AI Magazine Volume 12 Number 2 (1991) (© AAAI)

Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy

Marvin Minsky

Engineering and scientific education condition us to expect everything, including intelligence, to have a simple, compact explanation. Accordingly, when people new to AI ask "What's AI all about," they seem to expect an answer that defines AI in terms of a few basic mathematical laws.

Today, some researchers who seek a simple, compact explanation hope that systems modeled on neural nets or some other connectionist idea will quickly overtake more traditional systems based on symbol manipulation. Others believe that symbol manipulation, with a history that goes back millennia, remains the only viable approach.

Marvin Minsky subscribes to neither of these extremist views. Instead, he argues that AI must use many approaches. AI is not like circuit theory and electromagnetism. There is nothing wonderfully unifying like Kirchhoff's laws are to circuit theory or Maxwell's equations are to electromagnetism. Instead of looking for a "right way," the time has come to build systems out of diverse components, some connectionist and some symbolic, each with its own diverse justification.

—Patrick Winston

Why is there so much excitement about neural networks today, and how is this related to research in AI? Much has been said, in the popular press, as though these were conflicting activities. This seems exceedingly strange to me because both are parts of the same enterprise. What caused this misconception?

The symbol-oriented community in AI has brought this rift upon itself by supporting models in research that are far too rigid and specialized. This focus on well-defined problems produced many successful applications, no matter that the underlying systems were too inflexible to function well outside the domains for which they were designed. (It seems to me that this occurred because of the researchers' excessive concern with logical consistency and provability. Ultimately, this concern would be a proper one but not in the subject's current state of immaturity.) Thus, contemporary symbolic AI systems are now too constrained to be able to deal with exceptions to rules or to exploit fuzzy, approximate, or heuristic fragments of knowledge. Partly in reaction to this constraint, the connectionist movement initially tried to develop more flexible systems but soon came to be imprisoned in its own peculiar ideology—trying to build learning systems endowed with as little architectural structure as possible, hoping to create machines that could serve all masters equally well. The trouble with this attempt is that even a seemingly neutral architecture still embodies an implicit assumption about which things are presumed to be similar.

The field called AI includes many different aspirations. Some researchers simply want machines to do the various sorts of things that people call intelligent. Others hope to understand what enables people to do such things. Still other researchers want to simplify programming. Why can't we build, once and for all, machines that grow and improve themselves by learning from experience? Why can't we simply explain what we want, and then let our machines do experiments or read some books or go to school—the sorts of things that people do. Our machines today do no such things: Connectionist networks learn a bit but show few signs of becoming smart; symbolic systems are shrewd from the start but don't yet show any common sense. How strange that our most advanced systems can compete with human specialists yet are unable to do many things that seem easy to children. I suggest that this stems from the nature of what we call *specialties*—because the very act of naming a specialty amounts to celebrating the discovery of some model of some aspect of reality, which is useful despite being isolated from most of our other concerns. These models have rules that reliably work—as long as we stay in their special domains. But when we return to the commonsense world, we rarely find rules that precisely apply. Instead, we must know how to adapt each fragment of knowledge to particular contexts and circumstances, and we must expect to need more and different kinds of knowledge as our concerns broaden. Inside such simple "toy" domains, a rule might seem to be general, but whenever we broaden these domains, we find more and more exceptions, and the early advantage of context-free rules then mutates into strong limitations.

AI research must now move from its traditional focus on particular schemes. There is no one best way to represent knowledge or to

…the time has come to build systems out of diverse components…

0738-4602/91/$4.00 ©1991 AAAI

Figure 1. Conflict between theoretical extremes.

solve problems, and the limitations of current machine intelligence largely stem from seeking unified theories or trying to repair the deficiencies of theoretically neat but conceptually impoverished ideological positions. Our purely numeric connectionist networks are inherently deficient in abilities to reason well; our purely symbolic logical systems are inherently deficient in abilities to represent the all-important *heuristic connections* between things—the uncertain, approximate, and analogical links that we need for making new hypotheses. The versatility that we need can be found only in larger-scale architectures that can exploit and manage the advantages of several types of representations at the same time. Then, each can be used to overcome the deficiencies of the others. To accomplish this task, each formally neat type of knowledge representation or inference must be complemented with some scruffier kind of machinery that can embody the heuristic connections between the knowledge itself and what we hope to do with it.

Top Down versus Bottom Up

Although different workers have diverse goals, all AI researchers seek to make machines that solve problems. One popular way to pursue this quest is to start with a top-down strategy: Begin at the level of commonsense psychology, and try to imagine processes that could play a certain game, solve a certain kind of puzzle, or recognize a certain kind of object. If this task can't be done in a single step, then break things down into simpler parts until you can actually embody them in hardware or software.

This basically reductionist technique is typical of the approach to AI called *heuristic programming*. These techniques have developed productively for several decades, and today, heuristic programs based on top-down analysis have found many successful applications in technical, specialized areas. This progress is largely the result of the maturation of many techniques for representing knowledge. However, the same techniques have seen less success when applied to commonsense problem solving. Why can we build robots that compete with highly trained workers to assemble intricate machinery in factories but not robots that can help with ordinary housework? It is because the conditions in factories are constrained, and the objects and activities of everyday life are too endlessly varied to be described by precise, logical definitions and deductions. Commonsense reality is too disorderly to represent in terms of universally valid axioms. To deal with such variety and novelty, we need more flexible styles of thought, such as those we see in human commonsense reasoning, which is based more on analogies and approximations than on precise formal procedures. Nonetheless, top-down procedures have important advantages in being able to perform efficient, systematic search procedures, manipulate and rearrange the elements of complex situations, and supervise the management of intricately interacting subgoals—all functions that seem beyond the capabilities of connectionist systems with weak architectures.

Shortsighted critics have *always* complained that progress in top-down symbolic AI research is slowing. In one way, this slowing is natural: In the early phases of any field, it becomes ever harder to make important new advances as we put the easier problems behind us; in addition, new workers must face a squared challenge because there is so much more to learn. However, the slowdown of progress in symbolic AI is not just a matter of laziness. Those top-down systems are inherently poor at solving problems that involve large num-

bers of weaker kinds of interactions such as occur in many areas of pattern recognition and knowledge retrieval. Hence, there has been a mounting clamor for finding another, new, more flexible approach, which is one reason for the recent popular turn toward connectionist models.

The bottom-up approach goes the opposite way. We begin with simpler elements—they might be small computer programs, elementary logical principles, or simplified models of what brain cells do—and then move upward in complexity by finding ways to interconnect these units to produce larger-scale phenomena. The currently popular form of this, the connectionist neural network approach, developed more sporadically than heuristic programming. This development was sporadic in part because heuristic programming developed so rapidly in the 1960s that connectionist networks were swiftly outclassed. Also, the networks needed computation and memory resources that were too prodigious for that period. Now that faster computers are available, bottom-up connectionist research has shown considerable promise in mimicking some of what we admire in the behavior of lower animals, particularly in the areas of pattern recognition, automatic optimization, clustering, and knowledge retrieval. However, their performances have been far weaker in precisely the areas in which symbolic systems have successfully mimicked much of what we admire in high-level human thinking, for example, in goal-based reasoning, parsing, and causal analysis. These weakly structured connectionist networks cannot deal with the sorts of tree search explorations and complex, composite knowledge structures required for parsing, recursion, complex scene analysis, or other sorts of problems that involve *functional* parallelism. It is an amusing paradox that connectionists frequently boast about the massive parallelism of their computations, yet the homogeneity and interconnectedness of these structures make them virtually unable to do more than one thing at a time—at least, at levels above that of their basic associative function. This is essentially because they lack the architecture needed to maintain adequate short-term memories.

Thus, the current systems of both types show serious limitations. The top-down systems are handicapped by inflexible mechanisms for dealing with very numerous, albeit very weak, interactions, while the bottom-up systems are crippled by inflexible architectures and organizational limitations. Neither type of system has been developed to be able to exploit multiple, diverse varieties of knowledge.

Which approach is best to pursue? This question itself is simply wrong. Each has virtues and deficiencies, and we need integrated systems that can exploit the advantages of both. In favor of the top-down side, AI research has told us a little—but only a little—about how to solve problems by using methods that resemble reasoning. If we understood more about such processes, perhaps we could more easily work down toward finding out how brain cells do such things. In favor of the bottom-up approach, the brain sciences have told us something—but again only a little—about the workings of brain cells and their connections. More research in this area might help us discover how the activities of brain cell networks support our higher-level processes. However, right now we're caught in the middle; neither purely connectionist nor purely symbolic systems seem able to support the sorts of intellectual performances we take for granted even in young children. This article aims at understanding why both types of AI systems have developed to become so inflexible. I'll argue that the solution lies somewhere between these two extremes, and our problem will be to find out how to build a suitable bridge. We already have plenty of ideas at either extreme. On the connectionist side, we can extend our efforts to design neural networks that can learn various ways to represent knowledge. On the symbolic side, we can extend our research on knowledge representations to the designing of systems that can more effectively exploit the knowledge thus represented. However, above all, we currently need more research on how to combine both types of ideas.

Representation and Retrieval: Structure and Function

In order that a machine may learn, it must represent what it will learn. The knowledge must be embodied in some form of mechanism, data structure, or representation. AI researchers have devised many ways to embody this knowledge, for example, in the forms of rule-based systems, frames with default assignments, predicate calculus, procedural representations, associative databases, semantic networks, object-oriented datastructures, conceptual dependency, action scripts, neural networks, and natural language.

In the 1960s and 1970s, students frequently asked, "Which kind of representation is best," and I usually replied that we'd need

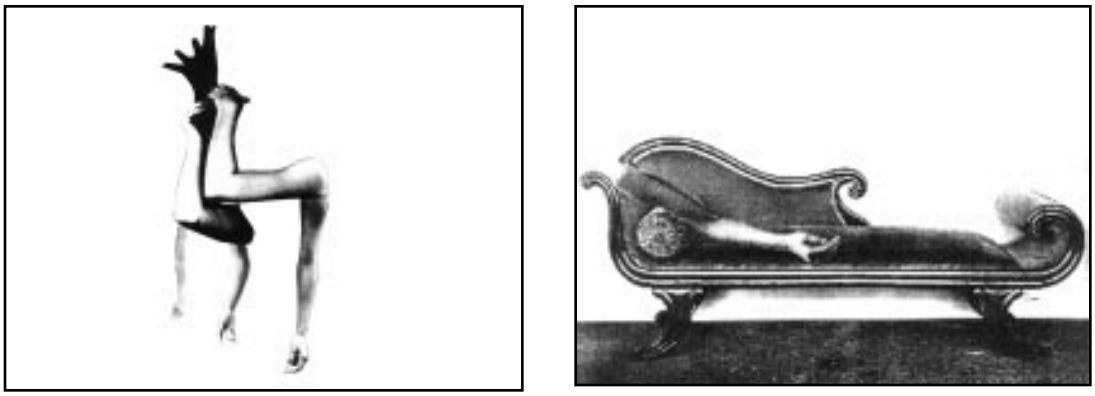
We should take our cue from biology rather than physics…

more research before answering. But now I would give a different reply: "To solve really hard problems, we'll have to use several different representations." This is because each particular kind of data structure has its own virtues and deficiencies, and none by itself seems adequate for all the different functions involved with what we call common sense. Each has domains of competence and efficiency, so that one might work where another fails. Furthermore, if we only rely on any single, unified scheme, then we'll have no way to recover from failure. As suggested in section 6.9 of *The Society of Mind* (Minsky 1987), "The secret of what something means lies in how it connects to other things we know. That's why it's almost always wrong to seek the *real meaning* of anything. A thing with just one meaning has scarcely any meaning at all."

To get around these limitations, we must develop systems that combine the expressiveness and procedural versatility of symbolic systems with the fuzziness and adaptiveness of connectionist representations. Why has there been so little work on synthesizing these techniques? I suspect that it is because both of these AI communities suffer from a common cultural-philosophical disposition: They would like to explain intelligence in the image of what was successful in physics—by minimizing the amount and variety of its assumptions. But this seems to be a wrong ideal. We should take our cue from biology rather than physics because what we call thinking does not directly emerge from a few fundamental principles of wave-function symmetry and exclusion rules. Mental activities are not the sort of unitary or elementary phenomenon that can be described by a few mathematical operations on logical axioms. Instead, the functions performed by the brain are the products of the work of thousands of different, specialized subsystems, the intricate product of hundreds of millions of years of biological evolution. We cannot hope to understand such an organization by emulating the techniques of those particle physicists who search for the simplest possible unifying conceptions. Constructing a mind is simply a different kind of problem—how to synthesize organizational systems that can support a large enough diversity of different schemes yet enable them to work together to exploit one another's abilities.

To solve typical real-world commonsense problems, a mind must have at least several different kinds of knowledge. First, we need to represent goals: What is the problem to be solved. Then, the system must also possess adequate knowledge about the domain or context in which this problem occurs. Finally, the system must know what kinds of reasoning are applicable in this area. Superimposed on all this knowledge, our systems must have management schemes that can operate different representations and procedures in parallel, so that when any particular method breaks down or gets stuck, the system can quickly shift to analogous operations in other realms that might be able to continue the work. For example, when you hear a natural language expression such as "Mary gave Jack the book," you will produce, albeit unconsciously, many different kinds of thoughts (Minsky [1987], section 29.2), that is, mental activities in such different realms as a visual representation of the scene; postural and tactile representations of the experience; a script sequence for a typical act of giving; representations of the participants' roles; representations of their social motivations; default assumptions about Jack, Mary, and the book; and other assumptions about past and future expectations.

How could a brain possibly coordinate the use of such different kinds of processes and representations? The conjecture is that our brains construct and maintain them in different brain agencies. (The corresponding neural structures need not, of course, be entirely separate in their spatial extents inside the brain.) However, it is not enough to maintain separate processes inside separate agencies; we also need additional mechanisms to enable each of them to support the activities of the others or, at least, to provide alternative operations in case of failures. Chapters 19 through 23 of *The Society of Mind* (Minsky 1987) sketch some ideas about how the representations in different agencies could be coordinated. These sections introduce the concepts of the *polyneme*, a hypothetical neuronal mechanism for activating corresponding slots in different representations; the *microneme*, a context-representing mechanism that similarly biases all the agencies to activate knowledge related to the current situation and goal; and the *paranome*, yet another mechanism that can simultaneously apply corresponding processes or operations to the short-term



Figures 2A and 2B. Armchair

memory agents—called *pronomes*—of these various agencies.

It is impossible to briefly summarize how all these mechanisms are imagined to work, but section 29.3 of *The Society of Mind* (Minsky 1987) gives some of the flavor of the theory. What controls those paranomes? I suspect that in human minds, this control comes from the mutual exploitation of a long-range planning agency (whose scripts are influenced by various strong goals and ideals; this agency resembles the Freudian superego and is based on early imprinting), another supervisory agency capable of using semiformal inferences and natural language reformulations, and a Freudian-like censorship agency that incorporates massive records of previous failures of various sorts.

Relevance and Similarity

Problem solvers must find relevant data. How does the human mind retrieve what it needs from among so many millions of knowledge items? Different AI systems have attempted to use a variety of different methods for this. Some assign keywords, attributes, or descriptors to each item and then locate data by feature-matching or using more sophisticated associative database methods. Others use graph matching or analogical case-based adaptation. Still others try to find relevant information by threading their way through systematic, usually hierarchical classifications of knowledge—sometimes called *ontologies*. To me, all such ideas seem deficient because it is not enough to classify items of information simply in terms of the features or structures of the items themselves: We rarely use a representation in an intentional vacuum, but we always have goals—and two objects might seem similar for one purpose but different for another purpose. Consequently, we must also account for the functional aspects of what we know, and therefore, we must classify things (and ideas) according to what they can be used for or which goals they can help us achieve. Two armchairs of identical shape might seem equally comfortable as objects for sitting in, but these same chairs might seem very different for other purposes, for example, if they differ much in weight, fragility, cost, or appearance. The further a feature or difference lies from the surface of the chosen representation, the harder it will be to respond to, exploit, or adapt to, which is why the choice of representation is so important. In each functional context, we need to represent particularly well the heuristic connections between each object's internal features and relationships and the possible functions of that object. That is, we must be able to easily relate the structural features of each object's representation to how this object might behave in regard to achieving our current goals (see sections 12.4, 12.5, 12.12, and 12.13, Minsky [1987]).

New problems, by definition, are different from those we have already encountered; so, we cannot always depend on using records of past experience. However, to do better than random search, we have to exploit what was learned from the past, no matter that it might not perfectly match. Which records should we retrieve as likely to be the most relevant?

Explanations of relevance in traditional theories abound with synonyms for nearness and similarity. If a certain item gives bad results, it makes sense to try something different. However, when something we try turns out to be good, then a similar one might be better. We see this idea in myriad forms, and whenever we solve problems, we find ourselves using metrical metaphors: We're "get-

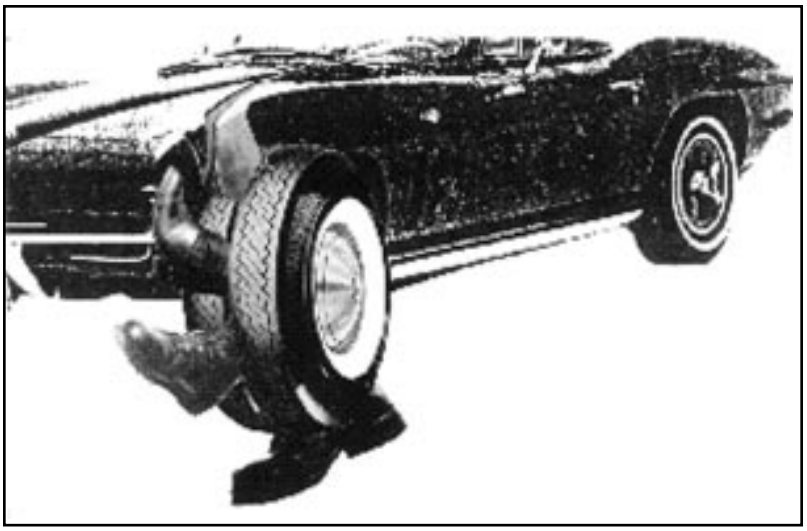


Figure 3. Functional similiarity

ting close" or "on the right track," using words that express proximity. But what do we mean by "close" or "near?" Decades of research on different forms of this question have produced theories and procedures for use in signal processing, pattern recognition, induction, classification, clustering, generalization, and so on, and each of these methods has been found useful for certain applications but ineffective for others. Recent connectionist research has considerably enlarged our resources in these areas. Each method has its advocates, but I contend that it is now time to move to another stage of research: Although each such concept or method might have merit in certain domains, none of them seem powerful enough alone to make our machines more intelligent. It is time to stop arguing over which type of pattern-classification technique is best because that depends on our context and goal. Instead, we should work at a higher level of organization and discover how to build managerial systems to exploit the different virtues and evade the different limitations of each of these ways of comparing things. Different types of problems and representations may require different concepts of similarity. Within each realm of discourse, some representation will make certain problems and concepts appear more closely related than others. To make matters worse, even within the same problem domain, we might need different notions of similarity for descriptions of problems and goals, descriptions of knowledge about the subject domain, and descriptions of procedures to be used.

Figure 4. "Heureka!"

For small domains, we can try to apply all our reasoning methods to all our knowledge and test for satisfactory solutions. However, this approach becomes impractical when the search becomes too huge—in both symbolic and connectionist systems. To constrain the extent of mindless search, we must incorporate additional kinds of knowledge, embodying expertise about problem solving itself and, particularly, about managing the resources that might be available. The spatial metaphor helps us think about such issues by providing us with a superficial unification: If we envision problem solving as searching for solutions in a spacelike realm, then it is tempting to analogize between the ideas of similarity and nearness, to think about similar things as being in some sense near or close to one another.

But near in what sense? To a mathematician, the most obvious idea would be to imagine the objects under comparison to be like points in some abstract space; then each representation of this space would induce (or reflect) some sort of topologylike structure or relationship among the possible objects being represented. Thus, the languages of many sciences, not merely those of AI and psychology, are replete with attempts to portray families of concepts in terms of various sorts of spaces equipped with various measures of similarity. If, for example, you represent things in terms of (allegedly independent) properties, then it seems natural to try to assign magnitudes to each and then to sum the squares of their differences—in effect, representing these objects as vectors in Euclidean space. This approach

further encourages us to formulate the function of knowledge in terms of helping us to decide "which way to go." This method is often usefully translated into the popular metaphor of hill climbing because if we can impose a suitable metric structure on this space, we might be able to devise iterative ways to find solutions by analogy with the method of hill climbing or gradient ascent; that is, when any experiment seems more or less successful than another, then we exploit this metrical structure to help us make the next move in the proper direction. (Later, I emphasize that having a sense of direction entails a little more than a sense of proximity: It is not enough just to know metrical distances; we must also respond to other kinds of heuristic differences, and these differences might be difficult to detect.)

Whenever we design or select a particular representation, this particular choice will bias our dispositions about which objects to consider more or less similar to us (or to the programs we apply to them) and, thus, will affect how we apply our knowledge to achieve goals and solve problems. Once we understand the effects of such commitments, we will be better prepared to select and modify these representations to produce more heuristically useful distinctions and confusions. Thus, let us now examine, from this point of view, some of the representations that have become popular in the AI field.

Heuristic Connections of Pure Logic

Why have logic-based formalisms been so widely used in AI research? I see two motives for selecting this type of representation. One virtue of logic is clarity, its lack of ambiguity. Another advantage is the preexistence of many technical mathematical theories about logic. But logic also has its disadvantages. Logical generalizations only apply to their literal lexical instances, and logical implications only apply to expressions that precisely instantiate their antecedent conditions. No exceptions are allowed, no matter how closely they match. This approach permits you to use no near misses, no suggestive clues, no compromises, no analogies, and no metaphors. To shackle yourself so inflexibly is to shoot your own mind in the foot—if you know what I mean.

These limitations of logic begin at the foundation with the basic connectives and quantifiers. The trouble is that worldly statements of the form "For all *x, P(x)*" are never beyond

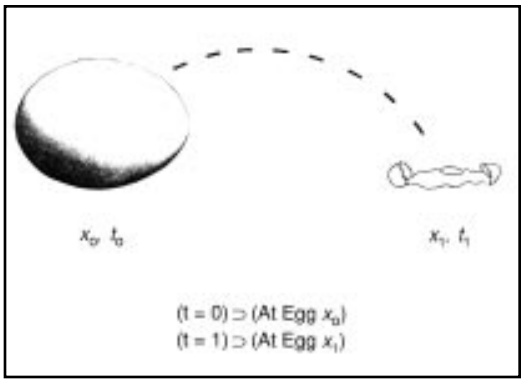


Figure 5. Default assumption

suspicion. To be sure, such a statement can indeed be universally valid inside a mathematical realm; however, this validity is because such realms are themselves based on expressions of this kind. The use of such formalisms in AI has led most researchers to seek universal validity, to the virtual exclusion of practicality or interest, as though nothing would do except certainty. Now, this approach is acceptable in mathematics (wherein we ourselves define the worlds in which we solve problems), but when it comes to reality, there is little advantage in demanding inferential perfection when there is no guarantee that even our assumptions will always be correct. Logic theorists seem to have forgotten that any expression in actual life—that is, in a world that we find but don't make—such as *(x)(Px)* must be seen as only a convenient abbreviation for something more like the following: "For any thing *x* being considered in the current context, the assertion *P{x}* is likely to be useful for achieving goals like *G*, provided that we apply it in conjunction with other heuristically appropriate inference methods." In other words, we cannot ask our problem-solving systems to be absolutely perfect or even consistent; we can only hope that they will grow increasingly better than blind search at generating, justifying, supporting, rejecting, modifying, and developing evidence for new hypotheses.

It has become particularly popular in AI logic programming to restrict the representation to expressions written in first-order predicate calculus. This practice, which is so pervasive that most students engaged in it don't even know what "first order" means here, facilitates the use of certain types of inference but at a high price: The predicates of such expressions are prohibited from referring in certain ways to one another. This

…I try to regard conflicts as opportunities rather than obstacles…

restriction prevents the representation of metaknowledge, rendering these systems incapable, for example, of describing what the knowledge that they contain can be used for. In effect, it precludes the use of functional descriptions. We need to develop systems for logic that can reason about their own knowledge and make heuristic adaptations and interpretations of it by using knowledge about this knowledge; however, the aforementioned limitations of expressiveness make logic unsuitable for such purposes.

Furthermore, it must be obvious that to apply our knowledge to commonsense problems, we need to be able to recognize which pairs of expressions are similar in whatever heuristic sense may be appropriate. But this seems too technically hard to do—at least for the most commonly used logical formalisms—namely, expressions in which absolute quantifiers range over stringlike normal forms. For example, to use the popular method of resolution theorem proving, one usually ends up using expressions that consist of logical disjunctions of separate, almost meaningless conjunctions. Consequently, the natural topology of any such representation will almost surely be heuristically irrelevant to any real-life problem space. Consider how dissimilar these three expressions seem when written in conjunctive form:

A ∨ B ∨ C ∨ D ,

AB ∨ AC ∨ AD ∨ BC ∨ BD ∨ CD ,

and

ABC ∨ ABD ∨ ACD ∨ BCD .

The simplest way to assess the distances or differences between expressions is to compare such superficial factors as the numbers of terms or subexpressions they have in common. Any such assessment would seem meaningless for expressions such as these. In most situations, however, it would almost surely be useful to recognize that these expressions are symmetric in their arguments and, hence, will clearly seem more similar if we rerepresent them—for example, by using S_n to mean n of S's arguments have truth value T—so that they can then be written in the form S_1, S_2 , and S_3. Even in mathematics itself, we consider it a great discovery to find a new representation for which the most natural-seeming heuristic connection can be recognized as close to the representation's surface structure. However, such a discovery is too much to expect in general, so it is usually necessary to gauge the similarity of two expressions by using more complex assessments based, for example, on the number of set-inclusion levels between them, or on the number of available operations required to transform one into the other, or on the basis of the partial ordering suggested by their lattice of common generalizations and instances. This means that making good similarity judgments might itself require the use of other heuristic kinds of knowledge, until eventually—that is, when our problems grow hard enough—we are forced to resort to techniques that exploit knowledge that is not so transparently expressed in any such "mathematically elegant" formulation.

Indeed, we can think about much of AI research in terms of a tension between solving problems by searching for solutions inside a compact and well-defined problem space (which is feasible only for prototypes) versus using external systems (that exploit larger amounts of heuristic knowledge) to reduce the complexity of that inner search. Compound systems of this sort need retrieval machinery that can select and extract knowledge that is relevant to the problem at hand. Although it is not especially hard to write such programs, it cannot be done in first-order systems. In my view, this can best be achieved in systems that allow us to simultaneously use object-oriented structure-based descriptions and goal-oriented functional descriptions.

How can we make logic more expressive given that each fundamental quantifier and connective is defined so narrowly from the start? This deficiency could well be beyond repair, and the most satisfactory replacement might be some sort of object-oriented frame-based language. After all, once we leave the domain of abstract mathematics and free ourselves from these rigid notations, we can see that some virtues of logiclike reasoning might remain, for example, in the sorts of deductive chaining we used and the kinds of substitution procedures we applied to these expressions. The spirit of some of these formal techniques can then be approximated by other, less formal techniques of making chains (see chapter 18, Minsky [1987]). For example, the mechanisms of defaults and frame arrays could be used to approximate the formal effects of instantiating generalizations. When we use heuristic chaining, of course, we cannot assume absolute validity of the result; so, after each reasoning step, we

might have to look for more evidence. If we notice exceptions and disparities, then later we must return to each or else remember them as assumptions or problems to be justified or settled at some later time, all things that humans so often do.

Heuristic Connections of Rule-Based Systems

Although logical representations have been used in research, rule-based representations have been more successful in applications. In these systems, each fragment of knowledge is represented by an *if-then* rule, so that whenever a description of the current problem situation precisely matches the rule's antecedent *if* condition, the system performs the action described by this rule's *then* consequent. What if no antecedent condition applies? The answer is simple: The programmer adds another rule. It is this seeming modularity that made rule-based systems so attractive. You don't have to write complicated programs. Instead, whenever the system fails to perform or does something wrong, you simply add another rule. This approach usually works well at first, but whenever we try to move beyond the realm of toy problems and start to accumulate more and more rules, we usually get into trouble because each added rule is increasingly likely to interact in unexpected ways with the others. Then, what should we ask the program to do when no antecedent fits perfectly? We can equip the program to select the rule whose antecedent most closely describes the situation; again, we're back to "similar." To make any real-world application program resourceful, we must supplement its formal reasoning facilities with matching facilities that are heuristically appropriate for the problem domain it is working in.

What if several rules match equally well? Of course, we could choose the first on the list, choose one at random, or use some other superficial scheme—but why be so unimaginative? In *The Society of Mind*, I try to regard conflicts as opportunities rather than obstacles, as openings that we can use to exploit other kinds of knowledge. For example, section 3.2 of *The Society of Mind* (Minsky 1987) suggests invoking a *principle of noncompromise* to discard sets of rules with conflicting antecedents or consequents. The general idea is that whenever two fragments of knowledge disagree, it may be better to ignore them both and refer to some other, independent agency. In effect, this approach is managerial: One agency can engage some other body of expertise to help decide which rules to apply. For example, one might turn to case-based reasoning to ask which method worked best in similar previous situations.

Yet another approach would be to engage a mechanism for inventing a new rule by trying to combine elements of those rules that almost fit already. Section 8.2 of *The Society of Mind* (Minsky 1987) suggests using K-line representations for this purpose. To do so, we must be immersed in a society-of-agents framework in which each response to a situation involves activating not one but a variety of interacting processes. In such a system, all the agents activated by several rules can then be left to interact, if only momentarily (both with one another and with the input signals) to make a useful self-selection about which of the agents should remain active. This could be done by combining certain current connectionist concepts with other ideas about K-line mechanisms. But that can't be done until we learn how to design network architectures that can support new forms of management and supervision of developmental staging.

In any case, current rule-based systems are still too limited in ability to express "typical" knowledge. They need better default machinery. They deal with exceptions too passively; they need censors. They need better "ring-closing" mechanisms for retrieving knowledge (see Minsky [1987], section 19.10). Above all, we need better ways to connect them with other kinds of representations so that we can use them in problem-solving organizations that can exploit other kinds of models and search procedures.

Connectionist Networks

Up to this point, we have considered ways to overcome the deficiencies of symbolic sys-

…it is less important for agencies to cooperate than to exploit one another…

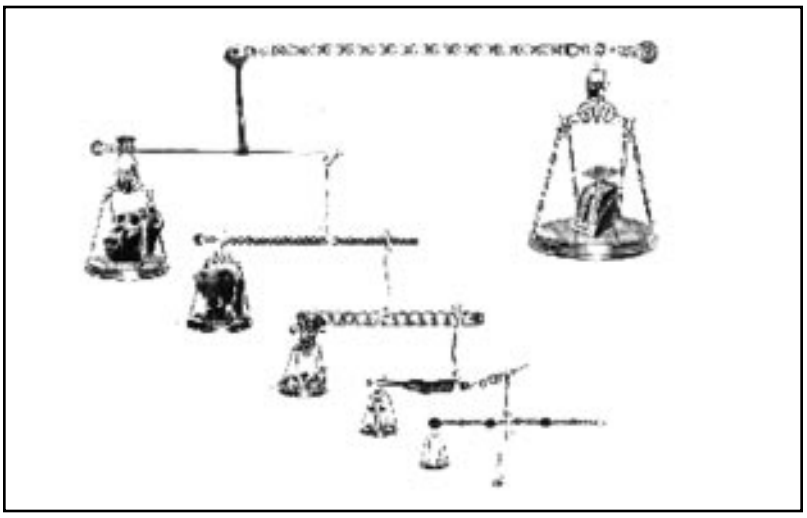


Figure 6. Weighty decisions

tems by augmenting them with connectionist machinery. However, this kind of research should go both ways. Connectionist systems have equally crippling limitations, which might be ameliorated by augmentation with the sorts of architectures developed for symbolic applications. Perhaps such extensions and synthesis will recapitulate some aspects of how the primate brain grew over millions of years by evolving symbolic systems to supervise its primitive connectionist learning mechanisms.

What do we mean by *connectionist*? The use of this term is still rapidly evolving, but here it refers to attempts to embody knowledge by assigning numeric conductivities or weights to the connections inside a network of nodes. The most common form of such a node is made by combining an analog, nearly linear part that "adds up evidence" with a nonlinear, nearly digital part that makes a decision based on a threshold. The most popular such networks today take the form of *multilayer perceptrons*, that is, sequences of layers of such nodes, each layer sending signals to the next. More complex arrangements are also under study that can support cyclic internal activities; hence, they are potentially more versatile but harder to understand. What makes such architectures attractive? Mainly, they appear to be so simple and homogeneous. At least on the surface, they can be seen as ways to represent knowledge without any complex syntax. The entire configuration state of such a net can be described as nothing more than a simple vector—and the network's input-output characteristics as nothing more than a map from one vector space into another. This arrangement makes it easy to reformulate pattern recognition and learning problems in simple terms, for example, of finding the best such mapping. Seen in this way, the subject presents a pleasing mathematical simplicity. It is often not mentioned that we still possess little theoretical understanding of the computational complexity of finding such mappings, that is, of how to discover good values for the connection weights. Most current publications still merely exhibit successful small-scale examples without probing into either assessing the computational difficulty of these problems themselves or of scaling these results to similar problems of larger size.

However, we now know of many situations in which even such simple systems have been made to compute (and, more important, to learn to compute) interesting functions, particularly in such domains as clustering, classification, and pattern recognition. In some instances, this has occurred without any external supervision; furthermore, some of these systems have also performed acceptably in the presence of incomplete or noisy input and, thus, correctly recognized patterns that were novel or incomplete. This achievement means that the architectures of those systems must indeed have embodied heuristic connectivities that were appropriate for those particular problem domains. In such situations, these networks can be useful for the kind of reconstruction-retrieval operations we call *ring closing*.

However, connectionist networks have limitations as well. The next few sections discuss some of these limitations along with suggestions on how to overcome them by embedding such networks in more advanced architectural schemes.

Limitation of Fragmentation: The Parallel Paradox

In the Epilogue to *Perceptrons*, Papert and I argued as follows:

> It is often argued that the use of distributed representations enables a system to exploit the advantages of parallel processing. But what are the advantages of parallel processing? Suppose that a certain task involves two unrelated parts. To deal with both concurrently, we would have to maintain their representations in two decoupled agencies, both active at

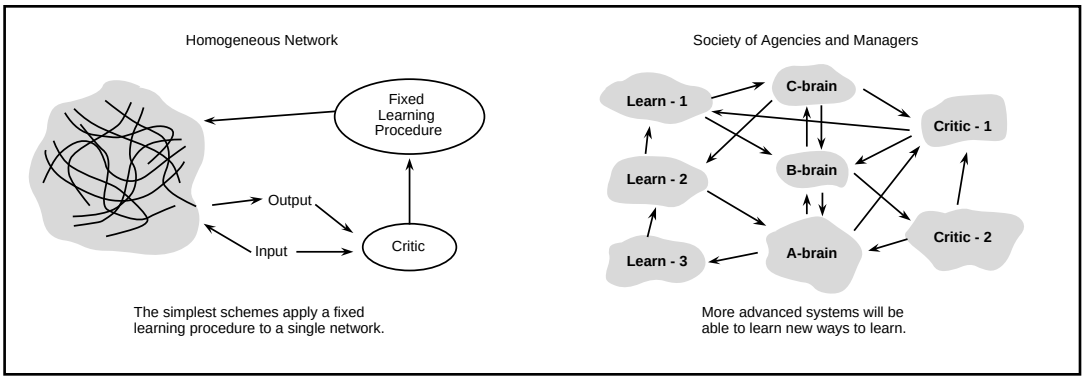


Figure 7. Homostructural vs. heterostructural

the same time. Then, should either of those agencies become involved with two or more sub-tasks, we'd have to deal with each of them with no more than a quarter of the available resources! If that proceeded on and on, the system would become so fragmented that each job would end up with virtually no resources assigned to it. In this regard, distribution may oppose parallelism: the more distributed a system is—that is, the more intimately its parts interact—the fewer different things it can do at the same time. On the other side, the more we do separately in parallel, the less machinery can be assigned to each element of what we do, and that ultimately leads to increasing fragmentation and incompetence. This is not to say that distributed representations and parallel processing are always incompatible. When we simultaneously activate two distributed representations in the same network, they will be forced to interact. In favorable circumstances, those interactions can lead to useful parallel computations, such as the satisfaction of simultaneous constraints. But that will not happen in general; it will occur only when the representations happen to mesh in suitably fortunate ways. Such problems will be especially serious when we try to train distributed systems to deal with problems that require any sort of structural analysis in which the system must represent relationships between substructures of related types—that is, problems that are likely to demand the same structural resources. (Minsky and Papert 1988, p. 277)

(See also Minsky [1987], section 15.11.)

For these reasons, it will always be hard for a homogeneous network to perform parallel *high-level* computations, unless we can arrange for it to become divided into effectively disconnected parts. There is no general remedy for this problem, and it is no special peculiarity of connectionist hardware; computers have similar limitations, and the only answer is providing more hardware. More generally, it seems obvious that without adequate memory buffering, homogeneous networks must remain incapable of recursion, as long as successive function calls have to use the same hardware. This inability is because without such facilities, either the different calls will cause side effects for one another, or some of them must be erased, leaving the system unable to execute proper returns or continuations. Again, this might easily be fixed by providing enough short-term memory, for example, in the form of a stack of temporary K-lines.

Limitations of Specialization and Efficiency

Each connectionist net, once trained, can only do what it has learned to do. To make it do something else—for example, to compute a different measure of similarity or to recognize a different class of patterns—would, in general, require a complete change in the matrix of connection coefficients. Usually, we can change the function of a computer much more easily (at least, when the desired functions can each be computed by compact algorithms) because a computer's memory cells are so much more interchangeable. It is curious how even technically well-informed people tend to forget how computationally massive a fully connected neural network is. It is instructive to compare its size with the

few hundred rules that drive a typically successful commercial rule-based expert system.

How connected do networks need to be? Several points in *The Society of Mind* suggest that commonsense reasoning systems might not need to increase the density of physical connectivity as fast as they increase the complexity and scope of their performances. Chapter 6 (Minsky 1987) argues that knowledge systems must evolve into clumps of specialized agencies, rather than homogeneous networks, because they develop different types of internal representations. As this evolution proceeds, it will become decreasingly feasible for any of these agencies directly to communicate with the interior of others. Furthermore, there will be a tendency for most newly acquired skills to develop from the relatively few that are already well developed, which again will bias the largest-scale connections toward evolving into recursively clumped, rather than uniformly connected, arrangements. A different tendency to limit connectivities is discussed in section 20.8, which proposes a sparse connection scheme that can simulate in real time the behavior of fully connected nets—in which only a small proportion of agents are simultaneously active. This method, based on a half-century–old idea of Calvin Mooers, allows many intermittently active agents to share the same relatively narrow, common connection bus. This might seem, at first, a mere economy, but section 20.9 suggests that this technique could also induce a more heuristically useful tendency if the separate signals on that bus were to represent meaningful symbols. Finally, chapter 17 suggests other developmental reasons why minds might virtually be forced to grow in relatively discrete stages rather than as homogeneous networks. Our progress in making theories about this area might parallel our progress in understanding the stages we see in the growth of every child's thought.

If our minds are assembled of agencies with so little intercommunication, how can those parts cooperate? What keeps them working on related aspects of the same problem? The first answer I propose in *The Society of Mind* is that it is less important for agencies to cooperate than to exploit one another because those agencies tend to become specialized, developing their own internal languages and representations. Consequently, they cannot understand each other's internal operations well—and each must learn to exploit some of the others for the effects that those others produce—without knowing in any detail how these other effects are produced. Similarly, there must be other agencies to manage all these specialists, to keep the system from too much fruitless conflict for access to limited resources. These management agencies cannot directly deal with all the small interior details of what happens inside their subordinates. Instead, they must work with summaries of what those subordinates seem to do. This also suggests that there must be constraints on internal connectivity: Too much detailed information would overwhelm those managers. Such constraints also apply recursively to the insides of every large agency. Thus, in chapter 8 of *The Society of Mind* (Minsky 1987), I argue that relatively few direct connections are needed except between adjacent level bands.

All this suggests (but does not prove) that large commonsense reasoning systems will not need to be fully connected. Instead, the system could consist of localized clumps of expertise. At the lowest levels, these clumps would have to be densely connected to support the sort of associativity required to learn low-level pattern-detecting agents. However, as we ascend to higher levels, the individual signals must become increasingly abstract and significant, and accordingly, the density of connection paths between agencies can become increasingly (but only relatively) smaller. Eventually, we should be able to build a sound technical theory about the connection densities required for commonsense thinking, but I don't think that we have the right foundations yet. The problem is that contemporary theories of computational complexity are still based too much on worst-case analyses or coarse statistical assumptions, neither of which suitably represents realistic heuristic conditions. The worst-case theories unduly emphasize the intractable versions of problems that, in their usual forms, present less practical difficulty. The statistical theories tend to uniformly weight all instances for lack of systematic ways to emphasize the types of situations of most practical interest. However, the AI systems of the future, like their human counterparts, will normally prefer to satisfice rather than optimize—and we don't yet have theories that can realistically portray these mundane sorts of requirements.

Limitations of Context, Segmentation, and Parsing

When we see seemingly successful demonstrations of machine learning in carefully prepared test situations, we must be careful about how we draw more general conclu-

sions. This is because there is a large step between the abilities to recognize objects or patterns when they are isolated and when they appear as components of more complex scenes. In section 6.6 of *Perceptrons* (Minsky and Papert 1988), we see that we must be prepared to find that even after training a certain network to recognize a certain type of pattern, we might find it unable to recognize this same pattern when embedded in a more complicated context or environment. (Some reviewers have objected that our proofs of this fact applied only to simple three-layer networks; however, most of these theorems are much more general, as these critics might see if they'd take the time to extend those proofs.) The problem is that it is usually easy to make isolated recognitions by detecting the presence of various features and then computing weighted conjunctions of them. This is easy to do in three-layer acyclic nets. But in compound scenes, this method won't work unless the separate features of all the distinct objects are somehow properly assigned to the correct objects. Similarly, we cannot expect neural networks generally to be able to parse the treelike or embedded structures found in the phrase structure of natural language.

How could we augment connectionist networks to make them able to do such things as analyze complex visual scenes or extract and assign the referents of linguistic expressions to the appropriate contents of short-term memories? It will surely need additional architecture to represent the structural analysis of a visual scene into objects and their relationships, for example, by protecting each midlevel recognizer from seeing input derived from other objects, perhaps by arranging for the object-recognizing agents to compete in assigning each feature to itself, but denying it to competitors. This method has been successfully used in symbolic systems, and parts have been done in connectionist systems (for example, by Waltz and Pollack), but many conceptual missing links remain in this area, particularly in regard to how a second connectionist system could use the output of one that managed to parse the scene. In any case, we should not expect to see simple solutions to these problems. It might be an accident that so much of the brain is occupied with such functions.

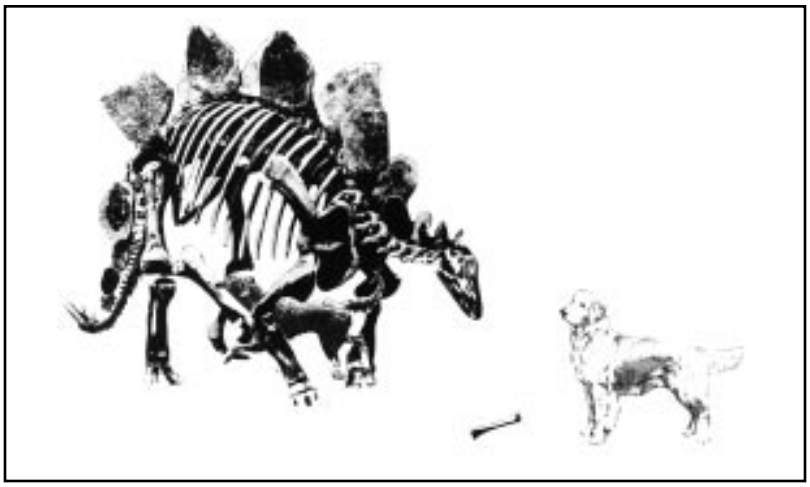


Figure 8. Recognition in context

Limitations of Opacity

Most serious of all is what we might call the problem of *opacity*, that the knowledge embodied inside a network's numeric coefficients is not accessible outside that net. This challenge is not one we should expect our connectionists to easily solve. I suspect it is so intractable that even our own brains have

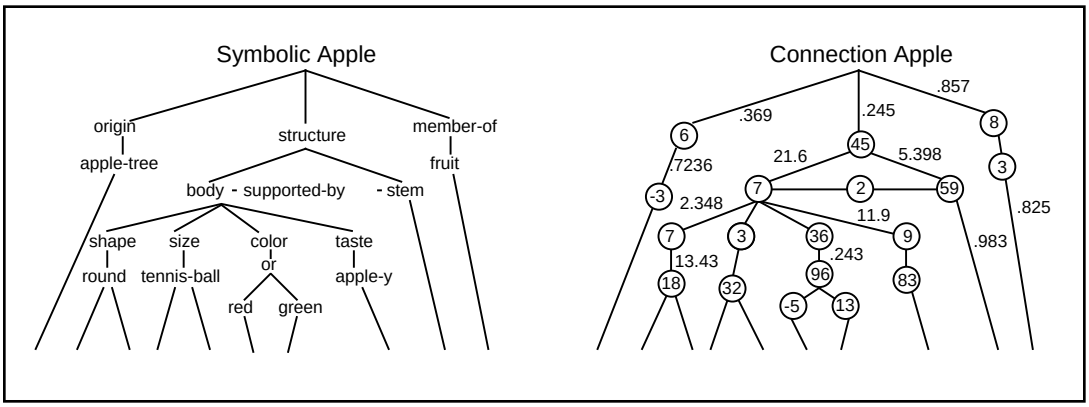


Figure 9. Numerical opacity

The future work of mind design will not be much like what we do today.

evolved little such capacity over the billions of years it took to evolve from anemonelike reticulae. Instead, I suspect that our societies and hierarchies of subsystems have evolved ways to evade the problem by arranging for some of our systems to learn to model what some of our other systems do (see Minsky [1987], section 6.12). They might do this modeling in part by using information obtained from direct channels into the interiors of these other networks, but mostly, I suspect, they do it less directly—so to speak, behavioristically—by making generalizations based on external observations, as though they were like miniature scientists. In effect, some of our agents invent models of others. Regardless of whether these models might be defective or even entirely wrong (and here I refrain from directing my aim at peculiarly faulty philosophers), it suffices for these models to be useful in enough situations. To be sure, it might be feasible, in principle, for an external system to accurately model a connectionist network from outside by formulating and testing hypotheses about its internal structure. However, of what use would such a model be if it merely repeated redundantly? It would not only be simpler but also more useful for that higher-level agency to assemble only a pragmatic, heuristic model of this other network's activity based on concepts already available to that observer. (This is evidently the situation in human psychology. The apparent insights we gain from meditation and other forms of self-examination are only infrequently genuine.)

The problem of opacity grows more acute as representations become more distributed—that is, as we move from symbolic to connectionist poles—and it becomes increasingly more difficult for external systems to analyze and reason about the delocalized ingredients of the knowledge inside distributed representations. It also makes it harder to learn, past a certain degree of complexity, because it is hard to assign credit for success, or formulate new hypotheses (because the old hypotheses themselves are not "formulated"). Thus, distributed learning ultimately limits growth, no matter how convenient it might be in the short term, because "the idea of a thing with no parts provides nothing that we can use as pieces of explanation" (Minsky 1987).

For such reasons, although homogeneous, distributed learning systems may work well to a certain point, they eventually may tend to fail when confronted with problems of larger scale—unless we find ways to compensate the accumulation of many weak connections with some opposing mechanism that favors internal simplification and localization. Many connectionist writers positively seem to rejoice in the holistic opacity of representations within which even they are unable to discern the significant parts and relationships. However, unless a distributed system has enough ability to crystallize its knowledge into lucid representations of its new subconcepts and substructures, its ability to learn will eventually slow, and it will be unable to solve problems beyond a certain degree of complexity. In addition, although this situation suggests that homogeneous network architectures might not work well past a certain size, this restriction should be bad news only for those ideologically committed to minimal architectures. For all we currently know, the scales at which such systems crash are large enough for our purposes. Indeed, the society of mind thesis holds that most of the agents that grow in our brains only need to operate on scales so small that each by itself seems no more than a toy. But when we combine enough of them—in ways that are not too delocalized—we can make them do almost anything.

In any case, we should not assume that we always can—or always should—avoid the use of opaque schemes. The circumstances of daily life compel us to make decisions based on "adding up the evidence." We frequently find (when we value our time) that even if we had the means, it wouldn't pay to analyze. The society of mind theory of human thinking doesn't suggest otherwise; on the contrary, it leads us to expect to encounter incomprehensible representations at every level of the mind. A typical agent does little more than exploit other agents' abilities; hence, most of our agents accomplish their job knowing virtually nothing of how it is done.

Analogous issues of opacity arise in the symbolic domain. Just as networks sometimes solve problems by using massive combinations of elements, each of which has little individual significance, symbolic systems sometimes solve problems by manipulating large expressions with similarly insignificant terms, such as when we replace the explicit structure of a composite Boolean function with a locally senseless canonical form. Although this technique simplifies some computations by making them more homogeneous, it disperses knowledge about the structure and composition of the data and, thus, disables our ability to solve harder problems. At both extremes—in representations that are either too distributed or too discrete—we lose the structural knowledge embodied in the form of intermediate-level concepts. This loss might not be evident as long as our problems are easy to solve, but those intermediate concepts might be indispensable for solving more advanced problems. Comprehending complex situations usually hinges on discovering a good analogy or variation on a theme. However, it is virtually impossible to do this with a representation, such as a logical form, a linear sum, or a holographic transformation—each of whose elements seem meaningless because they are either too large or too small—thus leaving no way to represent significant parts and relationships.

Many other problems invite the synthesis of symbolic and connectionist architectures. How can we find ways for nodes to refer to other nodes or to represent knowledge about the roles of particular coefficients? To see the difficulty, imagine trying to represent the structure of the arch in Patrick Winston's thesis—without simply reproducing its topology. Another critical issue is how to enable nets to make comparisons. This problem is more serious than it might seem. Section 23.1 of *The Society of Mind* discusses the importance of differences and goals, and section 23.2 points out that connectionist networks deficient in short term memory will find it peculiarly difficult to detect differences between patterns (Minsky 1987). Networks with weak architectures will also find it difficult to detect or represent (invariant) abstractions; this problem was discussed as early as the Pitts-McCulloch paper of 1947. Still another important problem for memory-weak, bottom-up mechanisms is controlling search: To solve hard problems, one might have to consider different alternatives, explore their subalternatives, and then make comparisons among them—yet still be able to return to the initial situation without forgetting what was accomplished. This kind of activity, which we call thinking, requires facilities for temporarily storing partial states of the system without confusing these memories. One answer is to provide, along with the required memory, some systems for learning and executing control scripts, as suggested in section 13.5 of *The Society of Mind* (Minsky 1987). To do this effectively, we must have some *insulationism* to counterbalance our *connectionism*. Smart systems need both of these components, so the symbolic-connectionist antagonism is not a valid technical issue but only a transient concern in contemporary scientific politics.

Mind Sculpture

The future work of mind design will not be much like what we do today. Some programmers will continue to use traditional languages and processes. Other programmers will turn toward new kinds of knowledge-based expert systems. However, eventually all this work will be incorporated into systems that exploit two new kinds of resources. On one side, we will use huge preprogrammed reservoirs of commonsense knowledge. On the other side, we will have powerful, modular learning machines equipped with no knowledge at all. Then, what we know as programming will change its character entirely—to an activity that I envision to be more like sculpturing. To program today, we must describe things very carefully because nowhere is there any margin for error. But once we have modules that know how to learn, we won't have to specify nearly so much—and we'll program on a grander scale, relying on learning to fill in details.

This doesn't mean, I hasten to add, that things will be simpler than they are now. Instead, we'll make our projects more ambitious. Designing an artificial mind will be much like evolving an animal. Imagine yourself at a terminal, assembling various parts of a brain. You'll be specifying the sorts of things that were only heretofore described in texts about neuroanatomy. "Here," you'll find yourself thinking, "we'll need two similar networks that can learn to shift time signals into spatial patterns so that they can be compared by a feature extractor sensitive to a context about this wide." Then, you'll have to sketch the architectures of organs that can learn to supply appropriate input to those agencies and draft the outlines of intermediate organs for learning to suitably encode the output to suit the needs of other agencies. Section 31.3 of *The Society of Mind* (Minsky 1987) suggests how a genetic system might mold the form of an agency that is predes-

…as in any society, there must be watchers to watch each watcher, lest any one or a few of them get too much control of the rest.

tined to learn to recognize the presence of particular human individuals. A functional sketch of such a design might turn out to involve dozens of different sorts of organs, centers, layers, and pathways. The human brain might have many thousands of such components.

A functional sketch is only the start. Whenever you use a learning machine, you must specify more than just the sources of input and the destinations of output. It must also somehow be impelled toward the sorts of things you want it to learn—what sorts of hypotheses it should make, how it should compare alternatives, how many examples should be required, how to decide when enough has been done, when to decide that things have gone wrong, and how to deal with bugs and exceptions. It is all very well for theorists to speak about "spontaneous learning and generalization," but there are too many contingencies in real life for such words to mean anything by themselves. Should this agency be an adventurous risk taker or a careful, conservative reductionist? One person's intelligence is another's stupidity. And how should that learning machine divide and budget its resources of hardware, time, and memory?

How will we build such grand machines when so many design constraints are involved? No one will be able to track all the details because just as a human brain is constituted by interconnecting hundreds of different kinds of highly evolved subarchitectures, so will these new kinds of thinking machines. Each new design will have to be assembled by using libraries of already developed, off-the-shelf subsystems already known to be able to handle particular kinds of representations and processes. Also, the designer will be less concerned with what happens inside these units and more concerned with their interconnections and interrelationships. Because most components will be learning machines, the designer will have to specify not only what each one will learn but also which agencies should provide what incentives and rewards for which others. Every such decision about one agency imposes additional constraints and requirements on several others and, in turn, on how to train those others. In addition, as in any society, there must be watchers to watch each watcher, lest any one or a few of them get too much control of the rest.

Each agency will need nerve bundle–like connections to certain other ones for sending and receiving signals about representations, goals, and constraints, and we'll have to make decisions about the relative size and influence of every such parameter. Consequently, I expect that the future art of brain design will have to be more like sculpturing than like our current craft of programming. It will be much less concerned with the algorithmic details of the submachines than with the balancing of their relationships; perhaps this situation better resembles politics, sociology, or management than present-day engineering.

Some neural network advocates might hope that all this will be superfluous. Why not seek to find, instead, how to build one single, huge net that can learn to do all these things by itself. Did not our own human brains come about as the outcome of one great learning search? We could only regard this as feasible by ignoring the facts—the unthinkable scale of that billion-year venture and the octillions of lives of our ancestors. Remember, too, that even so, in all that evolutionary search, not all the problems have yet been solved. What will we do when our sculptures don't work? Consider a few of the wonderful bugs that still afflict even our own grand human brains: *obsessive preoccupation with inappropriate goals; inattention and inability to concentrate; bad representations; excessively broad or narrow generalizations; excessive accumulation of useless information; superstition: defective credit-assignment schema; unrealistic cost-benefit analyses; unbalanced, fanatical search strategies; formation of defective categorizations; inability to deal with exceptions to rules; improper staging of development, or living in the past; unwillingness to acknowledge loss; depression or maniacal optimism; and excessive confusion from cross-coupling.*

Seeing this list, one has to wonder, "Can people think?" I suspect there is no simple and magical way to avoid such problems in our new machines; it will require a great deal of research and engineering. I suspect that it is no accident that human brains contain so many different and specialized brain centers. To suppress the emergence of serious bugs, both those natural systems and the artificial ones we shall construct will probably need intricate arrangements of interlocking checks and balances, in which each agency is supervised by several others. Furthermore, each of these other agencies must themselves learn when and how to use the resources available to them. How, for example, should each learning system balance the advantages of immediate gain over those of conservative, long-term growth? When should it favor the accumulation of competence over comprehension? In the large-scale design of our human brains, we still don't know much of what all the different organs do, but I'm willing to bet that many of them are largely involved in regulating others, to keep the system as a whole from falling prey to the sorts of bugs that were mentioned above. Until we start building brains ourselves to learn what bugs are most probable, it will remain hard for us to guess the actual functions of much of that hardware.

There are countless wonders yet to be discovered in these exciting new fields of research. We can still learn a great many things from experiments on even the simplest nets. We'll learn even more from trying to make theories about what we observe. And surely, soon we'll start to prepare for that future art-of-mind design by experimenting with societies of nets that embody more structured strategies—and, consequently, make more progress on the networks that make up our own human minds. And in doing all these experiments, we'll discover how to make symbolic representations that are more adaptable and connectionist representations that are more expressive.

It is amusing how persistently people express the view that machines based on symbolic representations (as opposed, presumably, to connectionist representations) could never achieve much or ever be conscious and self-aware. I maintain it is precisely because our brains are still mostly connectionist, that we humans have so little consciousness! It's also why we're capable of so little parallelism of thought—and why we have such limited insight into the nature of our own machinery.

Acknowledgment

This research was funded over a period of years by the Computer Science Division of the Office of Naval Research. Illustrations by Juliana Minsky.

Notes

1. Adapted from Logical versus Analogical or Symbolic versus Connectionist or Neat versus Scruffy. In *AI at MIT: Expanding Frontiers*, eds. Patrick H. Winston, with S. A. Shellard, 219–243. Cambridge, Mass.: MIT Press.

Bibliography

Minsky, M. 1988. Preface. In *Connectionist Models and Their Implications: Readings from Cognitive Science*, eds. D. L. Waltz and J. Feldman, vii–xvi. Norwood, N.J.: Ablex.

Minsky, M. 1987. *The Society of Mind*. New York: Simon and Schuster.

Minsky, M. 1974. A Framework for Representing Knowledge, Report AIM, 306, Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Reprinted in *The Psychology of Computer Vision*, ed. P. H. Winston, 211–277. New York: McGraw-Hill.

Minsky, M., and Papert, S. 1988. *Perceptrons,* 2d ed. Cambridge, Mass.: MIT Press.

Marvin Minsky's work in machine cognition has not only involved heuristic programming, cognitive psychology, and connectionist learning networks but also the mathematical foundations of computer science and the practical technology of mechanical robotics. He has contributed to the domains of symbolic description, knowledge representation, symbolic applied mathematics, computational semantics, and machine perception. His main concern over the past decade has been to discover how to make machines do commonsense reasoning. The foundations of his new conception of human psychology are described in his book *The Society of Mind*, which proposes revolutionary concepts about thinking, learning, memory, language, and conceptual development as well as the administrative structures that underlie the functions of the brain. Minsky is a 1990 recipient of the prestigious Japan Prize that recognizes original and outstanding achievements in science and technology.