can only hope presages exciting discoveries yet to come.