Computer, cognitive, and neurological researchers who seek to mimic the mind are returning to wiring that imitates the brain's as an alternative to expert systems, which try to find and follow the rules by which people appear to reason.

Give a computer the right instructions and, some say, it will act like a brain. Those "right instructions" are artificial-intelligence programs that run according to the rules and procedures that minds follow. But in fact, what AI programs do well—reasoning logically—people do badly. On the other hand, what human brains do well—generalizing from examples, making associations where none existed—AI cannot do at all.

A second way to offer a computer the attributes of a brain is to build it like one in layers of interconnected electronic surrogate neurons whose organization mimics the brain's. These computers, called neural networks, seem more apt than AI programs at functioning like the brain does. They make generalizations and original associations, and they have trouble with the multiplication table.

The difference between artificial intelligence and neural networks is, at bottom, the familiar debate over how best to describe the mind. Investigators of artificial intelligence and neural networks find themselves on opposite sides of the argument.

The AI side of the debate argues that the mind is best described by the rules it follows, especially the rules of logical reasoning. To reproduce the mind, a computer is programmed to follow the rules of logical reasoning. According to Tomaso Poggio and Anya Hurlbert, AI researchers at the Massachusetts Institute of Technology, rules include "logic, mathematical proofs, legal debates, and the systematic elimination of all possible bugs in computer code."

The neural network side holds that because the mind is the brain's behavior, any description of the mind must be grounded in the mechanics of the brain. To reproduce the mind, the structure and wiring of the brain are imitated. The neural networks that emerge from this approach seem to replicate abilities that are uniquely human.

Humans are astonishingly good at dealing with an unorganized world. The world presents sights, sounds, smells, textures—all in disarray. From these, humans learn to associate, to recognize people and things as related to each other and as familiar. "We wade through the quicksand of multiple constraints," Poggio and Hurlbert have written, "talking, humming, driving cars, reaching for coffee cups, recognizing faces in the crowd." Human associative learning, moreover, seems effortless: Something seen a few times is somehow, mysteriously, recognized. "The things that we're computationally most powerful at," says John Hopfield, a physicist at the California Institute of Technology, "are the things we can't explain at all."

On the other hand, humans are pretty bad at math and logic. "Logical thought," says physicist Edward Teller, "is so rare in humans that it is almost a perversity." For example, cognitive scientist James Anderson at Brown University tried out that excellent example of logical, linear thinking, the multiplication table, on a neural network. His finding: "It did fine, if an acceptable answer is '7 × 5 is 40ish.' Otherwise, it did a terrible job."

Neural networks are designed to replicate associative learning. Larry Jackel, a physicist at AT&T Bell Laboratories, presents a neural network with handwritten numerals and it can pick out the number 9. When Demetri Psaltis, an electrical engineer at the California Institute of Technology, presents examples of faces to a network, it recognizes his. Terry Sejnowski, a biophysicist at Johns Hopkins University, presents examples of words to a network, and it learns to pronounce aloud. Christof Koch, a biophysicist at Caltech, designs and builds simple networks that enable a robot to navigate through a room or, he says, across a Martian landscape. Neural networks now play backgammon, recognize faces from parts of photos, classify animals, learn irregular verbs, recognize airplanes, and even evaluate credit for loans.

To understand the brain

Neural networks began with attempts to understand the mind by understanding the brain. In 1943, Warren McCulloch, a psychiatrist at the Universities of Illinois and Chicago, proposed a theory of the mind in collaboration with Walter Pitts ("a brilliant and unstable undergraduate," according to Anderson, "who

Sejnowski. His neural nets learned to pronounce English, corrected their own errors.

never graduated"). Their joint article in *The Bulletin of Mathematical Biophysics* bore the title "A Logical Calculus of the Ideas Immanent in Nervous Activity." McCulloch and Pitts argued that a cure for "diseased mentalities" had to begin with a rigorous, scientific description of the mind—that is, the mind defined as the workings of the brain's most basic units, the neurons. Neurons could be seen as logic devices, they wrote: "Neural events and the relations among them can be treated by means of propositional logic."

By the phrase "neural events and relations," McCulloch and Pitts meant the rules under which neurons operate when they communicate with each other. The communication is done electrochemically. An electric current triggered in a neuron travels through the cell's long axon to a gap, the synapse. Into the synapse, then, the neuron deposits chemical neurotransmitters; these cross to the next neurons and provoke other electrochemical events. The brain has about 10^{11} neurons, each one communicating with some 10^4 other neurons.

The process, however, is not a simple domino-like line of electrochemical triggers and responses. It follows certain rules. Neurons fire only if the signal is strong enough; every neuron has a threshold, below which it is silent and above which it fires. Furthermore, neurons fire only if the current is positive. A negative current will not only inhibit the neuron from firing but also prevent any other stimulus from exciting it. Each neuron sums up all the exciting and inhibiting signals from its thousands of connections. Then, depending on its threshold, it decides whether to fire. The neuron will either fire or not fire, will be either on or off.

McCulloch and Pitts stated that this two-state neuron could be treated as a logic device. For instance, they declared, it can compute the logical *and* operation. If it is assumed that a neuron has a threshold of 2 and has two connections, A and B; then if A and B are off, the neuron is off; if either A or B is on, the neuron remains off. Only if both A and B are on is the neuron on. If the threshold is changed to 1, the neuron instead computes the *inclusive or* operation: if either A or B is on, the neuron is on. If both A and B are on, the neuron is also on. Neurons that behave so are now called McCulloch-Pitts neurons.

Two years later, in 1945, John von Neumann—the mathematician who (among other things) designed the first digital computers—wrote, "Following W. Pitts and W. S. McCulloch, . . . it can easily be seen that these simplified neuron functions can be imitated by telegraph relays or by vacuum tubes." In this view, devices like relays, or tubes, or transistors are substituted for neurons; each device is given a certain threshold, and then the devices are connected by wires. The result is the skeleton of a simple computer: a brain machine with electronic analogues for neurons, thresholds, and synapses.

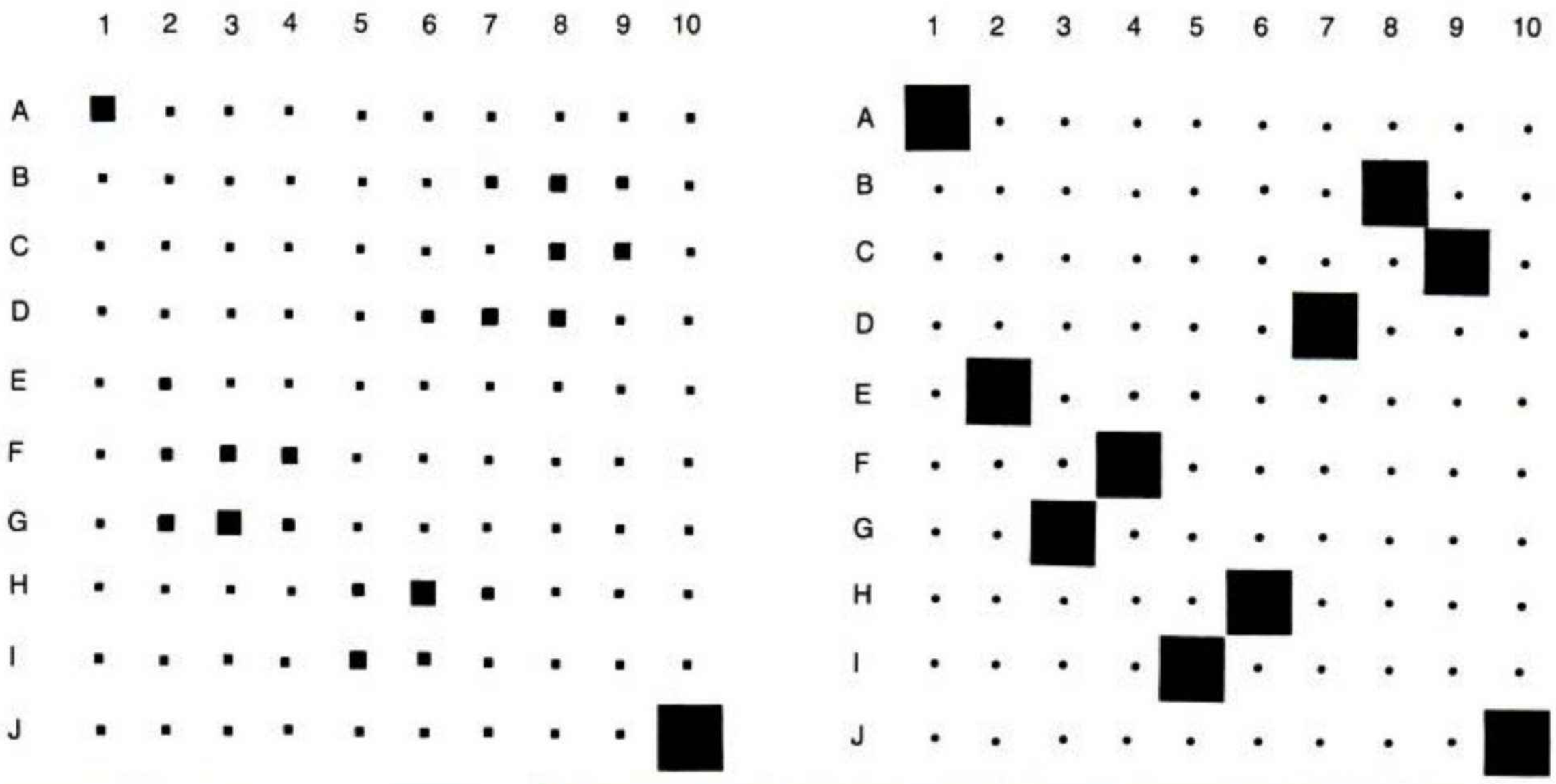
Perceptrons

In 1958, Frank Rosenblatt, a psychologist at Cornell Aeronautics Laboratory, reduced von Neumann's idea to hardware. Rosenblatt built a machine that made its own associations—specifically, a machine that learned to classify shapes. He called it a perceptron.

The perceptron imitated the brain's organization, though not its complexity. In this machine, neurons are arranged in a rough three-level hierarchy according to function: sensory neurons take in information, motor neurons control the body's response to the information, and interneurons (by far the most numerous) communicate between the other two types. Rosenblatt's Mark I *Perceptron* had only two layers. In one layer were 512 relaylike associator units, triggered by light sensors. In the second layer were eight response units, each correspond-

---

Finkbeiner wrote "Demographics or Market Forces," on problems related to the supply of scientists and engineers, in Mosaic *Volume 18 Number 1 1987.*



A short enough path. A neural network in the process of solving a traveling salesman problem (left) and the final solution (right). The size of each square (or neuron) represents the activity of that neuron. City J has been placed arbitrarily in location 10 in the intermediate solution; thus, the square at location J-10 is of maximum size. During the decision-making process, many neurons may be partially on. In the final form, however, each of the outputs is either on or off. The final state represents the path AEGFIHDBCJ.

Hopfield. A physicist who added credibility.

ing to one of eight classifications. All associator units were connected to all response units.

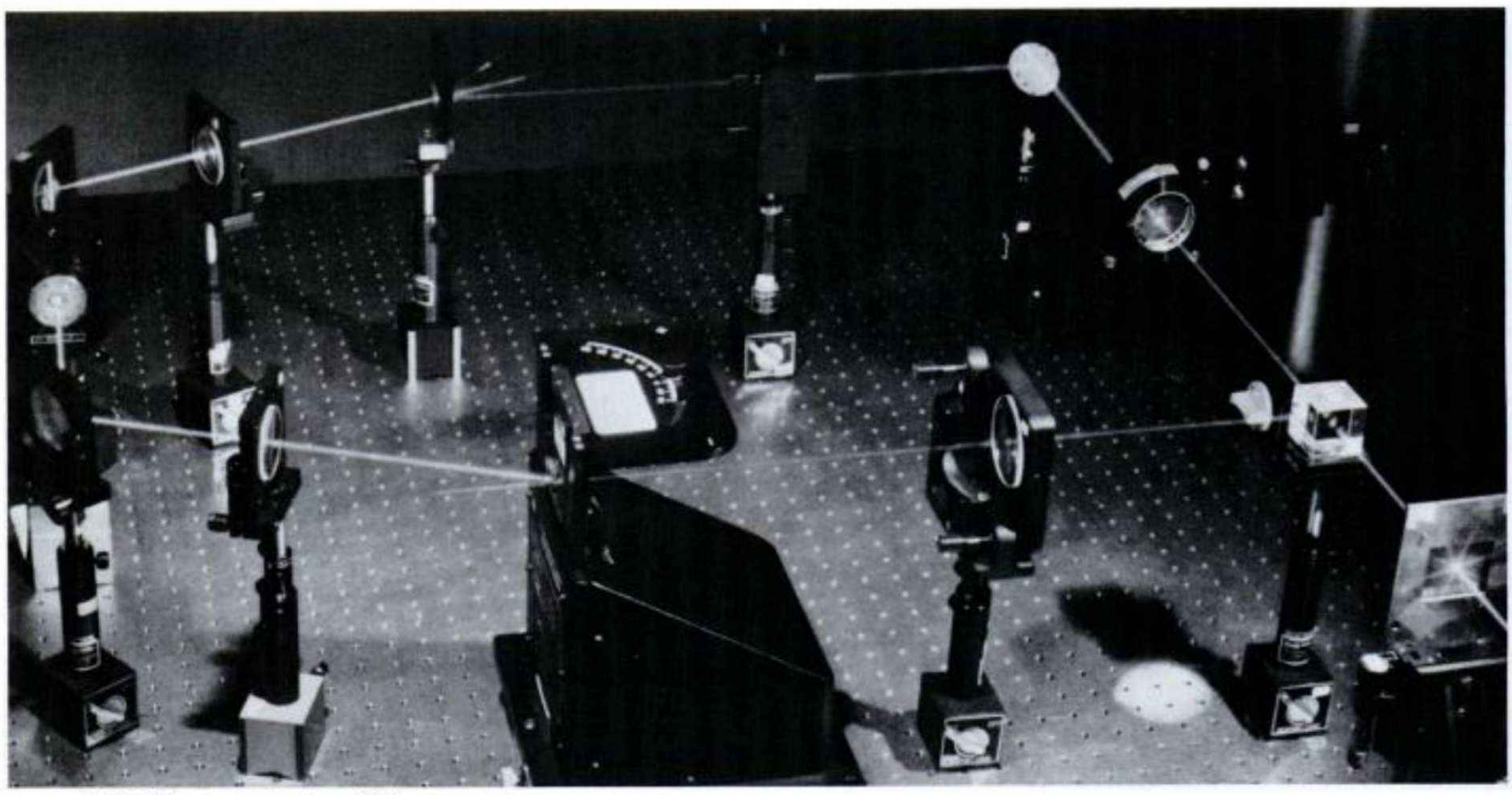
Simple devices with thresholds and wires, arranged in layers, however, do not learn. Neurons "learn," or are modified, when somehow conversations between certain neurons become more important, when signals pass between them more readily. Neurologists say the synapse modifies, or synaptic strength changes. Exactly how this happens no one knows: "At the moment," says Caltech's Christof Koch, "the question of what makes the synapse stronger is very open, very controversial." (See "Learning at the Sub-Neural Level" by Robert Kanigel, *Mosaic* Volume 18 Number 3 Fall 1987.)

A machine that can learn, then, needs a modifiable synapse. The electronic analogue to a synapse is a device, attached to the wires between transistors, called a resistor. Resistors specify the amount of current leaving one transistor that will reach the next one—much the way a volume control turns a radio up or down. A resistor set to a low value makes for a strong connection.

To decide how to change the strength of the connections between units in the perceptron, Rosenblatt used a variation on an old rule. In 1949, Donald Hebb, psychologist and chancellor at McGill University, had stated in *Organization of Behavior* what is now called Hebb's rule: When two connected nerve cells fire simultaneously, the connection between them becomes stronger. The idea, explains Koch, is that "neurons detect coincidences, correlations. Probably if event 1 and event 2 happen at the same time very often, they're related." For example: Event 1 triggers cell A; event 2 trig-

Mechanical analogues

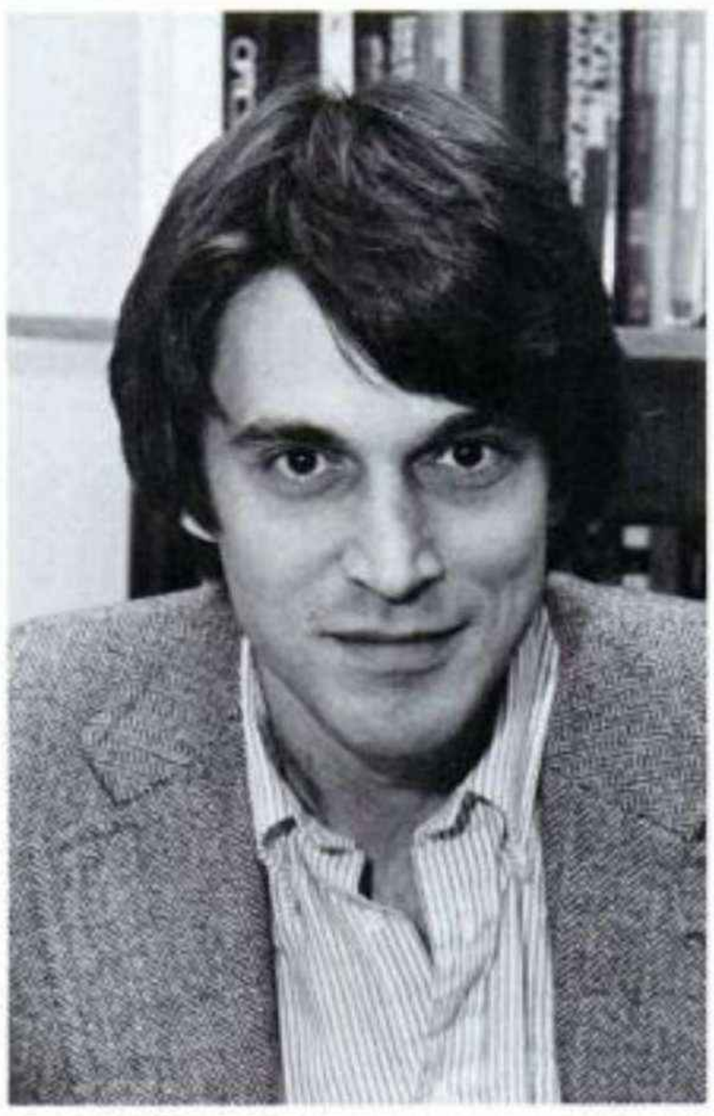
Finding mechanical analogues for the brain has a long tradition. In 1664, René Descartes compared the brain to hydraulic systems driving the waterworks at Versailles. A century later, Denis Diderot described the brain as a network of little threads that resonate to the touch like the quill-plucked strings of a harpsichord; the threads of a well-educated person's brain resonate in particular harmony. The nineteenth century discovered electrical analogues: Herbert Spencer, the Victorian philosopher, compared the nervous system to telegraph relays; and philosopher of science Karl Pearson said the brain was like the central exchange of a telephone system. The twentieth century has two analogues, both actually capable of behaving enough like the brain to convince many humans: One is the conventional digital computer, the other the neural network. •



gers cell B. When events 1 and 2 happen at the same time, the connection between A and B gets stronger. The next time A is on, B is more likely to go on too. Furthermore, if cells Y and Z are connected to A and B, then by extension, the connections between cells Y and Z should be strengthened as well. This is one possible basis—in theory anyway—of associative learning.

Perceptrons, then, learned because they were constructed of devices that behaved like McCulloch-Pitts neurons and obeyed Hebb's rule. For example, an investigator can "show" a perceptron a square. If it responds by indicating a circle, the investigator can lower the values of the resistors, strengthening the signal between *circle* and the units representing *square*. If the resistors between the two concepts are lowered, the next time the perceptron sees a square, it is less likely to respond with a circle. If it responds with a square, the investigator will leave the values alone: perceptrons punish failure and ignore success. The next square a perceptron encounters, it will be able to recognize.

At first, Rosenblatt's perceptron and his book outlining other perceptrons were a success. Following psychologists McCulloch, Hebb, and Rosenblatt, who worked with ideas from neurology, engineers then tried similar problems on similar machines. Bernard Widrow, an electrical engineer at Stanford University, had a machine he called *Adaline (for*



*ada*ptive *li*near *ne*urons), which could give "reliable, reasonable responses to new patterns; that is, it could generalize."

Widrow trained *Adaline* to identify simple patterns that were first held vertically, then rotated by 90 degrees, 180 degrees, and 270 degrees. After training, *Adaline* could recognize other patterns similarly rotated. After further training, it recognized patterns translated from left to right, from up to down, or from large to small. Electrical engineers at the Massachusetts Institute of

Training laser beams. Psaltis (left) and his optical memory system, simulating a neural network of 10,000 neurons. The interconnecting holograms can store images; when presented with a partial image, the system can retrieve the complete version.

Technology, building on research with cells in frogs' eyes that responded to moving black spots, proposed sending an artificial frog, equipped with a perceptron, to Mars to detect Martian flies. At the time, as Anderson notes in a forthcoming book, "it seemed as if [perceptrons] could do anything. A hundred algorithms bloomed, a hundred schools of learning machines contended."

Limits

"Unfortunately," says Hopfield, "though perceptrons were an interesting thing to try, they ran into a wall." The wall had several components. One was that research into perceptrons was not producing practical applications. Another was that the alternative approach to making machines act like brains—artificial intelligence—was having remarkable success. A third was a book written in 1969 by Marvin Minsky and Seymour Papert, who were the co-directors of the Massachusetts Institute of Technology's Artificial Intelligence Laboratory and founders of artificial intelligence. Their book, *Perceptrons*, outlined the reasons perceptrons could go no farther than they had.

One problem with perceptrons was that they were structured less like the brain than like the spinal cord. Whereas the brain has three general layers of neurons—input, connections, and output—the two-layer spinal cord handles only reflexes, in which input leads directly to output. Two-layered machines are limited that way in the kinds of learning they can achieve.

Another limitation was that perceptrons were able to handle only certain kinds of logic. To go beyond those limits, perceptrons would have to be equipped either with more layers or with more logic operations—specifically the *exclusive or* operation, which requires it to turn on if either A or B is on but not if both are on.

Moreover, as Minsky and Papert explained in a 1987 update of *Perceptrons*, "when failure occurred, neither prolonging the training experiments nor building larger machines helped. All perceptrons would fail to learn to do those things. . . ." The result of all this was what Anderson called the dark ages, "where suddenly research on neural networks was unloved, unwanted—and most importantly—unfunded."

For the next 15 years, AI flourished. The first AI programs, written in the mid-1950s by Allen Newell and Herbert Simon at Carnegie Mellon University and Clifford Shaw at the Rand Corporation, were mathematical proofs that were neater and shorter than the standard proofs. By the late 1960s, Newell and Simon had written one of the first programs for an expert system. Such programs include precisely the hundreds of steps, in sequence, that human experts seemed to follow in solving problems. Thanks to sophisticated programs, AI expert systems can now play chess, read texts, analyze the structure of chemical compounds, diagnose diseases, and prescribe medication. (See "Before They Can Speak, They Must Know" by William J. Cromie and Lee Edson, and other articles in *Mosaic* Volume 15 Number 1 1984, a special issue on advanced computer research.)

But AI, however successful, has a blind spot: Real-world problems, like recognizing a tree or translating a foreign language, entail endless variants and possibilities. Hopfield points out that AI "is strictly logic-based, and so, with real-world problems and better data bases, it tends to fall apart." For example, he says, "since most words can be translated in a number of different ways, by the time the expert system has a ten-word sentence, it's in trouble. You've got a combinatorial explosion."

Combinatorial explosions, although they disable AI systems, make the reasoning process easier for humans. "English has a lot of ambiguous words," says Anderson, "*Bat*, *ball*, and *diamond* have three or four different meanings apiece. The correct meaning depends on the context. So if I group bat, ball, and diamond together, you know what I'm talking about. If I tell you *game*, you won't have any doubt.

"People are terrific at this. But suppose we're doing an AI search. To disambiguate the meanings, [the expert system] has to check all possible associations and look for the one that agrees with all of them. With AI, the more information you have, the longer the search takes. People work just the other way. If I give you more associations, your reaction time speeds up." In short, says Anderson, "AI expert systems work like human novices. Human experts not only follow rules, they have connections between the rules—intuition, hunches."

The neural approach, which had been plugging along at a low but steady level, revived in the early 1980s, this time under the name connectionism. One reason for this revival was that the new neural, or connectionist, networks seemed to promise things that neither perceptrons nor AI could. "If you give a neural net only the word *ball*, it flounders around, [then] chooses a possible association in about 100 iterations," says Anderson. "If you give it *ball* and *bat*, it disambiguates and gets *baseball* in 30 iterations. Put in *bat*, *ball*, and *diamond*, [and] it gets *baseball* in 14." Neural nets seem to have some of the same talents as human beings.

Networks revisited

The new neural networks have had several incarnations: as especially wired computers linked to conventional digital computers, as custom-built VLSI chips, and as simulations on a more conventional computer.

Neural networks use the perceptron's basic principles but make two changes.

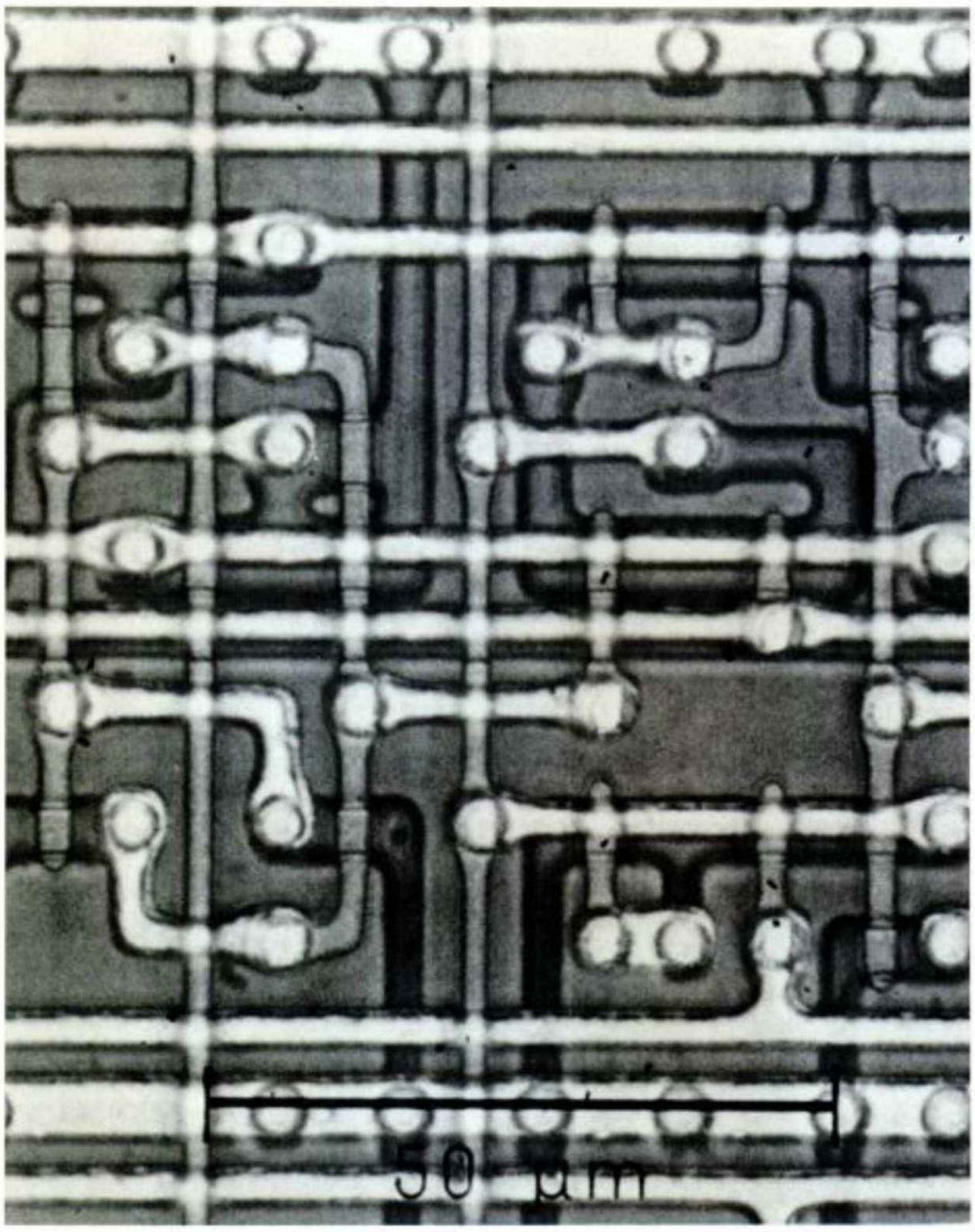
Nineteenth-century prescience

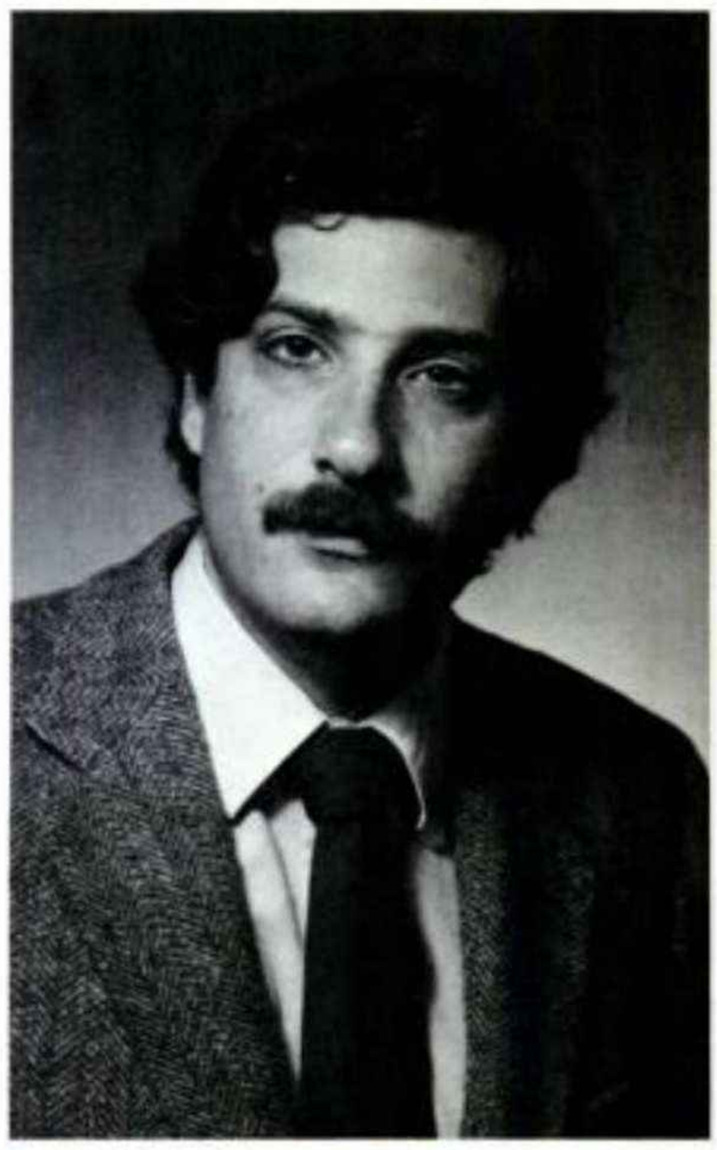
Connectionism could conceivably be said to date from the ideas of William James. In his treatise called *Psychology: The Briefer Course* (1892), James described the process as follows:

> The manner in which trains of imagery and consideration follow each other through our thinking, the restless flight of one idea before the next, the transitions our minds make between things wide as the poles asunder, transitions which at first sight startle us by their abruptness, but which, when scrutinized closely, often reveal intermediating links of perfect naturalness and propriety—all this magical, imponderable streaming has from time immemorial excited the admiration of all whose attention happened to be caught by its omnipresent mystery. [Therefore we should ascertain] between the thoughts which thus appear to sprout one out of the other, principles of connection. . . .

James also anticipated the rule on reinforced synapses that psychologist Donald Hebb would develop 50 years later. In *The Briefer Course* James postulated, "Let us then assume as the basis of all our subsequent reasoning this law: When two elementary brain-processes have been active together or in immediate succession, one of them, on recurring, tends to propagate its excitement into the other."

At the same time, James described a mechanism for association that is uncannily like neural networks. Thought A, of a dinner party, is composed of details *a*, *b*, *c*, *d*, and *e*. Thought B, of walking home afterward, is similarly composed of details *l*, *m*, *n*, *o*, and *p*. All details connect all other details, "discharging into each other," according to James. As a result, then, "the thought of A must awaken that of B, because *a*, *b*, *c*, *d*, and *e* will each and all discharge into *l* . . ."; and *l* "vibrates in unison" with *m*, *n*, *o*, and *p*. ●





Neuron chip. Jackel and (left) a small section of chip used to recognize handwritten numbers.

One is in the networks' architecture. Terry Sejnowski of Johns Hopkins and Geoffrey Hinton, a computer scientist now at the University of Toronto, added to the two-layered perceptron a third layer they call the hidden layer. The hidden layer corresponds to the brain's interneurons, the neurons in the middle that handle neither input nor output but only communicate. In three-layered networks, the input units all talk to the hidden units—which in turn talk to the output units. (See "Inside the Hidden Layers" accompanying this article.)

Another change is in the learning rules, or algorithms. Sejnowski and Hinton developed extensions of Hebb's rule that allow the networks to correct their own errors. Others independently developed a similarly self-correcting algorithm, now called back-propagation. Among them were David Rumelhart and Ronald Williams at the University of California at San Diego, working with Geoffrey Hinton; David Parker at Stanford; and Paul Werbos (now at the Energy Information Agency of the Department of Energy in Washington, D.C.) in his Ph.D. thesis at Harvard. What the algorithms do is allow the networks to compare their output to a standard, note the extent of the rightness or wrongness, and adjust the connection strengths in the hidden layer accordingly. The most powerful of the new algorithms, Anderson says, is back-propagation.

Back-propagation is like hide-and-seek with clues—like when someone tells the seeker he is getting either warmer, or cooler; now cold, now hot. For every occurrence of the signal "warmer," the connections along a particular path become stronger; for every instance of "cooler," the connections become weaker. Because the algorithm tells the network nothing more than that, the new networks are self-teaching. They also contain many paths (some more direct than others) to the same answer.

So to speak

For example: Sejnowski, with Charles Rosenberg at Princeton University, ran a neural network simulation they called *NetTalk* that learned to pronounce English. That is, *NetTalk*, when presented with the letter *n* (embedded in a word) learned to come up with the sound *nnn*. The letter *n* was represented by the start-up of a certain input unit. When the N unit turned on in the input layer, a corresponding set of units then lit up in the hidden layer.

The first time through, the N unit activated a random pattern of units in the hidden layer. That pattern happened to trigger the unit in the output layer that corresponded to the phoneme *ah*. (Each output unit represented one of the 55 possible phonemes in the English language.) A teacher—a program with the correct letter-to-phoneme relationships—sent back a message to the hidden layer: Wrong. So the next time the hidden layer encountered input from the N unit, it knew it had to avoid the output unit for *ah*.

In less anthropomorphic terms, the learning algorithm reduced the strengths

of the connections between the pattern in the hidden layer that corresponded to N and the output phoneme *ah*. In effect, the connections between N and *ah* weakened. After several more iterations, the output layer tried *mmm*. This time the teacher sent back another message: Slightly Right. Now the connections between the pattern in the hidden layer corresponding to N and the phoneme *mmm* were strengthened, but only a little. When the hidden layer finally got around to triggering *nnn*, those connections were simply left alone: Don't argue with success! The next time the network saw N, it had several ways (some better than others) to get to *nnn*.

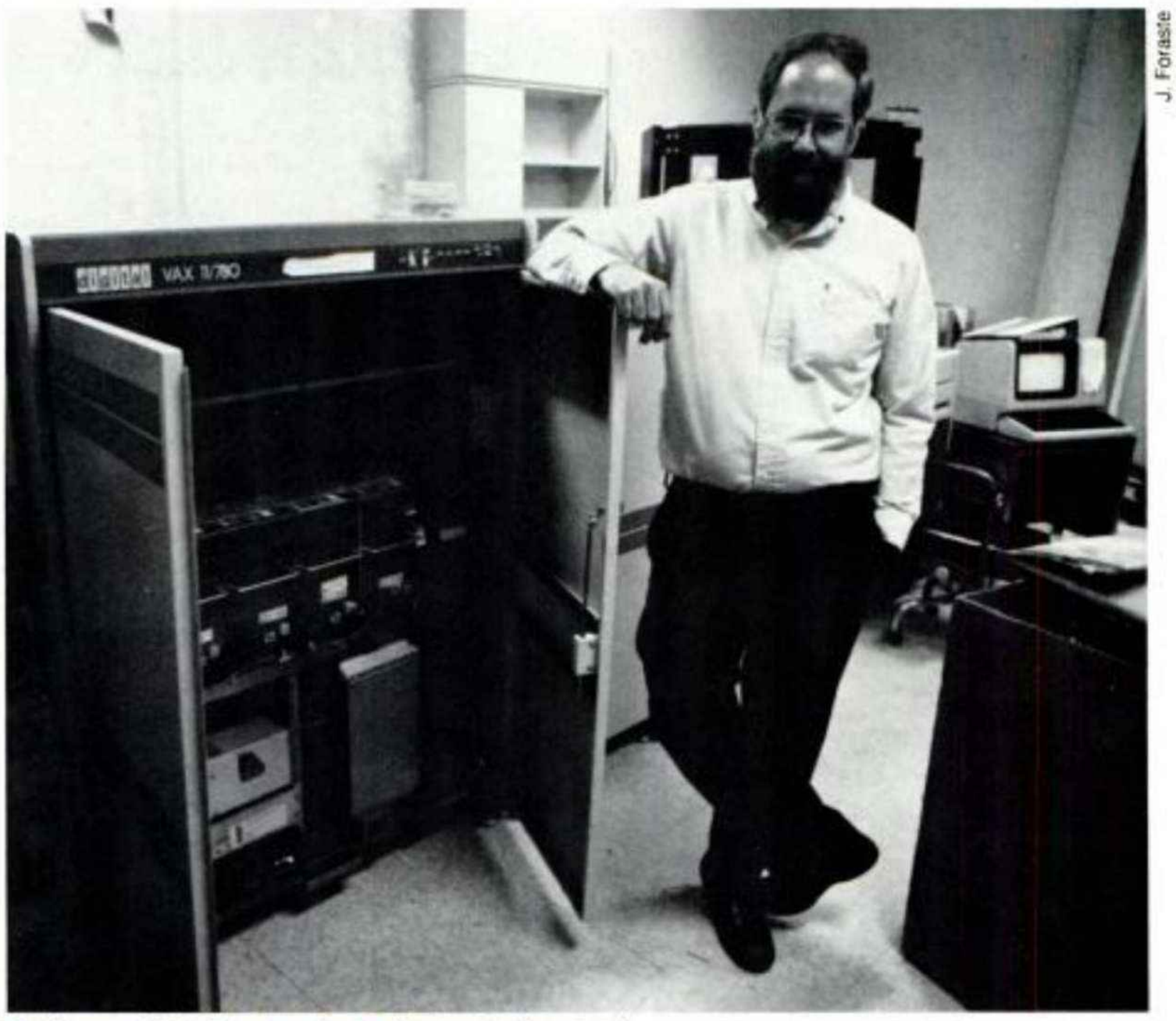
The recipe is fairly simple. For example, to get a network to recognize a cat, assign each input unit some feature of animals. Assign each output unit an animal. Then tweak the connections between the units. Large size, inhibit strongly; four legs, excite weakly; whisker, excite strongly; color, excite weakly.

Paths to "Grandmother"

This recipe, simple though it is, has interesting implications. One is that because neural networks make decisions by means of a multiplicity of connections (some stronger, some weaker), concepts or memories are spread throughout the network. Researchers call this phenomenon distributed processing. The image or concept of "grandmother," for example, does not reside in the triggering of a certain processing unit. Nobody, researchers say, has yet found a "grandmother neuron" in the brain.

Just as "cat" could be reached via component concepts such as "whiskers" and even "large size," "Grandmother" can be reached by paths that might include "Sunday-night popcorn," "gray gloves," and "leaky ball-point pens." If the path via gray gloves were somehow blocked, ways to "Grandmother"—though fewer—would still exist. After Bernard Widrow had successfully demonstrated his network *Adaline* in 1960, he found out that 25 percent of the circuitry had been defective. Researchers call this ability to operate despite impairments "graceful degradation": network performance, instead of toppling over dead, degrades gracefully, instead.

Another implication of the way neural nets operate is that some networks can suggest associations that they have not been taught to make. Says Hopfield,



J. Foraste

Anderson. Tried hard to fool a hard-to-fool network.

"You put memories into [some networks], and they develop other memories. A network that holds a memory of John-red-hat, and John-green-ball, and Bill-red-ball, will also be able to conceptualize John-red-ball. If you give it *John* and *ball*, it might give you *red*."

A third implication of the way neural networks operate is that they reach answers through a series of connections that are strong, but not necessarily the strongest possible. As a result, the answer will be pretty good but not the best.

Hopfield's network solves the classic traveling-salesman problem: 20 cities—each one visited once only—by the shortest possible route. Out of the 10^{15} possible paths, Hopfield's network finds not the shortest path, but one that is short enough.

One of Anderson's neural-network simulations has learned, if given the concepts of a general pathogen and a disease, to specify the exact pathogen involved and to prescribe medication. This network is hard to fool.

Anderson tried, for instance, to stump the network with a trick question. He linked a Gram-positive bacillus with meningitis—a combination the data base

Terminology

Terminology in the field of neural networks has not settled into a pattern of consistent usage yet. The networks are variously called connectionist machines, adaptive resonance machines, neural networks, neural computers, and parallel distributed processors. The electronic analogues to neurons, the transistors, are called units, processors, or even "neurons" (with quotation marks). The electronic analogues to synapses—the resistors (or in simulations, the values of stored numbers)—are referred to as connection strengths or weights, or synaptic weights. The tendency to use physiological terms to refer to electronics, however, annoys some researchers.

Certain machines decide by consensus on solutions that are not perfect, but are good enough for current purposes. These machines are said to operate on a principle denoted as any of the following: optimization, constraint satisfaction, global energy minima, goodness, or harmony. ●

had never been shown. Then he asked it for a prescription and a pathogen.

After about 30 iterations, the machine came up with what Anderson calls a consensus prescription: penicillin. After 100 or so iterations, the network identified the organism as *Corynebacterium*, a Gram-positive bacillus that is in fact not likely to cause meningitis. "The machine guessed," Anderson declares, "and penicillin is a pretty reasonable guess. The other guess is not so good, and also took more time—which usually means the system is unsure of its answer."

More experiments

Neural networks, or simulations of them, or neural chips, all differ in their architectures and their learning algorithms. In some networks, like *NetTalk*, the connections are changed by a mechanism in the network's program. Other networks have their connection strengths preset by the researchers. "Presetting the weights can be done with clearly specified problems," says John Hopkins's Terry Sejnowski, "like the rules for the traveling-salesman problem."

Neural networks also differ in the types of problems they are designed to handle. Physicist Alan Lapedes and computer scientist Robert Farber, both at Los Alamos National Laboratory, used a neural-network simulation to determine whether a given sequence of base pairs on a strand of DNA could be responsible for producing a protein.

Conventional methods can reliably predict protein production only when a strand is long. However, the strands that produce proteins are short. Lapedes and Farber trained a neural net by showing it short strands that do make proteins and short strands that do not. Not only could the neural net predict whether a new strand would produce proteins but it also found a mistake in *GenBank*, a library of protein-producing DNA strands at Los Alamos National Laboratory.

Electrical engineer Demetri Psaltis has built what he calls an optical neural computer. The advantage of optical neural networks, says Psaltis, is that a one-centimeter crystal cube can store a million connections—storage that requires some 100 chips to achieve. Instead of using transistors triggered by electric current,

Inside the hidden layers

Hidden layers, processing units between input and output layers and analogues to the brain's interneurons, are something of a black box. Researchers are sure of what ought to happen inside them, but vague on what actually does happen. In general, says Caltech's John Hopfield, researchers think that "hidden layers detect broad classes of features of things that are present in the input data. Hidden layers represent structure in the stimulus."

After the machines have learned and generalized, some researchers have gone back inside to find out exactly what features of the world the hidden units responded to: Grandma's gray gloves? popcorn?—or something else altogether? To investigate the various features, in simulations of neural networks for instance, researchers make use of a program that monitors how active each of the hidden units is.

To analyze hidden units is tricky, and it is not routinely done. For one thing, specifying exact features is a painstaking process. At Johns Hopkins, Terry Sejnowski's *NetTalk* has 18,000 connections to analyze—a year's work. For another, the difficulty depends on the nature and complexity of the problem. In harder problems, says James Anderson at Brown University, "units turn out to respond to all kinds of things."

If the researcher understands the problem and knows how a stimulus ought to break down under analysis, the features that hidden units map or correspond to make sense. Sejnowski and Princeton's Charles Rosenberg found that *NetTalk*'s hidden layer performed coding for particular letter-to-sound relationships—some consonants, some vowels. Within each vowel or consonant, *NetTalk* had also isolated individual clusters of phonemes. That kind of obviousness, said Anderson, "gives the researcher the warm, toasty feeling that the net operates the way you do and you understand it."

Sometimes the hidden units see features that are sensible but less obvious. David Rumelhart, Geoffrey Hinton, and Ronald Williams (all of the University of California at San Diego) fashioned an experiment that gave a five-layer network two family trees of the same shape—one Italian, one English. They trained the network in the people's names and such relationships as "Colin is the son of Victoria," "Victoria is the sister of Arthur."

The network, if given Colin and "nephew of," could supply the name of Victoria's brother, Arthur. Because of the similarity between family trees, the network could generalize from English family relationships to Italian ones: If Alfonso is the son of Lucia, he must also be the nephew of Lucia's brother, Emilio. When the researchers looked at the connections between the input layer and the first of the hidden layers, they found a unit that distinguished between Italian and English, one that identified the particular generation, and one that identified the branch (left or right) of the family.

Sometimes, however, if the researcher does not know the answer to the problem or if the answer is one of several good possibilities, the features to which the hidden units respond do not make much sense. In the case of the English-Italian family problem, the network has many layers. Researchers could analyze the connections between the input layer and the first hidden layer. But after that, says Hinton, "the connections between the input layer and the second hidden layer are hard to even figure out, let alone see if they make sense. Obviously it's something sensible, because the network learns. It's just hard to see what connections are being made."

On the whole, says Anderson, "what happens in the hidden layer is a little mysterious. Loose thinking about hidden units has always haunted the field of learning systems. It's a mysticism I'm not happy with."

Sejnowski, by contrast, views the prospects less bleakly. "In every problem that I've looked at," he says, "from *NetTalk* to sonar target identification, not only have we been able to understand the hidden units but we have also learned a lot about the nature of the problem. There is nothing mysterious about it." •

the optical neural net contains what Psaltis calls a "threshold device," whose 10,000 sensors are triggered by a light beam. Instead of using resistors to vary the strength of connection, the optical neural net compares an input image—such as a slide projection of a ginkgo tree—to images of several trees stored in a hologram. The closer the match between the patterns of light and dark in the ginkgo tree and those of a tree in the hologram, the more intense the light that is allowed to leave the threshold device. After being trained, the optical neural computer can recognize a ginkgo tree even when shown only its parts. It could not, however, recognize a different ginkgo, nor could it identify the same tree from another angle.

Larry Jackel, an electrical engineer, working with Hans Peter Graf, a physicist also at Bell Laboratories, has built a neural chip that recognizes handwritten numbers from zero to nine. "You write on ordinary paper with an ordinary pen in ordinary handwriting," Jackel explains. "The network looks for the distinguishing features of the number: a three is three horizontal lines, three stops, and two vertical lines. If the handwriting is sloppy," says Jackel, "the net gets it with 80 percent accuracy; with neat writing, around 95 percent."

Toy problems

Once again, as they did with perceptrons, Minsky and Papert are poised to pounce. They are reissuing their book *Perceptrons*, with a new prologue and an epilogue to address directly the development of neural networks. "Our position remains what it was when we wrote the book," they write. "We believe this realm of work to be immensely important and rich, but we expect its growth to require a degree of critical analysis that its more romantic advocates have always been reluctant to pursue—perhaps because the spirit of connectionism seems itself to go somewhat against the grain of analytic rigor."

One problem with neural networks, say Minsky and Papert, is that neither the networks nor the problems they work through can be scaled up. Neural networks have solved only what Minsky and Papert call "toy problems": A given neural net can recognize a particular cat, but no neural network is equal to generalized pattern recognition. "A net can be tailored to do anything particular," said Minsky. "When someone says, 'I have a net that will solve this specific problem,' I stop listening."

Nor does anyone know, according to Minsky and Papert, the conditions under which bigger nets with more units arranged in more layers would solve harder problems. Part of the reason is that currently available learning algorithms will work on larger networks only if the tasks are, in the MIT authors' words, "of low order."

At the heart of the difficulty is the field's lack of a theory. The need is for a theory that "classifies the problems, then predicts what types of problems different machines can learn most efficiently," says Minsky. The problems that neural networks solve seem to defy classification. Learning to pronounce words seems to have little to do with finding the shortest route among 20 cities, which in turn seems unrelated to recognizing a ginkgo tree by its branches.

Tomaso Poggio, at MIT's Artificial Intelligence Laboratory, points out that neural networks cannot yet solve any problems that conventional computers could not also solve. "Neural networks are accompanied by a lot of irritating hype," Poggio declares. "Some comes from the press, a little of it from the scientists in the field. Neural nets point out interesting problems but have not solved the big problems of vision or speech.

Minsky. A cerebral role for toy problems?

Ultimately, in my view, when the hype disappears, there's a good possibility they will go the way of perceptrons."

Neural-network researchers do not dispute the scaling problems, the hype problem, or the fact that (as Hopfield says) "right now we can't do anything with neural networks that digital computers can't do." Or in Anderson's words, "It's a lot of fun, messing with

A theory from physics

Most of the work in neural networks has been experimental—trying out various problems on various architectures, applying various learning algorithms. Theoretical understanding of what goes on in the networks, however, is harder to come by. So far, the best theory is Hopfield's. Hopfield applied to networks the most basic of all sciences: physics. According to psychologist Anderson, until Hopfield's entry into the field "nobody paid much attention to the psychologists—psychologists are 'squishy.' A Caltech physicist, though, is impressive."

To describe what happens in neural networks as they learn, Hopfield used a model that physicists know well: the Ising. Ising models describe such systems as magnets or spin glasses in which large numbers of two-state units change their states in dependence on their neighbors' states. "Hidden in the mathematics of the Ising model," says Hopfield, "is a quantity which, as processing units change their firing, always decreases." For lack of a better name, Hopfield calls that quantity energy: "The quantity simply somehow drives the change, just as entropy increase drives the expansion of a gas. It's a mathematical construct of 'downhill.' "

The energy in the whole system goes downhill whenever the units change states. The units stop changing, says Hopfield, when "each unit agrees with all of its inputs." If a unit's inputs are positive, the unit is on; if negative, it is off. "After changes toward consensus, the system doesn't change any more, the energy gets no lower. Once at the lowest state, you don't go anywhere else because there's nowhere else to go, no more reason to roll." •

Biological neural networks

Connectionists, those computer scientists and engineers devising what they metaphorically call neural networks, define their networks as collections of electronic surrogate neurons connected in ways that are designed to achieve some specified function: They can solve the traveling salesman problem; they can learn to pronounce words.

Neurobiologists accept this definition, but with reservations; specific networks indeed should be responsible for specific functions: digestion or spatial orientation to name two. In practice, however, the practical relationship between network and function in neurobiology is neither necessary nor clear, and battle lines do get drawn. "To a neurobiologist," says David Hubel, Nobel Prize winner and Harvard University neurobiologist, "network is just a gimmicky term to mean connected neurons." If a network's identity is constrained to include its function, says Larry Cohen of the Yale University School of Medicine, "we're still trying to define network and, depending on the scale, sometimes we're not even close."

Largest and smallest

In a sense, all neurobiologists who study collections of connected neurons can be said to be studying neural networks. The scales, however, can range from pairs of neurons to millions of neurons in animals ranging from nematodes to humans, and differences in scale can produce differences in perspective.

Neurobiologists studying the largest scales in the most complex animals do know function: The 10 billion neurons in the human cerebellum coordinate the initiation and control of movement. (John Kauer, neurobiologist at Tufts Medical School, advises care in talking about numbers of neurons. "Nobody's done a good count in the human brain—one number often used is 10 billion. The counts in the cerebellum have been more careful, and that's 10 billion, too. So we tell our students the brain has 10 billion neurons, 10 billion of which are in the cerebellum.")

Kauer and his colleagues relate structure to function by presenting a subject with a stimulus—a moving black speck—and then watching to see which area of the brain becomes most active. Activity is measured with electroencephalograms that trace electrical responses in areas of the brain, or with positron emission tomography that traces the areas of the brain that most take up radioactive glucose and therefore use the most energy. The oldest method for finding the functions of areas of the brain is to see what happens when something goes wrong with one area. Neurobiologists studying the largest scales, however, do not say they are studying networks of neurons; on this scale, connections between neurons are impossibly labyrinthine and numerous.

Neurobiologists studying the smallest scales in the simplest animals know the functions of some of the networks—the 14 neurons in a ganglion in a lobster's stomach, for example, that move the stomach back and forth to grind food. Neurobiologists working on this scale relate structure to function in the most direct way possible: by presenting a stimulus using an electrode to detect a specific neuron's response, and then inserting other electrodes and watching other neurons' responses.

They have found neurons in frogs that discriminate between a small black moving object and a large one, and neurons in rats that respond to one smell or to being in a particular place. They have found that neurons responding to movement do not also respond to color. They have charted all 300 neurons in the nematode *C. elegans*. One hundred of the 10,000 neurons in the well-studied sea slug *Aplysia* have names.

"Only the single-cell guys," says Larry Cohen, "can say an action potential in cell *X* causes a synaptic potential in cell *Y*, which causes a muscle to contract." Single-cell research provides the most compelling link between structure and function, says David Hubel. "It is what neurobiology can with most justification call a network."

The drawback of single-cell research is size. Electrodes can be inserted only in tens of neurons at a time. Hundreds of electrodes in hundreds of neurons, says Cohen, "would make scrambled eggs of the brain. But try as it will the 14-cell network in the lobster's stomach can't do the traveling salesman problem."

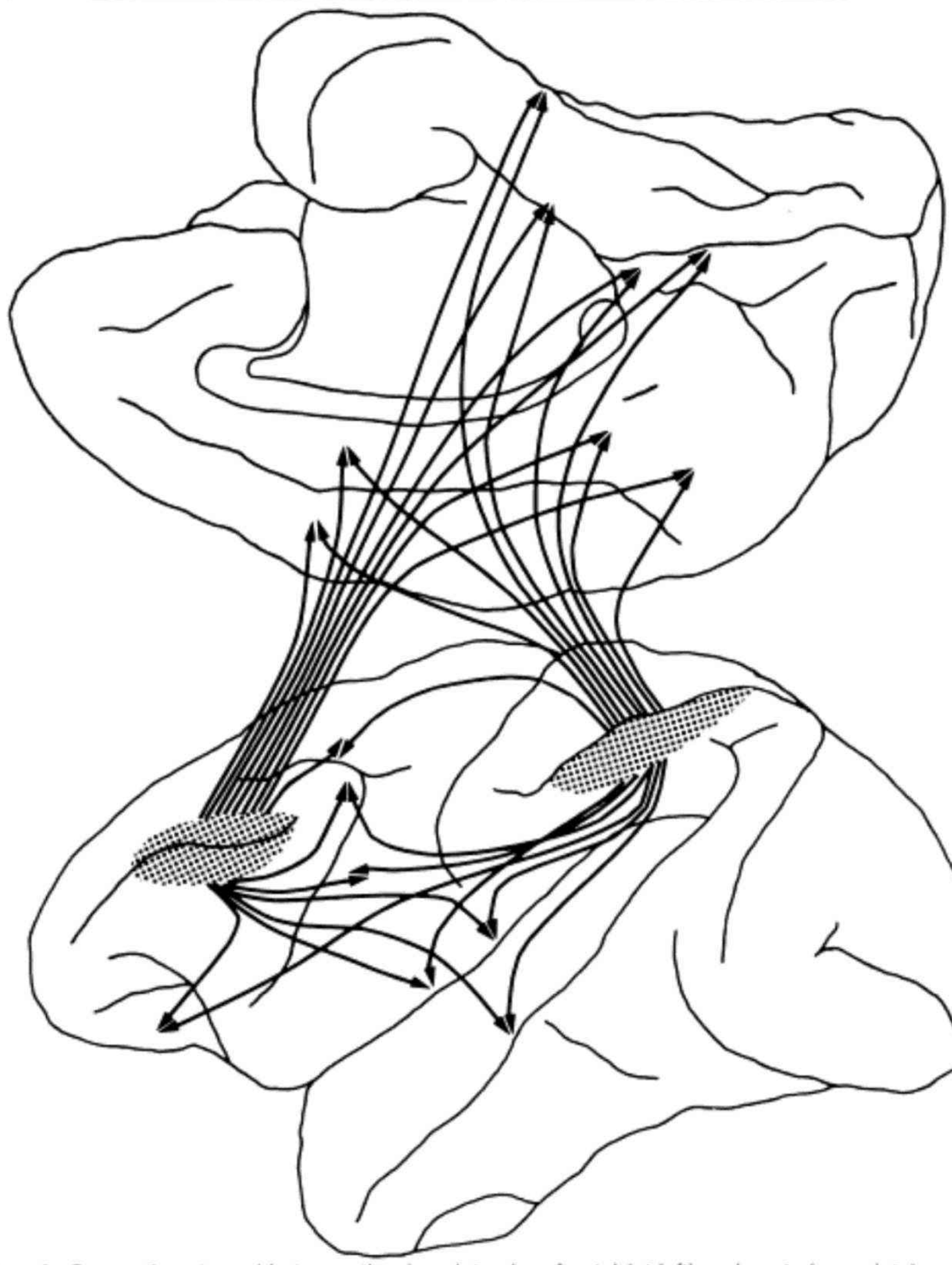
Midrange

The midrange networks in humans are most likely to be the ones solving the traveling salesman problem and learning to pronounce. Unlike single-cell networks, midrange networks can be related to specific functions only indirectly. If a puff of odor excites a salamander's olfactory bulb, then a network of neurons in the olfactory bulb probably has something to do with smell. Midrange researchers study networks on the order of thousands to millions of cells situated in the visual, motor, and olfactory areas of brains of animals including salamanders, rattlesnakes, gerbils, opossums, rats, cats, and monkeys.

Patricia Goldman-Rakic, neurobiologist at the Yale School of Medicine, studies monkey-brain networks that probably are responsible for processing information on the monkey's position in space. Goldman-Rakic uses a combination of radioactive tracers that are taken up by cells in the region in which they are injected. The tracers travel along axons, marking where the axons end. After the tracers have reached their destinations, the animals are sacrificed and slices of their brains are studied.

Goldman-Rakic and her colleagues have identified a network connecting some 17 different areas of the cortex. These areas include the prefrontal cortex, responsible for memories of spatial information; the posterior parietal cortex, responsible for the monkey's knowing where it is in space relative to its environment; and the parahippocampal gyrus, responsible for the long-term storage of memory. Some areas in the same network are responsible for sensory information, others have access to motor centers.

"Several brain centers may be interconnected in the net-



After L. D. Selemon, P. S. Goldman-Rakic, *J. of Neurosciences*, in press 1988.

Neural network. Connections traced between the dorsolateral prefrontal (at left) and posterior parietal cortices of a rhesus monkey's brain. Neuronal bundles link regions of the brain that contribute to the monkey's ability to locate itself in space.

work such that function is distributed among the interconnected areas," Goldman-Rakic writes in a forthcoming issue of the *Journal of Neuroscience*. "Furthermore, we suggest that this network is involved in all aspects of spatial perception and behavior, including attention, perception, memory, and motor control."

John Kauer also works on midrange networks, but in salamanders and in real time. Kauer exposes a salamander's olfactory bulb and applies to it a dye that fluoresces when it encounters electrical voltage. Then Kauer shocks the nerve leading to the olfactory bulb and, as successive neurons fire, the fluorescence spreads. "It's like having electrodes everywhere throughout the tissue," he says.

Kauer's videotape shows differing areas of the olfactory bulb fluorescing with differing intensities. Each intensity may represent activity in a different network. But because specifics about the way a salamander's olfactory bulb handles odors are less well known than specifics about the way a primate's motor cortex handles movement, Kauer cannot determine exactly what these networks do.

Although the tracers and dyes used by both Goldman-Rakic and Kauer spread neuron by neuron, neither researcher is detecting networks of single neurons. Instead, each point of the spread probably represents hundreds to thousands of neurons. What the researcher sees are networks of networks; David Hubel says they should be called supernetworks.

Goldman-Rakic says she hopes that connectionists will use the results of research in midrange networks in their models. "It should be useful to the modelers that there are 17 areas in a monkey's cortex dealing with spatial information," she says. —*Ann Finkbeiner*

these models. These little straightforward simulations are provocative. But now we have to do the hard stuff." That includes the finding of more powerful learning algorithms, solutions to more general problems, development of bigger machines, and the invention of some sort of unifying concept. "We're just starting to realize how hard the hard stuff is," says Anderson.

Cui bono?

In the end, neural networks as a field seems more suggestive than substantive. Part of the reason is that the field is young. Another part of the reason is that, like AI, neural networks look two ways simultaneously: toward building better computers and toward understanding the mind and/or brain. So the irresistible question arises: What drives research into neural networks? or: Of all the researchers in all the fields that work on neural networks, who benefits?

So far, not neurobiologists; but that is beginning to change. Some neurobiologists find that neurons behave according to more-or-less Hebbian models of synaptic modification and that the brain does indeed store information in modified synapses. Others use neural-network techniques to study the way the olfactory system recognizes different smells. But for neurobiologists, says Anderson, "neural networks are so extreme a stylization of the brain it's like describing the Rockies by saying they're higher than the surrounding land." On the whole, says Bell's Larry Jackel, "no one really believes electronics can literally imitate biology."

Neural-net researchers would reply that duplicating the brain is not their goal. They do not use silicon to mimic biology, Hopfield and David Tank of Bell Laboratories have written, for the same reason that carmakers do not make mechanical horses and airplanes do not flap their wings. What silicon can mimic, says Sejnowski, are the principles behind the biology. And that work, he says, "is teaching us more and more about styles of computation."

So maybe computer science is the beneficiary. The technology of neural networks may apply to the next generation of computers. The programs, the mathematics required to write the programs, the arrangement of the processing units, and the connections between them—all present alternatives to today's digital computers. Several companies are already manufacturing commercial neural networks that can identify vehicles by their sonar or radar images and can read handwriting not only in cursive English, but in Japanese. Other companies manufacture prototype neural chips. Jackel says that computer science pays attention to neural-net research, but is waiting: "The issue now is delivering the goods, seeing what works."

For cognitive science, the resemblance of neural networks to minds, though inexact, is also fascinating. People learn without consciously applying logic. They routinely wipe out neurons with alcohol, drugs, and accidents, but rarely lose their memories of Grandmother. They find answers that are good enough, if not the best possible. They come up with associations they never learned.

For example, James McClelland, a cognitive psychologist at Carnegie Mellon University, compared the performance of children to that of a neural network on the same problem. The problem: A balance beam is divided evenly along its length by several pins. If equal weights are laid on the pins third from the fulcrum on both sides, the beam will balance. What will happen if two weights are placed on the second pin out on the left, and one weight on the fourth pin out on the right? "Both the kids and the nets were confused," McClelland reports, "and like the kid who never goes on to take general physics, [the network] never learns to say that one weight on the fourth pin equals two weights on the second." In other words, general-physics students, like AI programs, can solve the problem by multiplying. Neural networks and (presumably) English majors, remain confused and, says McClelland, "favor the weight cue from the beginning."

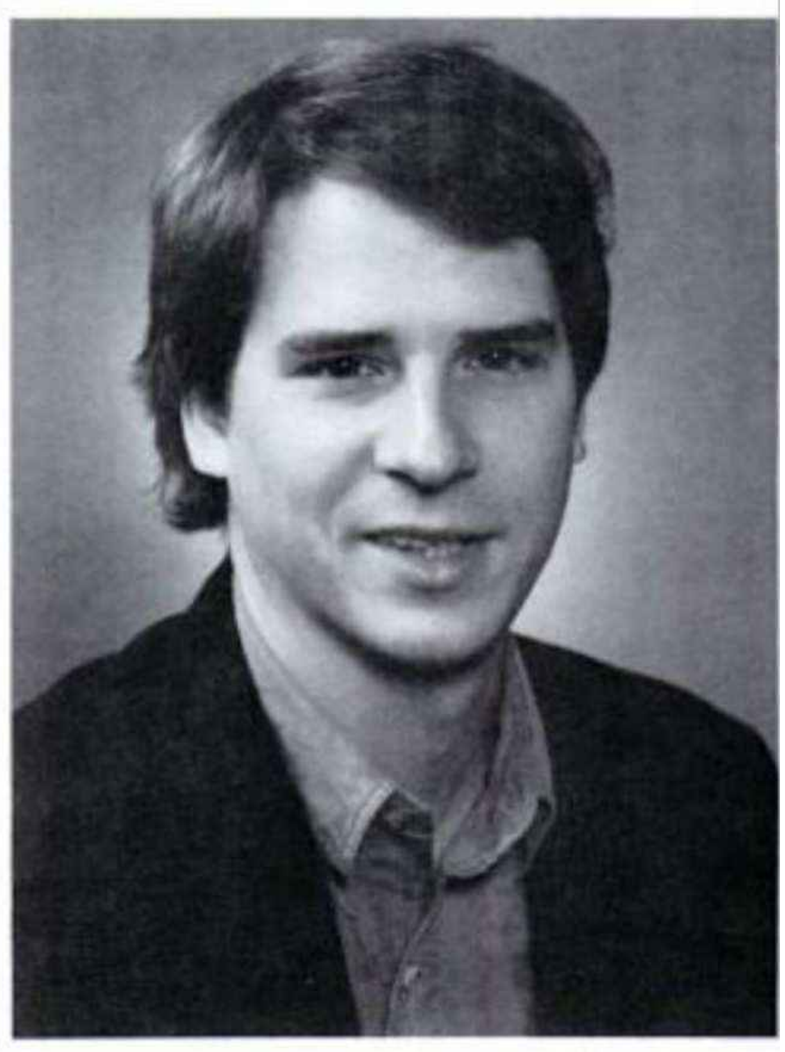
But the resemblances between neural-network performances and human mental abilities, however intriguing, point to no general principles of cognition. Neural networks have given cognitive scientists interesting things to think about, McClelland says: "how we might represent the world, how we might learn from experience, how we might apply past experience to new information in a world where inputs are ambiguous and constraints are soft." Nevertheless, neural networks can only suggest questions. Resemblance is not identity; analogy is never proof.

The future of neural networks will certainly hold benefits for computer science, probably for cognitive science, maybe even for neurobiology. But for now the field has an atmosphere of playfulness and curiosity, of trying to see, in McClelland's words, "how much we can squeeze out of this idea." If science is the process of finding precise and useful metaphors, then, says McClelland, "we're looking for the extent to which reality is mirrored in the metaphor."

Weighing the metaphor

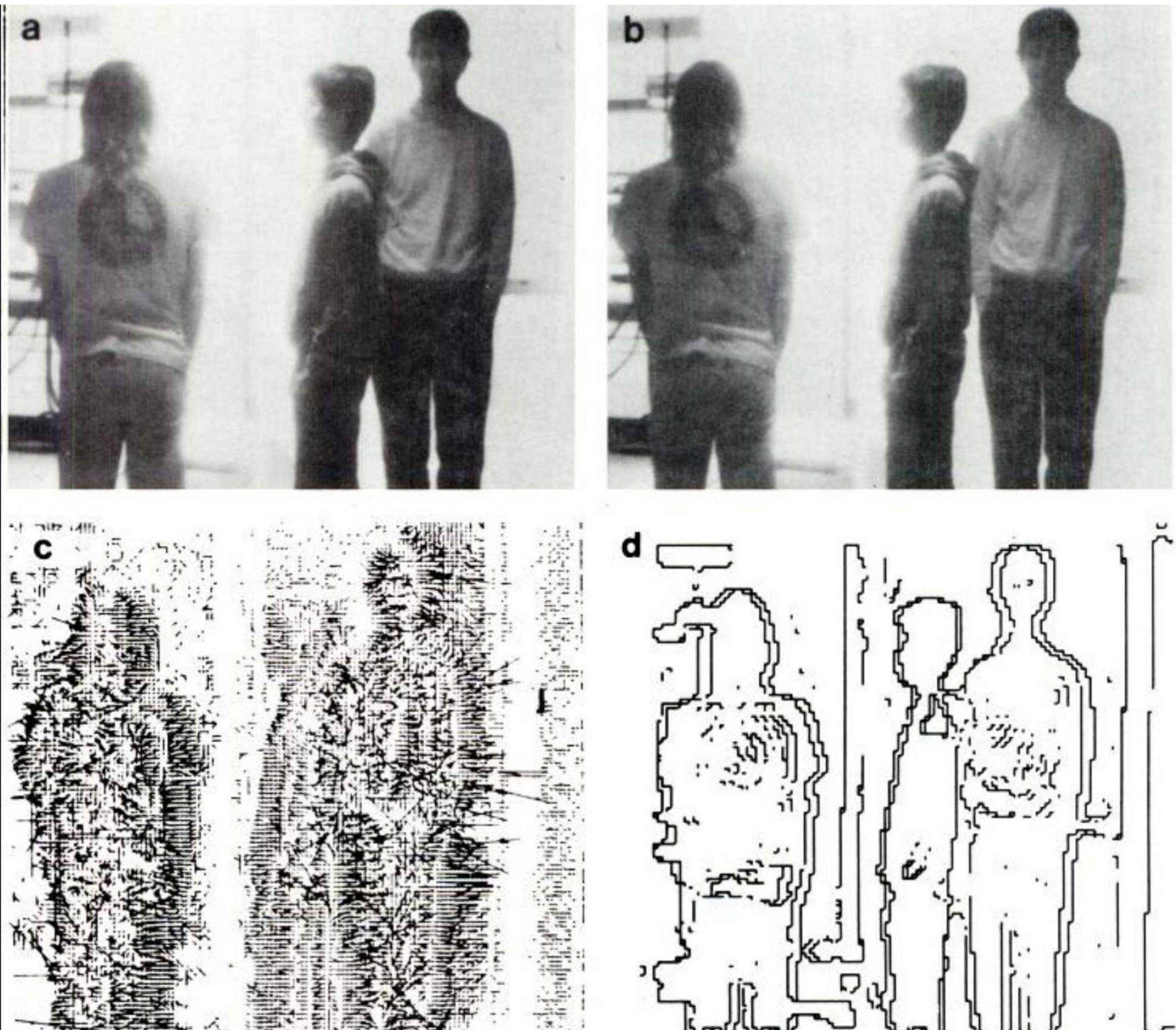
Both neural-network and AI researchers agree that a binding choice among brain-machine models need not be made. "Conventional computers are so good at what they do," says AT&T's Larry Jackel, "you'd be crazy not to use them, to use only the neural approach. You need to suit the machine to the problem."

Marvin Minsky and Seymour Papert, AI researchers at MIT, agree. They have criticized neural networks as limited to



solving what they call "toy problems" as opposed to having a more general pattern-recognition ability. Nevertheless, Minsky and Papert propose, "toy problems" may be less a limitation than a prototype: "Perhaps the scale of the toy problem is that on which, in physiological actuality, much of the functioning of intelligence operates."

In the epilogue to their reissued *Perceptrons*, Minsky and Papert argue that the human brain is more or less built up of many small neural networks. Each small network solves a few interrelated "toy problems." Late in the development of the embryo, nature adds a "se-



Defining movement. Koch (left) and an example of his neural net (above). (A) and (B) are before-and-after video images of moving people. (C) is a grid of vectors representing movement. The application of a neural net to the grid produces an image with well-defined lines (D).

rial system," like an AI program, that directs the smaller networks. "And that leads us to ask," write Minsky and Papert, "how such systems could develop managers for deciding, in different circumstances, which of those diverse procedures to use."

So the proper focus of research in brain machines, they contend, is not the search for universal principles but the search for what kinds of processing best serve which kinds of problems: "In fact, research on networks in which different parts do different things—and learn those things in different ways—has become our principal personal concern."

Ultimately, Minsky and Papert conclude, they see no need to choose between AI and neural networks: "Both are partial and manifestly useful views of a reality of which science is still far from a comprehensive understanding. . . . Maybe, since the brain is a hierarchy of systems the best machine will be too."

Herbert Simon, who won a Nobel Prize for his work with computers, agrees with Minsky and Papert: "Any valid theory of how the mind works will have to explain what part neurons play in symbol processing. That's a lot like asking what part quarks play in hydrogen bonds. The field of neural networks might help do that. It's certainly not there yet. But we have to push it to get it there." •

The National Science Foundation contributes to the support of the areas of research described in this article principally through its Neural Engineering and Memory and Cognitive Processes programs, and several other programs in its Division of Behavioral and Neural Sciences.