HIGH-PERFORMANCE COMPUTING AND NEURAL AND PHYSIOLOGICAL PROCESSES

Chaired by Jay Samuels, *University of Minnesota*

If we compute faster, do we understand better?

JAMES A. ANDERSON

Brown University, Providence, Rhode Island

and

JEFFREY P. SUTTON

Harvard Medical School, Charlestown, Massachusetts

Practitioners of cognitive science, “theoretical” neuroscience, and psychology have made less use of high-performance computing for testing theories than have those in many other areas of science. Why is this? In high-performance scientific computation, potentially billions of operations must lead to a trustable conclusion. Technical problems with the stability of algorithms aside, this requirement also places extremely rigorous constraints on the accuracy of the underlying theory. For example, electromagnetic interactions seem to hold accurately from atomic to galactic scales. Large-scale computations using elementary principles are possible and useful. Many have commented that the behavioral and neural sciences are largely pretheoretical. One consequence is that we cannot trust our few theories to scale well for a very good reason: They don’t. We have some quite good computational theories for single neurons and some large-scale aspects of behavior seem to be surprisingly lawful. However, we have little idea about how to go from the behavior of a single neuron to the behavior of the 10^{11} neurons involved when the brain actually does something. Neural networks have offered one potential way to leap this enormous gap in scale, since many elementary units cooperate in a neural network computation. As currently formulated, however, neural networks seem to lack essential mechanisms that are required for flexible control of the computation, and they also neglect structure at intermediate scales of organization. We will present some speculations related to controllability and scaling in neural networks.

This meeting is a celebration of high-performance computation. We are using a marvelous, costly tool, a supercomputer. Justification for supercomputer resources to those who provide the funds seems to have followed the *Field of Dreams* scenario: “If you give them cycles, they will compute.” This motto accurately describes the behavior of many of the users of high-performance computing. However, it has been less true in the behavioral sciences.

A few years ago, one of us (J.A.A.) was on a supercomputer resource allocations committee for medium-sized allocations, as their token behavioral scientist/neurobiologist/neuroscientist/social scientist.

There was not much to do. For most committee sessions, there were no proposals to review at all, or, if there was a complaint about lack of work, an evaluation was requested

for a proposal on something like parallel FORTRAN compilers. The theoretical neuroscience/cognitive science community was not using the big machines. The bulk of the proposals were in physics, engineering, or chemistry. In most of these areas, the proposal titles sounded very familiar. Often, a proposal with such a title could have been written at any time over the past few decades. The conclusion was that users of supercomputers remain the traditional ones: physicists, engineers, and some chemists. Since traditional users by now have decades of experience with supercomputers, the machines themselves and their software are tuned for a particular class of problems. It was a not uncommon occurrence for new users, especially in the behavioral and social sciences, to be rated low for allocations because they were, not surprisingly, making inefficient use of hardware and software designed for a different class of problem than the ones they proposed to investigate.

One possibility for the absence of other users was that a culture of high-performance scientific computer use had

Correspondence should be addressed to J. A. Anderson, Department of Cognitive and Linguistic Sciences, Brown University, Providence, RI 02912 (e-mail: james_anderson@brown.edu).

developed that was hostile to outsiders. Like all cultures, it is transparent to its inhabitants, who had developed a consensus regarding the "right" way to do things. Newcomers from other tribes were discouraged not because they were new, but because they were in error and did not sufficiently respect time-honored traditions, such as the following: "Thou shalt economize on cycles."

This culture of high-performance computing does exist and in fact is quite intolerant of outsiders. However, there are more interesting reasons why there is proportionally little use of high-performance machinery among theorists about behavior and neurobiology. The reasons fall into several classes.

1. We don't think the way computers do.
2. Behavioral theories don't obey the scaling relations physical theories do.
3. A complex behaving system may require flexibility, controllability, and feedback across levels of organization more than accurate learning. Behavioral problems may be different in kind from those in physics or chemistry.

"We Don't Think That Way": Impedance Matching

It was fascinating to see during service on the allocation committee how many cycles were involved indirectly with an important issue in cognitive science: how to understand and control complex systems. The evolution of computer software itself has provided a dramatic demonstration of this process in action. In a couple of human generations, we have progressed from dealing with the computer as it really is, an arithmetic and logic engine programmed in machine language, through assembly language, FORTRAN, powerful operating systems, and, now, the Macintosh and Windows 95, with icons, buttons, and sliders. More cycles are used in making the operating system suitable for human interaction than for actually computing anything. Underneath it all, however, the computer is still the same logical engine it has always been. All those cycles are devoted to rectifying the impedance mismatch between a logic processor and our kind of intelligence.

Another place where this mismatch appears is to be found in the fact that a significant fraction of the cycles that are used for computing even in physics and chemistry is used for making pictures. The primate cerebral cortex is about 50% devoted to visual processing. When complex data sets are turned into pictures, we can apply a powerful neural processor to the results even though we may not be able to describe in words exactly what we are seeing with understanding. Again, the computer is serving as an impedance matcher between a logic engine and a visual perceiver.

Humans Dealing With Abstractions

The way in which humans deal with abstractions is an interesting, controversial, and important problem in cognitive science. It also directly affects supercomputer use. As a simple test case, a few years ago, one of us (J.A.) became interested in the problem of how humans deal with

abstractions in mathematics and physics. This project at its beginning actually involved a supercomputer, the Cyber 205 at the late lamented John von Neumann supercomputer center in Princeton. We started by considering the most elementary interaction of humans with mathematics: the learning of arithmetic facts. There is one striking fact about arithmetic learning: It is hard, and it should be easy. Given the extraordinary ability of second and third graders to learn, it is remarkable that they have so much trouble learning a few hundred arithmetic facts.

Knowledge of the basic arithmetic facts is a memory task. If memory is partly or largely associative, as most believe, a problem called "associative interference" may arise, in which an item may have several potential associations that compete with each other. This problem is present in a strong form in the structure of arithmetic where the multiplicand 4 might be associated with all the 4s products: 4, 8, 12, 16, Studies of arithmetic errors in both children and adults have shown that many errors are associative in nature. That is, a common erroneous answer to "four times six" would be "twenty-eight."

We know that the concept of number as used by humans must contain the idea of magnitude. An eight-pound sack of potatoes acts more like a nine-pound sack than does a one-pound sack. In many real-world contexts, "eight" is more similar to "nine" than it is to "one." Children are exposed to this aspect of number from an early age.

A number of pertinent psychological experiments can be interpreted as supporting the idea that magnitude information is deeply embedded in how we interact with numbers. A dramatic example would be a series of reaction time experiments on number comparisons. Suppose that a subject is asked to verify the truth or falsity of an expression such as "87 is bigger than 86" or "87 is bigger than 17." The response time to make the decision is much faster for the second comparison (87 with 17) than the first (87 with 86). The effects observed are extremely large. Link's (1990) data show that response times when the numbers are very close to each other are nearly 600 msec; response times for more distant comparisons can drop below 400 msec.

Such an effect is sometimes called a *symbolic distance* effect, because what seems to be an abstract quantity is behaving like a concrete quantity. This particular response time pattern is also typical of what is seen when sensory magnitudes are compared: It takes much longer to decide which of two lights is brighter or which of two sounds is louder when they have nearly equal intensities than when they are very different in intensities. Observing similar effects with abstract symbols that have no necessary magnitudes attached to them is surprising.

One reasonable conclusion from this, and from many other experiments, is that the mental representation of number acts more like a sensory quantity than an abstract symbol.

Although this conclusion seems firm when children learn arithmetic, there is also ample evidence that the highest levels of creative thought in mathematics and physics

contain major components based much more on sensory and perceptual information than on abstract reasoning.

Several studies have shown that mathematicians and theoretical physicists also make extensive use of what might be called sensory or perceptual information in the most creative and intuitive aspects of their dealings with complex abstract structures. They specifically do not make much use of the formal manipulation of abstractions.

One of the most interesting discussions of mathematical thinking is found in Jacques Hadamard's 1949 book *The Psychology of Invention in the Mathematical Field*. Hadamard's comments have special weight, because he was a creative mathematician of considerable stature. In this book, Hadamard describes an informal survey that he took of North American mathematicians around 1940. He asked whether they thought in terms of sensory images or in terms of abstractions. Some of his findings are described thus:

The mental pictures of the mathematicians whose answers I have received are most frequently visual, but they may also be of another kind—for example kinetic. There can be auditive ones.... practically all of [the mathematicians]... avoided not only the use of mental words but also... the mental use of algebraic or any precise signs... they use vague images. There are two or three exceptional cases, the most important of which is the mathematician George D. Birkhoff, one of the greatest in the world, who is accustomed to visualize algebraic symbols and work with them mentally.... (Hadamard, 1949, p. 83)

Einstein commented in a famous passage in an Appendix to Hadamard's book that "the words or the language, as they are written or spoken, do not seem to play any role in my mechanism of thought" (p. 142).

It is worth thinking a bit about the implications of these quotes, which suggest that some of the most powerful scientific and mathematical intellects of the world do not think abstractly and logically but in visual images, sounds, and muscular movements. They think in this way in their most creative moments. They can and do think abstractly and logically when it comes time to confirm their results, to clean up the details, and to communicate their insights to others by using language.

Sensory-based thinking in the right mind can have great power. An impressive example of such thinking involves one of the basic insights behind the theory of special relativity. At the age of 16, Albert Einstein devised a powerful visual and kinesthetic thought experiment. There was no formal mathematics and no abstraction in the image. Einstein assumed that an observer was moving alongside an electromagnetic wave. His intuition was based on the image of a boat moving in the same speed and direction as an ocean wave. The wave would appear to those in the boat as a stationary hill of water (Figure 1). If you could travel at the same speed and in the same direction as an electromagnetic wave, you would observe a spatially varying, motionless, electric and magnetic field. Einstein knew that nothing like this had ever been seen to exist. Perhaps no one had ever seen such a wave because,

Figure 1. An example of the power of visual reasoning is given in this example from Albert Einstein. If a boat is traveling in the same direction as the waves, it will appear to be fixed in stationary hills of water. Similarly, if an observer could travel at the same speed as an electromagnetic wave, he/she would be surrounded by stationary electric and magnetic fields, yet such fields have never been seen. Why are these two situations so different?

unlike a water wave, it was impossible for an observer to travel at the same velocity as that of an electromagnetic wave. As Einstein (1949) put it in his *Autobiographical Notes*, "One sees that in this paradox, the germ of the special relativity theory is already contained" (p. 53).

These facts and speculations suggest that numbers, and, by extension, much mathematics, are only partially abstract but are also sensory, perceptual, analog, and non-verbal in their internal representation. A neural network model can be constructed that captures a small fragment of this idea if a good way can be found to represent a number so that the magnitude of a number can be directly represented in a pattern of activations of the units.

One suggestion, arrived at with some experimentation, is that a number can be represented for these elementary purposes as composed of two parts: a somewhat abstract portion corresponding to the individuality of a number (say its name) or its particular associations, or to its shape, for example, and another portion related to its magnitude. One approach to the "sensory" or "magnitude" part is based on the topographic maps of sensory magnitudes that are often found in cortical representations of sensory data. Suppose that somewhere in the brain an array of units exists to represent number magnitude. For convenience, assume that this array is in the form of a row of units. A "topographic map" can be formed by arranging it so that small numbers (low magnitudes) activate units at one end

Figure 2. A neural network number representation. Each dot is a computing element in a large state vector representing a number. One part of the state vector represents numerical magnitude by the spatial position of a "bar" of activity. The height of the bar suggests amount of activity of elements at that location.

of the map and large numbers (high magnitudes) activate units at the other (Figure 2). A similar representation for number has been used by others with good success in fitting experimental data (Dallaway, 1994; McCloskey & Lindemann, 1992).

If we assume this number representation, there are many ways to represent a multiplication problem. The technical details of the network and the learning algorithm are important, but less so than the number representation. The model described here was realized by a neural network called the BSB model. Briefly, this network is described formally as an autoassociative, energy-minimizing nonlinear dynamical system. The well-known Hopfield network is another example of this class of nonlinear dynamical systems. More complete explanations of both networks can be found in Anderson (1995).

The behavior of the nonlinear BSB neural network, like that of many other nonlinear systems that change with time, is dominated by a small number of stable states called "attractors." For the BSB network, these attractors are point attractors because they are specific unchanging activity patterns. More complex dynamical systems can show stable oscillations called "limit cycles."

One way to represent multiplication in this system was as the concatenation of smaller pattern vectors representing each of the three numbers: the two multiplicands and the product. This large pattern was learned by using one of the learning algorithms for correcting standard error, in this case, one called the "Widrow-Hoff" or "LMS" (least mean squares) algorithm. After learning, if a product was to be found, the part of the input pattern where the product was represented was set to zero, and the missing part of the pattern was filled in by the feedback connections.

These simulations and those of others using a similar data representation provided good matches for many of the experimental effects found in elementary arithmetic learning. Details can be found in Anderson, Spoehr, and Bennett (1994) and in Anderson (in press).

One conclusion, for example, is the suggestion that the human multiplication algorithm is something like, "The answer to a multiplication problem is (1) familiar, that is, a product for some elementary problem and (2) about the 'right size.'" Arithmetic fact learning and retrieval in humans seems to be a process that combines memory with estimation. Computation in the abstract sense does not really play a role in this process.

Predicting the Behavior of Learning Systems

Learning arithmetic facts is a memory task. Next, we will consider how to take the partly sensory-based number representation that we developed for fact learning and use it to generalize to what seem like "abstract" relationships such as "bigger" and "smaller."

By its nature, a system that must perform large-scale actions may be easy to understand at the global level. In this respect, the behavioral sciences may be a lot more like engineering—design toward a goal—than pure science. Neural structures exist to perform behavioral functions. Let us tell a fable.

Suppose we have a rat that we have trained to press a bar for a reward when a stimulus appears. The rat presses with its forepaw. We rearrange the apparatus (or the rat) so that the rat can no longer use its forepaw. The rat now presses the bar with its nose.

One way to model this experiment, and one that is not unknown among neural network modelers, is to have some representation of the stimulus connect through the network to drive the simulated motor neurons. But motor neurons are often a means to an end; that is, their discharge is to realize a goal. The goal can be achieved in many different ways. Sometimes the analysis in terms of goals can be simple compared with the immense complexity of trying to figure out what tens of thousands of motor neurons are doing in detail. For example, in our fable the new motor act has the same goal as the old one, but the rat is using its nose instead of its forepaw.

Biological systems often show surprisingly predictable gross behavior even though many millions of interacting neurons and muscle fibers are involved. In astrophysics, a white dwarf star is not actually doing anything but being itself. It has no function, just existence. The marvelous thing for physics is that the large-scale structure of a white dwarf can be understood in remarkable detail as the working out of general physical laws that apply at both the atomic and stellar scales. Such a situation is the prototypical place for the application of a supercomputer.

It is possible that shoveling all the information we have about neuron physiology at one end of a computer program would generate animal behavior at the other. We doubt this, however.

The question arises, how can simplicity be generated, given the complexity of the means giving rise to it? A nervous system is exceedingly complex, but the complexity has to be coordinated and controllable. This observation has two consequences: (1) Looked at in the right way,

animal behavior may sometimes be simple because it has to be simple to work properly; and (2) the requirement for flexible control may be a much more powerful design constraint than, for example, the formation of accurate input-output relationships in neural network learning.

Engineers understand this as a matter of course. Many of the most challenging problems in complex system design involve controllability. As only one example, many major compromises in automobile design in both efficiency and performance arise directly from the need to produce a car that is safe and tractable in the hands of drivers of varying skill levels under varying driving conditions.

Let us describe two neural network systems where simplicity of control is the major design goal, even at the sacrifice of accuracy.

The first example arose from design of a neural network that was capable of answering the totally trivial question, "Which is bigger, seven or eight?" The other is much less well developed and has to do with a modular cortical computational architecture.

Flexibility: Doing More Than You Learned

One of the cosmic issues in cognitive science is the long-running debate between those who believe that association of specific events and facts is the primary operation in mental computation and those who believe that symbol processing is primary. For example, Pinker and Prince (1988) have mentioned "the 'central dogma' of modern cognitive science, namely that intelligence is the result of processing symbolic expressions" (p. 74). Most linguists and many cognitive scientists would agree with this statement, though possibly with a few qualifications. Support for the claim that symbol processing is central to human cognition comes largely from the domain of human language behavior, where behavior can often be accurately described as operations on symbolic material. However, we saw that in what appears to be abstract human thought about mathematics, experiments suggest that perceptual and sensory mechanisms play a major role. Is it possible to reconcile these different approaches?

It was not until the recent popularity of neural networks that the view that cognition largely involved symbolic operations was seriously challenged. Some early neural network enthusiasts claimed that it was not necessary to have rules and symbolic operations at all; that a properly designed network would act as if it were rule governed after it had seen a number of examples, though in fact the rules were never explicitly formulated and used.

A neural network is capable of learning, though not easily, the facts involved in elementary arithmetic. However, there are terms even in elementary arithmetic that describe operations that seem at first a great deal like symbol processing.

The structures realized by neural networks have trouble handling general relationships. For example, consider the relationships "bigger" and "smaller." There are about 200 possible bigger-smaller relationships for single digits, each comparison having an associated truth or falsity. There are about 20,000 possible two-digit comparisons.

Clearly, comparisons are not learned individually. Comparison terms such as *bigger* and *smaller* seem to operate at a higher level of abstraction than the objects that they compare.

Interestingly, a model based on the sensory-based "magnitude" part of the number representation can generate "general" comparisons. The model provides an example of a system that might be called a "special purpose symbol." It acts somewhat like an abstraction, but in a limited domain, based directly on the data representation used in the network.

A network to do number comparisons might look something like this: We are given two numbers, and we wish to know which is bigger. Suppose we arrange a competition between the two numbers, using the nonlinear dynamics of the network such that the attractor corresponding to the biggest number would win; that is, the final state of the system would correspond to the bigger number.

The actual neural network model that we can use to compute "bigger" and "smaller" from the magnitude part of the representations of the two numbers is simple and depends on the ability to control the dynamics of the nonlinear dynamical system doing the computation. Consider the magnitude representation used in the number representation map; high numbers are on the other.

Suppose the term "bigger" becomes associated with a differential weighting of activities on the magnitude map so that the larger magnitude regions tend to be more active

In a subsequent comparison operation, the "bigger" representation dominates the system dynamics.

Figure 3. Because the number representation contains a part related to magnitude, it is possible to tell which of two numbers is larger (or smaller) by differentially weighting the state vector so as to increase the strength of large numbers. Proper use of nonlinear dynamics can then direct the state vector to the attractor state corresponding to the larger numbers. (See Anderson, 1995, for more details.)

than the smaller magnitude regions. There are many ways to do this. One would be multiplying all the unit activities by a larger value toward the large magnitude end, effectively increasing unit excitability, as is illustrated in Figure 3.

Then let the starting network activity pattern be the combination of the differentially weighted magnitude activities corresponding to the two numbers to be compared. As the system changes in time, the dynamics of the system will force the final state of the system to be in one or another of the two number attractor states. If we set the network parameters correctly, the differential weighting of the unit activities will arrange it so that the final attractor will be the larger number. When this model is implemented, it works nicely. Therefore, analog control of a continuous dynamical system combined with the discrete attractor states can be used to perform a very general computation. The only remaining trace of the underlying continuous system is the observation that the system generates a symbolic distance effect in the time required to get to the attractor because numbers near in magnitude are more evenly matched in dynamic properties than are distant numbers. Note that the seemingly more abstract part of the number code, the nonmagnitude part, plays no role in this "symbolic" computation.

Because the weighting patterns described above for the bigger-smaller comparisons are associated with general operations on the numbers—that is, operations that work on any pair of digits—it is possible to use them to suggest

the beginnings of a network "programming" system. Suppose that we ask the network to perform a novel task, such as, "Give me a number bigger than four and smaller than six." We can combine the "bigger" and "smaller" weighting patterns and the resulting pattern will cause "five" to win the competition for the attractor. The flexibility of human mental operations is at least as remarkable and important as the fact learning that underlies them. Capturing this flexibility requires extending neural network models in a way that allows effective control of the computation.

However, there is a consequence in building an "abstraction" by using the details of a sensory data representation. It makes "abstract" concepts concrete. Formal symbols as defined in mathematics are precise and fully general, but "symbols" as realized by a neural network using the kind of sensory-based data representation described here will be less general, more domain specific, and more likely to display arbitrary exceptions and special cases. However, human behavior in complex domains does contain such peculiarities.

Scaling: History and Basic Architecture

The conclusion from our discussion of mathematics is that (1) what we want from a neural net is not necessarily accuracy, but rather controllability; (2) controlling a properly designed dynamical system can be straightforward if the data representation is properly designed; and (3) there is an interplay between information that looks discrete and even symbolic and information that looks sensory and con-

Figure 4. The "network of networks" architecture. It is possible to make an interesting computing device from a large number of connected attractor neural networks. Here each local network is connected to its nearest neighbors. The connections communicate patterns of activity, not average strengths.

tinuous. Perhaps humans can therefore make use of the virtues of both domains to a limited degree.

A few years ago, it seemed to several of us that neural networks as they were then formulated would have serious trouble in the attempt to meet requirements for system flexibility. The “network of networks” model is an attempt to develop a more suitable architecture for mental computation using neural structures (Anderson & Sutton, 1995). The basic idea behind these models arose independently in Sutton’s PhD thesis work (Sutton, 1993; Sutton, Beis, & Trainor, 1989; Sutton & Breiter, 1994; Sutton & Trainor, 1990), as an extension of work by Anderson on radar classification (Anderson, Gately, Penz, & Collins, 1990), and, as just described, arithmetic fact learning.

Suppose that we have many computing elements arranged in a 2-dimensional array. In traditional computational neuroscience or neural network theory, it is usually assumed that these computing elements are neurons, or simple abstractions of neurons. A recurrent subtheme in neurobiology for many years has been the acknowledgment of the presence of groupings of interacting neurons, such as, the “cortical column.”

Assume that there are many local networks, and that the basic computing element in the overall system is not a model neuron but a neural network of some size. Suppose that these local networks communicate with other local networks by means of multiple connections—that is, by activity patterns. Let us investigate the implications of the idea that the communication between the elementary computing units—the local networks—is *vector valued* rather than a scalar representing average activity.

Let us assume that we have a larger network composed of many such interconnected but distinct local nonlinear attractor networks—that is, a “network of networks” (Figure 4). The qualitative behavior of an attractor network is dominated by a number of stable states, the attractors. Two such networks might interact by exchanging vector-valued information about their state. The degree of influence that the state of one network would have on the state of another would be given by a term in an “interaction matrix” coupling the attractor states of the two networks (Figure 5). In place of a scalar activity communicated between networks, we have a vector. In place of a set of connection strengths, we have a set of interaction matrices, describing the coupling between attractors in the two local networks.

A convenient elementary network for this investigation is a simple attractor network called BSB, an autoassociative, energy-minimizing, recurrent network (Anderson, 1993). Since it is a simple recurrent network, the temporal dynamics of the BSB network are easier to analyze than other suitable attractor networks such as Hopfield nets.

The network of networks model operates by amplifying the most useful information in the input to each network, suppressing less useful information, forming localized combinations of features, turning time relationships into spatial relationships, and vice versa, and providing a number of simple ways to direct and control the course of the computation. It is not designed to be an accurate associator.

One reason for the notorious difficulty in applying theory to neuroscience, certainly to cortical neuroscience, is the enormous gap in scale between single neurons and neural structures composed of tens or hundreds of millions of units. The network of networks model tries to make an attempt to bridge this gap. Lack of information about system organization at intermediate scales is one of the problems that limits the usefulness of supercomputers for exploring the connections between behavior and neurons. It would be hard to go from single atoms to the macroscopic properties of matter without acknowledging the existence of molecules.

Informal Analysis

We will discuss one version of this approach that is suitable for very large systems and that is reasonably intuitive in its behavior. Let us start by assuming that the local networks communicate only to adjacent local networks. Let us assume that each local network starts by responding to a few low-level aspects of the data input. Suppose that one module is excited with an activity pattern and starts to influence its neighbors. Let us assume a small degree of correlative learning. Then, a pattern in one network will be coupled to a neighboring network and produce a new state, let us suppose a sum of attractor states, in that local network. Eventually each network will enter one or another of its attractor states and remain stable. The initial activity

Figure 5. The attractor networks influence each other’s states through state interaction matrices. That is, State 7 in Module a gives rise to a certain amount of State 2 in Module b. The matrices M describe the possible interactions. These are not neural interactions but state interactions, giving the influence of one state on another.

pattern will spread out from its source. If another local network is excited by a pattern, that pattern will also start to spread. Suppose that the two local networks are excited at about the same time. Then at some points along the two expansion fronts, local networks will be excited by two distinct strong patterns simultaneously.

These are nonlinear networks. The new pattern will correspond to a nonlinear combination of local features. Let us assume that this new pattern is now learned. Simulations suggest that even simple nonlinear interactions are capable of generating sufficiently different interactions to produce a new attractor, corresponding to the combination of features simultaneously impinging on the network at the interference pattern. The network forms a representation of a feature combination, localized to a group of local networks forming the interference pattern. As time progresses and the individual networks learn, each local network will develop an individualized repertoire of high-level feature combinations. Therefore, the entire network will develop a rich structure of states at different levels of organization. Individualistic responses to high-level feature combinations develop adaptively and are somewhat spatially localized because they arise from collisions between widespread low-level features. Feature combinations are actively formed by the dynamics of the network.

Because it takes time for the system to reach an attractor, feature combinations may appear more slowly than the initial features. Experience with the time course of attractor networks (Anderson, 1992) suggests that the time required for combinations to occur might fall into an interesting gap between single neuron events (a few milliseconds) and psychological reaction times (hundreds of milliseconds). Therefore, such an approach may be suited for information processing involving complex temporal structure in the domain between 50 and 500 msec.

Simulation Strategy

Although the local networks may be large, it is fast and easy to simulate this system. We have modules—the local networks—that show a limited number of stable states. Learning involves formation of feature combinations. Therefore, it should be possible to approximate the behavior of a very large network by approximating the individual networks as multistable elements. Using this approximation for the behavior of a complex system of local networks may be simpler than simulating the detailed dynamics of single modules.

The final state of each individual network can only be in one or another of several point attractors. Each individual network communicates to its neighbors through an interaction matrix, so a state in one network is associatively linked to a state in an adjacent network. The interaction matrix controls system behavior and is more important than the details of the dynamics of the individual networks. The system is buffered to a degree from the properties of the underlying neural elements. The simulations to be described here used only nearest neighbor connectivity (Figure 5) and assumed that the local network was the BSB network. BSB is linear for small signals (early in processing) and nonlinear for large signals (late in processing). Essentially, the linear regime weighs possible final states, and the nonlinear regime chooses the most appropriate. A small BSB simulator program based on the existence of eigenvectors with these properties was written to approximate BSB dynamics. The BSB simulator assumes that the local networks have about 20 point attractors, each with an associated, initially predetermined eigenvalue. Learning influences this model by producing predictable changes in the eigenvalue spectrum.

The presence of 20 attractors implies that there is a 20×20 interaction matrix between modules. Each state in one local network has some degree of control over one or a

Figure 6. Network of networks simulation. The dashes correspond to networks that have not yet reached attractor states. Each network is assumed to have 20 attractor states. Initial inputs of States 1 and 2 were provided roughly at the centers of spreading.

Figure 7. The spreading patterns start to collide. Their nonlinear interaction can, in the proper circumstances, give rise to a new state representing the combination of Feature States 1 and 2. The combination attractor state is given the obvious label, 12.

The stable eigenstates for each network are labeled 1 to 20. The eigenvalues ranged from 1.3 to 1.1, spaced 0.1 apart. The simulations given in Figures 6, 7, and 8 assume

Note that patterns with the same number in different modules need not be identical in detail. The strong associative weight in the coupling matrix gives them functional equivalence. One might speculate that if this approach is

Figure 8. The process goes to completion with the entire network driven into one or another attractor state. There is no state decay in this simulation. More complicated patterns evolving in time occur when state decay is allowed.

correct, discussions of “high-level” interactions (e.g., symbol processing) *might* be replaced by discussion of the details of the interaction matrix.

The simulations shown in Figures 6, 7, and 8 show activity spreading from two modules receiving input patterns, on the left with Pattern 1, and on the right with Pattern 2. This simulation assumes that when Patterns 1 and 2 collide and form an interference front, a new pattern, 12, is created which forms a new stable attractor. In this simulation, the relevant components in the interaction matrix are set by hand. However, other simulations have shown that it is easy to learn them using any of a number of Hebbian learning algorithms. It was not hard to guess weights that worked. The simulations are not sensitive to the small number of system parameters. It is important that this new pattern (“12” in the example) is a strongly nonlinear combination of the component patterns. This can be accomplished by suitable use of the intrinsic nonlinearities of BSB combined with inhibitory patterns from nearby modules acting as decorrelators using lateral pattern inhibition.

The output of one run of the simulation program is given in Figures 6, 7 and 8.

When the simulation has progressed, as in Figure 7, the two spreading pattern fronts interact at the boundary. It is assumed that learning has taken place in the past and an interference (Pattern 12) boundary has been learned. (It was a little easier to work with a double boundary row in early simulations; hence it is used here.) Eventually, a complete boundary between regions forms (Figure 8).

Note that dynamics takes a local initial input, represents it as activity of large parts of the entire array, and forms a new class of spatially localized states corresponding to the combination of the initial states. Any of these processes would make the job of a higher level analyzer easier. This input pattern could be recognized only by looking at the feature combination states. A number of arrangements of input will form the same interference front. For example, if the excited module on the right is translated one unit to the right and the excited module on the left is translated one module to the left, the location of the interference front is unchanged, though it will take longer to excite. The topography of the module connections becomes an important information processing tool.

Controllability

Because of the tendency of the system to spatially localize feature combinations, if we enhance or suppress these regions—for example, by simple inhibition or excitation, or by control of module eigenvalues—it is possible to direct a network computation down different pathways. It seems to us that such a system will do its computation with control by associative branching rather than with control by logic. This computational style is well suited to an associative system—as has been known since Aristotle—but it leads to an unfamiliar form of computation.

Let us note that this class of models has two attractions for VLSI implementation. First, computation is performed by identical modules, and processing power is a function of the size of the number of modules; second, connectiv-

ity scales linearly with network size. The major problems with it are that the network computation has a serial component (activity spread), the time constants are analog in nature and, as always with new computer design, the software must be fitted to the architecture.

Psychophysical Tests

In the 1960s, Harry Blum (1973) proposed the “grass-fire” model of perception, which made visual perception an active process, that is, similar in operation to this model. The usefulness of interference patterns for mental operation was also mentioned by Lashley (1950). Recently, a striking paper by Kovacs and Julesz (1994) provided psychophysical evidence for effects somewhat like those predicted by the grassfire model, as the authors pointed out. The presence of a circular contour produces demonstrable effects on thresholds at the center of the circle, even when the circle is far too large to be seen by single receptive fields. Nearby points inside the circle seem not to be affected. Numerous other “traveling wave” effects are seen in early perception (e.g., various masking phenomena).

Conclusions

It is a fact that most of those interested in understanding the brain do not make much use of the highest performance machines. This is partly because of cultural barriers to supercomputer use, but there are some significant scientific reasons.

Many physical laws, such as electromagnetism or gravity, apply across many orders of magnitude in scale. For physics, chemistry, and engineering, the elementary rules are very good. If calculations are done properly, a plane will fly and the wings will not fall off. Therefore, a computation starting with very low-level assumptions can, after billions of simple operations, give trustworthy results.

In cognitive science and neuroscience, trustable scaling relationships do not exist. We have no first principles that can be extrapolated to such a degree. However, our problem is simplified to a considerable degree because the nervous system exists to generate behavior and behavior must show coordination, predictability, and controllability. The result of a nervous system computation may be understood far more simply than the detailed means used to attain it.

Structures such as the nervous system, which are designed to perform particular functions, can exhibit *decoupling*, wherein the computation can be remarkably independent of the details. Perhaps the best example of decoupling is the actual computer itself. The physical mechanism to produce the result of a computation can be a result of operations performed by VLSI, transistors, vacuum tubes, relays, or pencil and paper. There are obvious practical differences between the mechanisms, but if the algorithm is the same, the results will be the same. Perhaps the same is true of the neurocomputer. The network of networks—where the significant computational primitives are particular large-scale patterns, attractor states, and their interactions, and not single neurons—may provide a similar demonstration.

REFERENCES

- ANDERSON, J. A. (1992). Why, having so many neurons, do we have so few thoughts? In W. E. Hockley & S. Lewandowsky (Eds.), *Relating theory to data: Essays on human memory* (pp. 477-507). Hillsdale, NJ: Erlbaum.
- ANDERSON, J. A. (1993). The BSB model: A simple non-linear auto-associative network. In M. Hassoun (Ed.), *Associative neural memories: Theory and implementation* (pp. 77-102). Oxford: Oxford University Press.
- ANDERSON, J. A. (1995). *Introduction to neural networks*. Cambridge, MA: MIT Press.
- ANDERSON, J. A. (in press). Seven times seven is about fifty. In S. Sternberg (Ed.), *An invitation to cognitive science* (Vol. 4). Cambridge, MA: MIT Press.
- ANDERSON, J. A., GATELY, M. T., PENZ, P. A., & COLLINS, D. R. (1990). Radar signal categorization using a neural network. *IEEE Proceedings*, **78**, 1646-1657.
- ANDERSON, J. A., SPOEHR, K. T., & BENNETT, D. J. (1994). A study in numerical perversity: Teaching arithmetic to a neural network. In D. S. Levine & M. Aparicio IV (Eds.), *Neural networks for knowledge representation and inference* (pp. 311-335). Hillsdale, NJ: Erlbaum.
- ANDERSON, J. A., & SUTTON, J. P. (1995). The network of networks model. In *Proceedings of the World Congress of Neural Networks: Washington, DC* (Vol. 1, pp. 145-152). Hillsdale, NJ: Erlbaum.
- BLUM, H. (1973). Biological shape and visual science (Part I). *Journal of Theoretical Biology*, **38**, 205-287.
- DALLAWAY, R. (1994). *Dynamics of arithmetic: A connectionist view of arithmetic skills* (Cognitive Science Research Paper 306). Brighton, U.K.: University of Sussex.
- EINSTEIN, A. (1949). Autobiographical notes. In P. A. Schlipp (Ed.), *Albert Einstein: Philosopher scientist* (pp. 1-95). New York: Tudor.
- HADAMARD, J. (1949). *The psychology of invention in the mathematical field*. New York: Dover.
- KOVACS, I., & JULESZ, B. (1994). Perceptual sensitivity maps within globally defined visual shapes. *Nature*, **370**, 644-646.
- LASHLEY, K. S. (1950). In search of the engram. In *Society of Experimental Biology Symposium No. 4: Psychological mechanisms in animal behavior* (pp. 454-480). Cambridge: Cambridge University Press.
- LINK, S. (1990). Modeling imageless thought: The relative judgment theory of numerical comparisons. *Journal of Mathematical Psychology*, **34**, 2-41.
- MCCLOSKEY, M., & LINDEMANN, A. M. (1992). MATHNET: Preliminary results from a distributed model of arithmetic fact retrieval. In J. I. D. Campbell (Ed.), *The nature and origin of mathematical skills* (pp. 365-410). Amsterdam: North-Holland.
- PINKER, S., & PRINCE, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. In S. Pinker & J. Mehler (Eds.), *Connections and symbols* (pp. 73-193). Cambridge, MA: MIT Press.
- SUTTON, J. P. (1993). Learning by selection in multi-layered networks. In *Proceedings of the 1993 WCNN Conference* (Vol. 2, pp. 536-539). Hillsdale, NJ: Erlbaum.
- SUTTON, J. P., BEIS, J. S., & TRAINOR, L. E. H. (1989). Hierarchical model of memory and memory loss. *Journal of Physics A: Mathematical & General*, **21**, 4443-4454.
- SUTTON, J. P., & BREITER, H. C. (1994). Neural scale invariance: An integrative model with implications for neuropathology. In *Proceedings of the 1994 World Congress on Neural Networks* (Vol. 4, pp. 667-672). Hillsdale, NJ: Erlbaum.
- SUTTON, J. P., & TRAINOR, L. E. H. (1990). Real and artificial neural hierarchies. In C. N. Manikopoulos (Ed.), *Cybernetics and systems* (Vol. 1, pp. 229-236). Newark: New Jersey Institute of Technology.
- TANAKA, K. (1996). Inferotemporal cortex and object vision. In W. M. Cowan, E. M. Shooter, C. F. Stevens, & R. F. Thompson (Eds.), *Annual review of neuroscience* (Vol. 19, pp. 109-140). Palo Alto, CA: Annual Reviews.