CAN A MACHINE THINK?

Guest Letter from Clint Kelly, Friend of the National Zoo
February 1998

In one variant or another, the question, "can a machine think," has occupied the attention of philosophers and others for centuries, stimulated from time-to-time by the emergence of ingenious mechanisms which suggested at least the possibility of an affirmative answer.

Machines Which (Who?) Play Chess

Such a mechanism caught the public's attention 1809. The Emperor Napoleon I, a chess player certainly of prominence and reportedly of ability, lost a chess game in 15 moves to a seeming thinking machine; a clockwork automaton known as "the Turk." Reports of the time say that Napoleon "angrily stalked from the room." The Turk, named for one of its component parts, a mannequin dressed in elegant Turkish attire, was built for the Empress Maria Theresa in 1769 by the Baron von Kempelen, a Hungarian engineer. The machine consisted of the mannequin, whose mechanical arm moved the chess pieces, and a cabinet at which the mannequin sat. Inside the cabinet was a shining, brass clockwork mechanism. This mechanism was supposed to be responsible for deciding the automaton's moves and for positioning the mannequin's arm. The cabinet was opened for inspection before each game to convince the audience that the mechanism was all that it contained.

The Turk did not win every game but it won enough to establish a reputation as a player of rank. Its success gave rise to much speculation about how a cabinet full of gears could successfully compete with well-known human chess masters. How could this clockwork mechanism think? In fact, it couldn't. The Turk was a magician's illusion; the cabinet cleverly designed to conceal a human chess player.

Now, fast-forward some 188 years to 11 May 1997. Another prominent chess player, grandmaster and world champion Gary Kasparov has lost the sixth game and the match to a machine; but this automaton was no illusion. A creation of silicon and software (not brass gears), special purpose computer hardware, and clever instructions encoding the insights of skilled players, IBM's Deep Blue was the first machine to defeat a world champion. Within the limits imposed by the game, this machine seemed to think. Observers commented on its style of play almost as if they regarded it as human.

For many people, the outcome of this chess match between human and machine signaled the beginning of a major change in the way we view ourselves and our place in the universe. Chess has long been regarded as the most cerebral of games. The ability to play chess well has always been considered a hallmark of human logic and reasoning.

Now we have a machine that solves a problem that we have always solved with human intelligence. Suddenly, there appears to be a whole range of activities which had been the sole province of humans that is now open to computers. If a computer can unseat the world's best chess player, some wondered "How long will it be before I lose my seat down at the office?"

So Deep Blue plays a mean game of chess. What else can it do? Essentially nothing; but there are a very large number (tens of thousands?) of less well-known computer programs from the field of artificial intelligence (AI) with performance equally impressive in their areas as Deep Blue's. Each of these, like Deep Blue, captures in some limited way elements of intelligent behavior. There are programs that: play Chess, Backgammon, Go, and Bridge; solve word problems; detect credit card fraud; design jet engines and computer systems; pick stocks; troubleshoot machinery; find information on the World Wide Web; target advertising; screen loan applicants; monitor compliance of Bosnian combatants with arms restrictions; predict chemical reactions; make medical diagnoses; reduce emissions from an electric generating plant; daily schedule 100s of telephone repair personnel; screen pap smears; simulate production from an oil reservoir; conduct logistics planning for Operation Desert storm; control a process to make soup; detect violations of The Nuclear Test Ban Treaty; read handwriting; write poems; compose music; help explore Mars; paint a picture; prove mathematical theorems; and , in 1995, drove a car autonomously from Washington, D.C. to San Diego, California.

It would appear that some of those "seats down at the office" are indeed now occupied by computers. Yet somehow, none of these AI applications, impressive and useful as they are, would be called "intelligent' in the general, human sense. They do beautifully in their niches, but our common sense concept of intelligence requires something more.

<u>How Do We Know If We Have It?</u>

But how do we measure intelligence? This is a remarkably difficult issue. Common sense notions of intelligence are based on observations of behavior with the belief that the more complex the behavior, in some sense, the more intelligent the animal. The most complex human behavior is the use of language. Animals communicate, and studies of animal intelligence with data on such animals as: mantis shrimp, lobsters, horseshoe crabs, octopi, and sea anemones (as well as the more traditional studies of rats, mice, birds and of course, primates) show that they also exhibit other complex behavior. This research illustrates that whatever intelligence is, it exists to some degree across a wide range of organisms. But there is a qualitative difference between the ability of humans and that of animals to communicate. A clear and major difference exists; animals do not possess anything remotely like human language skills. The test for intelligence devised in 1950 by the British mathematician Alan M. Turing focuses on behavior which requires language skills.

The essence of the Turing test is a conversation, via a teletype, <u>on any topic whatsoever</u>, between a person, a computer, and a judge (a human). The judge's goal is to decide which of the respondents is human. If a computer, said Turing, could answer so as to convince the judge that it is a person (not a computer) then for all practical purposes the computer could be said to "think." More recently, the test is thought of as perhaps using speech instead of a teletype, and perhaps including some requirements for image understanding. No AI program has come close to passing the Turing test.

The philosopher John Searle of the University of California, Berkeley, has raised one of the most interesting and contentious philosophical arguments against the "sufficiency" of the Turing test and indeed against artificial intelligence. Searle asks us to imagine that he sits in a room with a slot in the door through which come slips of paper with questions written in Chinese characters. Searle does not understand Chinese but he has in the room with him a code book of instructions in English, which tell him how to develop answers. He follows the instructions, prepares the answer in Chinese, and pushes it through the slot. The answer makes sense to the Chinese speakers outside. Now to his outside observer, the "room" appears to understand Chinese. Searle, however, was just following formal rules and was completely ignorant of the meaning of either the questions or the answers. He concludes, since he knows he does not understand Chinese, that mere symbol manipulation, although producing the appearance of intelligence to the outside observer, cannot produce understanding or awareness in the mechanism doing the manipulation, in this instance Searle himself. What Searle is saying is that if he does not understand Chinese solely on the basis of running a computer program for understanding Chinese (the instructions in the code book), then neither does any digital computer. Digital computers says Searle "merely manipulate formal symbols according to rules in the program." He continues: "What goes for Chinese goes for other forms of cognition as well. Just manipulating the symbols is not by itself enough to guarantee consciousness, cognition, perception, understanding, thinking, and so forth."

Searle also links intelligence and consciousness, suggesting that conscious intentionality is the essence of intelligent behavior. Marvin Minsky of MIT, one of the founders of AI, believes that consciousness, specifically emotion, is critical for setting and changing goals; clearly an important part of intelligent behavior. Others have no difficulty in separating intelligence from consciousness. One author says "it's a lot easier to imagine the possibility of an intelligent computer than it is to imagine the possibility of a conscious computer or a computer with a free will."

Why Do We Believe We Can Construct A "Thinking Machine?" – What A Computer Can And Cannot Do

We now know that we can make computers excel on limited problems such as recognizing speech (if it is grammatical and carefully pronounced), scheduling a factory, recognizing a particular object in a scene, designing a jet engine, or even performing a complex medical diagnosis. But we are very far from creating a computer which can pass an unrestricted Turing Test. As one writer put it, "computers have mastered intellectual tasks, such as chess and integral calculus, but they have yet to attain the skills of a lobster in dealing with the real world." Given the gap between these niche capabilities and the requirements of the unrestricted Turing Test, why do we think computers may have the “right stuff?” The reasons are among some of the most significant philosophical concepts of the late 20th century.

The philosophy which dominated thinking about the mind for almost three centuries is called Cartesian dualism; the position first set forth by the French mathematician and philosopher Rene Descartes in the early 1600s, that there are two kinds of substances in the world: mental and physical or immaterial “mind stuff” apart from material substance. If we held this belief today, there would be little reason to suppose we could make much progress creating intelligence using a computer. Today, most philosophers instead argue that the mind (and intelligence) is an emergent property of material processes at the micro-level. This suggests that if we simulate the brain at the right level of detail, mind and intelligence may also emerge from the simulation. The open issue is how far down in the structure do we have to go? Can we get, by simulating brain processes, at the higher “psychological” level or do we require lower-level neuro-physiological detail?

What we know of the brain suggests it is almost unimaginably complex. It contains approximately 100 billion neurons, of different types, densely interconnected with each neuron linked to 10s of thousands of others. By contrast, a snail has about 1,000,000 neurons, a bee about 600,000 and a laboratory rat about 65 million. But a brain is much more than neurons. The neurons, and other types of cells, are immersed in a complicated chemistry which they influence and which affects them. The correct way to think of the brain is as a complex, dynamic, non-linear chemical system, not just as a network of neurons. Why do we think, even in principle, without for the moment considering the formidable practical issues, that we can simulate this? The answer requires examining the theoretical limits of computation; limits on what a computer can do.

The concept of computation was first formalized by Alan Turing in 1935, well before there were electronic computers. Turing’s goal was to formalize some intuitive concepts of methods for mathematical reasoning. To do this, he employed a mechanical metaphor which is called the Turing machine. It consists of an infinite tape, a sensing head for reading and writing symbols on the tape, and a control box with a finite number of internal states. In the control box is a table (the software program) which the machine uses to determine what action to take. For each possible state of the control box, and for each possible symbol being read by the sensing head, the table has an entry which tells the machine what symbol to print on the tape, in which direction to move the sensing head along the tape; and which state to enter next. So imagine the head scooting back and forth along the tape reading and writing symbols. Thought of this way, the Turing machine is simply a device for transforming one string of symbols into another string according to a predetermined set of rules; the table in the control box. The advantage of the Turing Machine is not as an actual device to do computation but to clarify operations masked in real computers. However, your personal computer (as well as the largest supercomputer) are Turing Machines at their core. The simplicity of the Turing Machine helps people establish theoretical limits on the ultimate problem-solving capabilities of real computers. One such result, called the Church-Turing thesis, is that if anything can be computed at all, it can be computed by a Turing Machine. No physical process is known to exist that can be used to build a device computationally more powerful than a Turing Machine. The Church-Turing Thesis is called a thesis, not a theorem, because it is not amenable to proof. Nevertheless, it is believed by most mathematicians; no evidence to the contrary has turned up.

There are things which a Turing machine cannot do. There are numbers which are uncomputable, numbers which a Turing machine cannot generate even by executing an infinite number of steps. Most people believe that uncomputability is not important in real-world processes. Thus, if we describe real-world processes (like the functioning of a neuron in a brain) by the appropriate equations, those equations can be solved/computed, by a Turing machine given sufficient time. So if you accept that the brain is a physical process, that there is no mysterious “mind stuff,” then in principle it can be simulated by a Turing machine. To restate the Church-Turing Thesis: the brain is a physical process, physical processes are computable, all computable processes can be computed by a Turing Machine (or any digital computer).

If you wish to simulate such features of intelligence/consciousness as playing chess or doing symbolic integration, then a relatively coarse level of simulation, a psychological-level, will likely suffice. If you wish to have creativity, emotional responses, an aesthetic sense, or even self-awareness, then a very fine-grained, neurophysiological-level simulation will likely be required, and the end may only be realized when we have totally duplicated a living brain; either as a simulation or in some, perhaps organic, type of “hardware.”

Not everyone agrees that simulation is the answer. John Searle argues that a simulation of a process is not that process. A simulation of an airplane does not fly; a simulation of the digestive process does not digest. Searle believes that consciousness emerges as a result of natural processes, but that simulation and computation cannot themselves create consciousness.

AI is based on faith that there are significant features of intelligence which can “be floated on top of entirely different sorts of substrates than those of organic brains.” This is a consequence of the Church-Turing thesis. The computing hardware doesn’t matter; silicon, Tinkertoys (MIT students built a Tic-Tac-Toe playing computer out of Tinkertoys), or living neurons are all computationally equivalent. But Searle claims the “hardware” does make a difference, and to achieve intelligence or consciousness we will have to replicate some of the organic processes themselves. Some counter by saying that a simulation of information processing is information processing. A simulation of two plus two still comes out four. Intelligence comes about through information processing, the argument goes, so a simulation of information processing can yield intelligence. That is, the simulation captures causality. It is difficult to confirm or refute Searle’s position on philosophical grounds. Ultimately, it and other positions will be decided empirically.

Another class of objections is raised by the British mathematician and physicist Roger Penrose. He bases this on his conviction that mathematicians can solve problems, which by a theorem proven by the mathematician Kurt Gödel, can have no guaranteed algorithmic solution. He therefore concludes that humans use non-algorithmic or uncomputable processes to solve these problems. Penrose argues at length and persuasively, but the argument may be flawed. Deep Blue is an algorithm which, although it does not guarantee a solution to the chess problem (a sin), still wins an impressive number of games. Humans often seem to use heuristics, rules-of-thumb, to attempt solutions. Like the algorithm underlying Deep Blue, these don't guarantee a solution, but nevertheless often produce one. The Gödel theorem speaks of algorithms which always guarantee a solution. The answer may be that humans succeed on these Gödel problems not through the use of incomputable (non-Turing) processes but through the use of heuristics. This doesn't rule out the possibility that humans may also use incomputable processes, but it seems to make it less likely. On the other hand, Penrose may be correct; in which case intelligence through computation will be unachievable.

Means to an End

Approaches to try to realize the potential in the Church-Turing Thesis fall into three categories: Symbolic ("Model the Mind"), Connectionist or Artificial Neural Systems ("Model the Brain"), and a relatively new body of practice grouped under the heading of Artificial Life ("Model Evolution").

Symbolic AI systems are designed and programmed "Top Down," rather than trained or evolved. They tend to be propositional using a list of rules and facts to simulate a general psychological theory of some aspect of intelligence, or to simulate the application of knowledge in some specific area of expertise; there are systems that simulate "reasoning," systems that simulate "knowing," and systems which do both. A common approach used in symbolic AI is the production system. It generally has three parts: a list of rules of the form IF-THEN, called production rules or productions; a control mechanism used to decide when and how to apply a given rule; and a working memory, a 'blackboard' where the results of rule activations or "firings" are posted. An IF-THEN rule representing facts might look like the following: If an animal has pointed teeth and if an animal has claws and if an animal has forward eyes, then the animal is likely a carnivore. A rule is "fired" when the IF-clauses are satisfied, and the results of the firing, the "THENs," are posted on the "blackboard." These results may be taken up by the IF-clauses of other rules causing them to fire in-turn. A typical production system will have thousands of rules. Many of the niche application examples cited earlier are based on production systems. Deep Blue incorporates production rules to evaluate the strategic worth of chess positions.

Production systems and other methods used in symbolic AI have been much less successful in more general problem solving; among the reasons is a lack of commonsense knowledge which, for example, would lead a system doing medical diagnoses to prescribe smallpox treatment for a car with rust spots. The number of "facts" which make up the body of commonsense knowledge is immense; probably millions of rules to represent enough knowledge so that new concepts could be "explained' in terms of previous rules and so that the system could "bootstrap" itself. A system exhibiting intelligence would also likely require additional millions of rules, describing reasoning processes. Aside from the fact that we don't understand these processes, the effort required to write such a program and to get it to work reliably poses a practical problem we don't know how to solve. A possible way around this problem is to create systems which can learn. While symbolic systems which learn can be constructed, they tend not to scale well; the more rules they learn, the slower they run. It is tempting to believe that the doubling of computational power every 18 months will solve this problem, but experience to date suggests that is unlikely. Symbolic approaches may also have a fundamental flaw; critics argue that rules and other symbolic means are only rough approximations of sub-symbolic processes underlying intelligence, and that to obtain intelligence these processes must be included.

Problems with scaling-up symbolic systems and concern about what they might have left out gave rise to the connectionist or artificial neural system approach, based on studies of the brain's architecture. The most salient characteristic of the brain is the dense interconnection among the neurons. Perhaps, the "hardware" does matter to some degree, and if many simple processors representing neurons were densely interconnected, brain-like behavior might result without writing millions of lines of code. Intelligence might spontaneously emerge from the interactions of many simple processors.

To investigate the connectionist of artificial neural system hypothesis, mathematical models loosely approximating some of the features of animal nervous systems have been constructed; these are largely limited from a few hundred to a few thousand "neurons" which are interconnected by links with variable weights or strengths. The neurons are generally modeled as a simple thresholding function. If the weighted sum of inputs to a neuron exceeds some set threshold value, the neuron fires and outputs a signal which goes to all those neurons to which it is connected. It is hard to see how such a simple process can give rise to complex behavior, but remarkable performances have been obtained; one neuron can't do much but networks of neurons can do a lot.

The key point about artificial neural systems is that they are trained, not programmed; they learn. Machines that learn are absolutely crucial to obtaining intelligent behavior. It is impossible to program in everything a machine must know to pass the Turing test or do much else. Deep Blue does not learn. If the size of the chess board were changed, another row or column added, or if some of the chess places were given additional moves, Deep Blue would be lost but a human would learn to cope. Learning in a neural net takes place by changing the pattern of weights which determines its response. Information is likewise stored in patterns of weights distributed across the net. Artificial neural systems excel at pattern recognition whether the pattern is visual, or exists in more abstract data. They do less well, thus far, in solving problems requiring explicit rational or logical thought where symbolic systems excel. The remarkable thing about artificial neural systems is that so much performance has been obtained out of (relative to the brain) ridiculously simple systems.

Mathematicians describe complex systems as ones which are composed of very large numbers of interesting parts. The brain certainly qualifies. A characteristic of complex systems is that they have emergent properties; properties which occur suddenly when a certain level of complexity is reached, and whose emergence could not have been predicted form knowledge of the parts and the interactions. Simulating at most a few thousand very simple neurons is well below the level of complexity at which intelligence might kick in. A single neuron doesn't think and isn't conscious, yet the brain does and is.

To explore the potential of artificial neural systems, and to see if intelligence will emerge at some level of complexity, will require the capability to simulate very large number of neurons and their interconnections. But numbers alone are not sufficient; it is necessary to model the neuron itself in more detail. To move beyond the simple caricature of thresholding and capture much more of the complexity inherent in the functioning of the biological neuron, and to acknowledge there are many different kinds of neurons.

Constructing a machine which might think using a connectionist approach now seems to be a hardware as well as a software problem; today's computers can't handle the required calculations in a reasonable time.

What is the size of the simulation problem? The human brain has about one-hundred billion neurons; by some reckoning it processes information at a rate of about a million-billion bits/sec (10^{15}) to 10 billion-billion bits/sec (10^{19}). It is much slower than a computer at the "component" level, but more than compensates with massive parallelism. The best simulations run at about 10 to 100 billion (10^{10}–10^{11}) bits/sec. The difference in performance is somewhere between a factor of 10,000 (10^4), to one of 1,000 million (10^9). If computer performance continues to double every 18 months, this difference will be erased somewhere between the years 2020 and 2040 without making any far-fetched assumptions about technology. That certainly will not automatically result in human intelligence, but it does suggest that if we understand enough about brain functioning, the computer capacity will exist in one form or another to exploit that understanding. But what it is we don't have is that level of understanding?

How, if we don't understand something can we replicate it? The answer leads us to the third approach to AI, derived from the relatively new discipline of artificial life. Artificial life started with the mathematician John von Neumann's work on self-reproducing automata and focuses on the simulation of biological processes. Two of these are key to creating an artificial nervous system: directed evolution and self-organization.

Directed evolution is a way to artificially speed up the process of evolution, and to direct the process toward an explicit goal; in this case, intelligence. One way to use it is to evolve the software we need to simulate a nervous system. We put code representing the functions of all the things we believe may be important to intelligence in a simulation. We randomly create some variants of these, run the simulations, pick some winners, let them "breed" (i.e., exchange some code), create offspring, throw in some mutations, and repeat the process. While this sounds simpler then it is, it captures the essential ingredients of the process. It has been used to create solutions to some very hard problems, but never on this scale. On the surface, this is what mathematicians would call an intractable problem; you not only have to simulate a nervous system, you have to simulate a large family of candidate systems, and do it perhaps thousands of times. We have now far outstripped the projected capabilities of computer technology. A possible solution may lie in using directed evolution on hardware and evolving a nervous system directly.

Imitating a nervous system in hardware is an enormous challenge. It may be that silicon itself, the stuff of chips, is unsuitable and that proteins or some other bio-material might be preferable. Two points seem clear: first, whatever the material, the system is so complex, containing perhaps millions of neurons, that it will likely have to be constructed by harnessing the capacity of some materials or components to self-organize or self-assemble into higher-order systems using instructions implicit in the components. Having to somehow otherwise connect up these neurons individually, as is done in semiconductor fabrication, seems impossible; particularly so when you realize that the architecture of the interconnections themselves is shaped by learning. You can't really specify all those connections in advance. Second, the self-organization would be pushed in a particular direction favoring intelligence, using directed evolution.

Two brief examples to suggest possibilities: first, a number of researchers have used self-organization to grow rat hippocampal neurons in patterns to form simple "circuits." Second, researchers at the University of Sussex have used directed evolution to evolve novel digital circuits. This is done by taking advantage of a type of integrated circuit called a field-programmable gate array in which connections among components on the chip are under software control. Very recently, other researchers, using the same technology, have started to evolve artificial neural systems in hardware.

Can A Machine Think?

There are good reasons to believe a sufficiently complex machine could one day pass the unrestricted Turing test. Whether or not that constitutes sufficient proof of intelligence or consciousness, will be the subject of continuing philosophical debate. Machines have been created (e.g., Deep Blue) which outperform humans in many niche areas. Some make extensive use of the particular strengths of computers such as rapid search and large memory, while others try to simulate human problem-solving or some of the machinery of the brain. Some of Deep Blue's predecessors tried to reproduce the methods of human grandmasters, i.e., recognizing key configurations of pieces of the board. Deep Blue relies more on massive and rapid searches of possible sequences of moves.

While it is not possible to predict which of the three major approaches to artificial intelligence might be the basis for an intelligent machine (perhaps all of them will be incorporated to some degree), it seems a safe bet that a major component will resemble the functioning of a human brain at the level of individual neurons. If we can simulate the functioning of the brain at a deep level, the resulting network would literally be a *tabula rasa*, a blank mind. It would not show intelligence nor consciousness unless it was subjected to experiences similar to those of a human brain. It must be able to investigate the environment around it, interact with that environment, and learn common sense and all the other things which contribute to intelligence, since you cannot directly program in intelligence. In the nearer-term, the pursuit of machine intelligence will continue to yield large benefits; we will be able to talk to our computers to dictate E-mail or documents, to command intelligent software agents to find information for us, and generally interact naturally with a variety of increasingly more complicated devices using spoken language.

There is no way to predict the impact of machines learning true intelligence; the only way to find out is to try and construct them. We may fail, but as we try we will learn much that is valuable about ourselves, and the brain that makes us human.