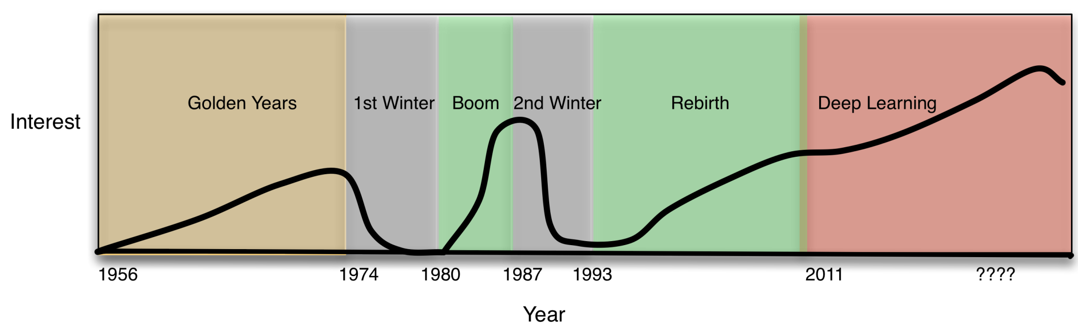
Intelligence Might Not Be Computational

Peter Denning
Ted G. Lewis

DRAFT OF 8/28/19 – v4

A hierarchy of AI machines by their learning power shows their limits and the possibility that general intelligence might not be computational

FIGURE IDEA: Time line showing boom and bust cycles in AI. The line shows the relative interest in supporting research in AI. The two busts ("AI winters") were the result of disillusionment about whether AI researchers could deliver what they promised in the first two booms. The third boom is different because AI is now supported by a large, thriving industry. But the boom depends on deep learning technologies. Artificial general intelligence, the next major goal, is beyond the reach of machine learning.



The goal of artificial intelligence (AI) is to construct machines that are at least as smart as humans at their tasks. AI has been successful with machines that learn how to recognize speech, find new classes of stars in sky surveys, win grandmaster chess matches, recognize faces, label images, diagnose diseases, hail taxis, drive cars, navigate around obstacles, and much more. Yet none of these machines is the slightest bit intelligent. How can they do intelligent things without being intelligent? Can these machines be trusted when presented with new data they never saw before? Businesses and governments are using AI machines in an exploding number of sensitive and critical applications without having a good grasp on when those machines can be trusted.

One way to answer these questions is to classify AI machines according to their relative power to learn and examine what makes machines in each class trustworthy. This way of classifying AI machines gives more insight into the trust question than the more common classifications by domains including speech, vision, natural language, games, healthcare, transportation, navigating, and so on. Domain overviews do not advise us on which machines are more powerful. For example, can a machine that learns how to win at chess be adapted to be a creativity machine? Or does a creativity machine require a new power not present in a game machine?

The AI machines we are about to discuss evolved in the AI field since its beginnings in the 1950s. The field has experienced three periods of boom punctuated by two periods of bust (often called "AI winters"). The first boom began around 1950. It produced useful prototypes of speech recognizers, language interpreters, game players, math word problem solvers, and simple robots. But the researchers were not able to deliver on their ambitious claims for production-quality systems and their research sponsors pulled back funds in the mid 1970s. Funding returned in the early 1980s, when the Japanese Fifth Generation Project poured large sums into AI research and high-performance logic machines. Other countries followed suit. That boom lasted until the late 1980s, when again the funding agencies were disappointed by lack of progress toward promised results. The third boom began in the early 1990s as technologies of machine learning began producing significant, useful, and often surprising results – accompanied by large doses of hype about the future of AI. A new bust is possible because AI researchers have placed big bets on achieving "artificial general intelligence"—which may be beyond the reach of machines.

An aspect of the hype that has been particularly troubling to us is the claim that all the advances in computing are from AI. In truth, computing has made steady progress in power and reliability over the past half century. By 2000, the available computing platforms were sufficiently powerful that they could support AI programs: modern AI would not exist except for the advances in computing. A recent report from the Organization for Economic Cooperation and Development (OEDC), a consortium of 34 countries, defined AI so broadly that any software is an AI and that all progress in computing is due to AI. Although that is nonsense, it shows the political power that can gather behind hype.

The term machine learning refers to machines that learn their function from many examples rather than from rules set by programmers. Machine learning has proved extremely useful and successful not only because of scientific advancement but also because of cheap, fast, hardware. In the task we set for ourselves -- classifying these machines and defining their limits – we struggled against two impediments. One is that there is no scientific definition of intelligence. Arthur C Clarke's admonition—"Any sufficiently advanced technology is indistinguishable from magic"—captures a well-known phenomenon in AI: once we succeed at building an intelligent machine, we no long consider it intelligent. As soon as the magic is explained, the allure fades.

The second impediment is our tendency to anthropomorphize – to project our beliefs and hopes about human intelligence on to machines. For example, we believe intelligent people think fast, and yet supercomputers that run a billion times faster than humans are not intelligent. We believe that interacting communities of AI machines will be collectively smart, and yet massively parallel computers and networks are not intelligent. We hope for creative machines that make new discoveries but do not know how to tell whether a Eureka come from a machine or from a human using it.

The hierarchy we will discuss does not rely on any definition of intelligence to classify AI machines. What differentiates the levels of the hierarchy is simply that the machines at a lower level cannot learn functions that higher level machines can. This is scientifically quantifiable. No anthropomorphizing is needed.

In 1988, AI pioneer Hans Moravec offered a paradox: that the hard problem of general intelligence seemed to require much less computation than easy problems such as motor skills or recognizing faces. Moravec's paradox has been explained by experience with AI systems: building machines to behave like humans is far harder than anyone thought. Intelligence might not even be a computation at all.

A Hierarchy of Learning Machines

There are three main ways that AI machines have been classified: by domains, by apparent intelligence, or by programming method. These approaches are insufficient for our purpose of defining the limits and capabilities of machines. The domains approach begins with a domain of human activity such as vision, speech, or gaming and discusses the kinds of machines that have been built for those domains. This approach is not useful for classification because there has been little machine transfer between domains: it is hard to tell if a machine for one domain (say vision) could be adapted for another (say speech).

The intelligence approach puts machines into categories depending on the degree of human intelligence apparently needed to do tasks. Because there is no scientific definition of intelligence, classifications based on apparent intelligence are controversial.

The programming approach considers how much programming a machine needs compared with self-learning. The programming approach has received a lot of attention in recent years. Some AI researchers make a distinction between an externally programmed machine and self-programming machine. Externally programmed means that human programmers design software (and hardware) that carries out the specified function of the machine. Self-programming means that the desired behavior emerges as the machine modifies itself while observing itself and its environment, For example, expert systems are programmed by designers who specify the inference rules of the system; neural nets are "trained" by showing them examples of proper input-output pairs and adjusting internal parameters so that the network provides the right output for each input. Unfortunately, this distinction is not as clear in practice as it sounds. For example, the training of a neural network is done by an algorithm called back propagation, which works backward from the output, adjusting network parameters for least error between the actual and desired outputs. Learning systems based on statistical inference are considered more powerful than neural networks because they can discover classes of input objects whereas the trainer of a neural network needs to know the classes before training starts. However, advanced statistical inference algorithms are programmed by humans; no self-programming is involved.

We propose a classification based on learning power – empirically verifiable assessments of the learning capabilities of machines. This is not a classification by computing power – all levels can be shown to be Turing complete. The hierarchy shows that none of the machines so far built has any intelligence at all, leading to the tantalizing possibility that human intelligence is not computable.

Table 1. A Machine Intelligence Hierarchy

0	Basic automation
1	Rule based systems
2	Supervised learning
3	Unsupervised learning
4	Multi-agent interactions
5	Creative AI
6	Aspirational AI

Level 0—Automation

The baseline of the hierarchy is basic automation—designing and implementing automata that carry out or control processes with little or no human intervention. The purpose of automation is to take the human out of the loop by substituting an automaton to do the work. Automation frequently includes simple feedback controls that maintain stable operation by adjusting and adapting to readings from sensors – for example, a computer-controlled thermostat for regulating the temperature of a building, an autopilot, or factory assembly robot. The automaton cannot learn any new actions because its feedback does not change its function. By taking over tasks that humans do poorly, these machines amplify and extend human performance. This kind of automation is not a form of machine intelligence because the automata do not learn anything beyond what they were built to do.

Nonetheless, automation is difficult to define precisely. There is a continuum from fully manual performance of a task to fully automated. Many automated systems do not fall at the extremes of this spectrum – for example, an autopilot has an off switch that enables the human pilot to take manual control when the conditions are not safe for autopiloting. In recent years, AI systems have enabled previously impossible automations – for example, speech recognizing robots regularly replace human operators in call centers. For these reasons, we say that AI is an enabler of automation, but not a form of automation.

Level 1—Rule-based systems

Philosophers through the centuries rated reasoning power as the highest manifestation of human intelligence. Rene Descartes and later Gottfried Leibniz argued that many disputes could be resolved if only people could learn to be logical and rational and keep emotion out of their discourse. They argued that reasoning is a process of making logical deductions from given axioms and facts. Within this background, AI researchers were attracted to programs capable of imitating the rational reasoning of humans. These were called "rule based programs" because they made their logical deductions by applying programmed logic rules to their inputs and intermediate results.

Board games were early targets for rule-based programs. Arthur Samuel of IBM in 1952 demonstrated a competent checkers program. AI researchers turned their attention to the much harder game of chess, which they thought could be mechanized by brute-force searching through thousands of future board positions and picking the best moves. That long line of work climaxed in 1997 when an advanced chess program running on an IBM Deep Blue computer beat Garry Kasparov, the world's grandmaster at Chess. Computer speed is the major reason for success – the computer can search through billions of moves in the same time a human can search through perhaps hundreds.

Another early target for rule-based programming was expert systems -- programs that solve problems requiring expertise in a domain. Their logical rules are derived from the knowledge of experts. The first prototypes were developed by Edward Feigenbaum at Stanford University in 1965: "Dendral" is a system for identifying unknown organic molecules, and "Mycin" is a system for diagnosing infectious blood diseases. In 1980 John McDermott of Carnegie Mellon University developed an expert system (XCON) for the Digital Equipment Corporation. Given customer requirements, XCON recommended configurations of their VAX computer systems and by 1986 was reckoned to have saved the company $25M annually.

The builders of expert systems soon discovered that getting experts to state their expertise as rules was an impossible task: experts seem to know things that cannot be described as rules. Hubert Dreyfus, a philosopher and an early critic of expert systems, argued that much of what we call expertise is not rule based, and a machine limited to rule-based operations might be competent but could not be expert. Although a few systems such as Mycin and XCON proved to be competent, no one has built a true expert system.

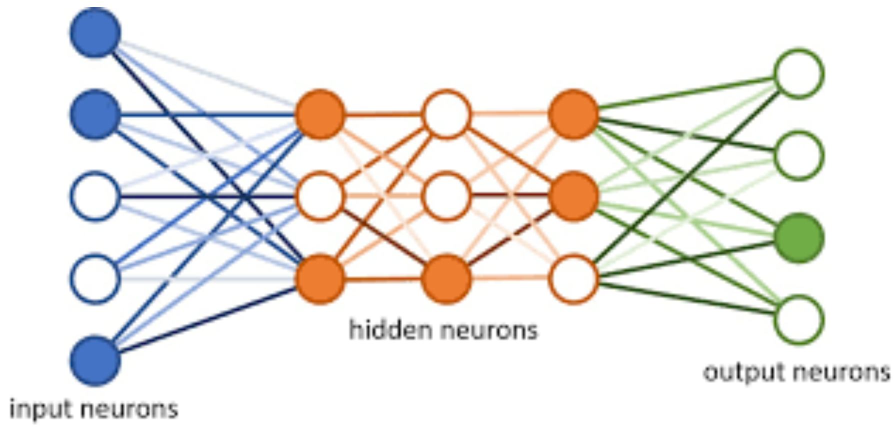


Figure 1. An artificial neural network (ANN) is a network of nodes, each an electronic component reminiscent of a logic gate. Nodes are arranged in a series of layers, each providing the input for the next. The input layer drives one or more hidden layers, the last of which drives an output layer. A node "fires" (enters the "1" state) when the weighted sum of its inputs from the previous layer exceeds a threshold. The weights are parameters that can be adjusted by a training algorithm so that the output for a given input matches the desired output. In this example, the network takes 5 input bits and produces 4 output bits. Whereas training a network is usually expensive and time consuming, the trained network is lightning-fast.

Level 2—Supervised Learning

At this level the machines computes outputs not by applying logic rules to inputs but by remembering in their structure the proper outputs for each of a set of inputs shown it by a trainer. The artificial neural network (ANN) is a common example (see Figure 1). The neural network is so named because its design loosely imitates the structure of a brain with many neurons interconnected by axons and dendrites. Natural neural networks from brains were studied by biologists in the late 1800s; artificial neural networks were studied in the 1940s because some engineers believed that a computer structured like a brain might be able to perform like a brain.

The trainer of an ANN works with a set of data consisting of input-output pairs. These pairs are numerous examples of the function the trainer desires the ANN to learn. The outputs are often called labels, because neural network is asked to recognize and label the data presented at the input. For example, the inputs may be bitmaps of photographs of faces, and the outputs are the names of the people in the photos. The trainer hopes that the trained ANN will correctly recognize all the faces in the training set. The trainer also hopes that the trained ANN will correctly recognize faces in photos that were not part of the training set.

The network trainer uses an algorithm called "back propagation" to set the inter-node connection weights. Back propagation means that the algorithm works backwards from the desired output, adjusting weights for inter-node connections until it finds a least-error set of weights. Because the errors between actual and desired outputs may not be zero once the weights are chosen, there is a chance that the network may produce incorrect outputs. Because of the sheer number of nodes and links in a useful network, ANNs can take days to train but, once trained, they compute their outputs within milliseconds.

Today's ANNs suffer from two main limitations. One is fragility. When presented with a new (untrained) input, the ANN's output may deviate significantly from the desired output. Moreover, when a small amount of noise disturbs a valid input, the ANN may label it incorrectly. For example, a road-sign recognizer for a driverless car can be fooled into labeling a stop sign as a speed limit sign simply by placing spots of masking tape at strategic locations on the stop sign. Two ANNs trained from different data samples of the same population may give very different outputs for the same input. All these behaviors reduce the trust that users are willing to place in ANNs.

The other ANN limitation is inscrutability. It is very difficult to "explain" how an ANN reached its conclusion. This problem does not exist with conventional programs, because we can trace the execution of the program and pinpoint the portion of code that causes the erroneous output. The only visible result of training an ANN is a very large, gigabyte-sized matrix of connection weights between nodes. It may not be possible to explain a problematic ANN because the "explanation" is diffusely distributed among thousands of weights.

Training ANNs is expensive. Some of the expense is in the training process, which can take several days of back-propagation computation on a large data set. But there is also an expense in getting training sets. There are commercial companies that hire armies of people to manually label images in their spare time. Because quality control may not be rigorously enforced in the army of labelers, these labeled image sets can not only be expensive to obtain, but may not be accurate examples of the desired function.

Level 3—Unsupervised learning

Machines at this level learn to improve their performance by making internal modifications without the assistance of an external training agent. They are attracting more research attention because they can potentially eliminate the large cost of obtaining training sets.

An early example is a program called AUTOCLASS, built in 1988 by Peter Cheeseman. AUTOCLASS computed the most probable classes of 5,425 experimental observations from the NASA infrared telescope; with one exception it agreed with the classification already determined by astronomers, and the exception was seen as a new discovery. AUTOCLASS accurately inferred the classes of the infrared objects without any advance knowledge of how professional astronomers had classified the objects.

To do this, Cheeseman used Bayesian Learning, an advanced method from statistics that creates a hypothesis (in this case, a proposed set of classes) and computes the conditional probability of that hypothesis given the data. It then iteratively modifies the hypothesis by exchanging objects between proposed classes, seeking a higher probability hypothesis. This iteration ends when the highest probability hypothesis is found.

Another example where there is no training set arises in the play of strategy games such as Chess and Go. The recent success of a machine called AlphaGo at the game of Go demonstrated the approach. Go, an ancient game very popular in Asia, is considered orders of magnitude more difficult than Chess. AlphaGo made a dramatic debut in 2016, beating Go grandmaster Lee Sedol of South Korea. The AlphaGo machine was trained by playing against another AlphaGo machine. The two played a massive number of rounds, recording all their moves in each round. When one won a game, it earned a reward, which was propagated back to all the moves that led to the win, thus reinforcing those moves in the next round. At the start, the only prior information given to the machines was the statement of the rules of Go – but no examples of Go games.

AlphaGo was built by the Google subsidiary DeepMind. After their success with Go, they wondered if the AlphaGo platform could be modified to learn Chess and another two-player strategy game Shogi. They renamed their machine AlphaZero because it was more general than AlphaGo. Using the two-machine training method outlined above, AlphaZero learned to play grandmaster Chess in 9 hours, Shogi in 12 hours, and Go in 13 days.

This short learning timeframe is significant; Chess programs took 30 years to develop to the point where they could beat a Chess grandmaster. AlphaZero, about six years in development, took 9 hours for the same accomplishment. No one had previously been able to build a grandmaster Go playing machine; AlphaZero did that in 13 days.

The most common speculations for what AlphaZero might be used for next involves extending the term "game" to include not only games such as chess or Go, but also business games, marketplace games, or war games. These generalized games all have well defined rule sets describing reward functions, allowable moves, and prohibited moves. The AlphaZero method may not work with social systems where the game is not well defined but must be inferred by observing the play.

These two examples – AUTOLCASS and AlphaZero – illustrate the principle of unsupervised learning. Machines can learn complex tasks without being shown examples of those tasks. They do so without expensive, difficult-to-obtain training sets.

Level 4—Multiagent interactions

At this level, machine intelligence emerges from the interaction of thousands or millions of agents, each with a specific function. An agent is an autonomous machine or code segment. The machine learning capability arises from the collective. This idea was discussed by AI researchers beginning in the 1960s and was the basis of HEARSAY, a speech-recognition system, in the 1970s. It morphed into "blackboard systems" in the 1980s and was epitomized by the late AI pioneer Marvin Minsky in his theory on *Society of Mind*. A blackboard is a shared knowledge space that agents continually read and update until they converge on a collective solution to a problem. Since then a large amount of AI research has focused on building large networks of interacting agents. So far, nothing close to human intelligence has emerged when the agents are all machines.

The story is different when some of the agents are humans. In the 1960s Doug Englebart of SRI International proposed that computers are most useful when they amplify human intelligence by supporting collaboration among humans. He invented a bundle of key tools that are in common use today, including the mouse, windows, hypertext lists, real time voice and video, video inset windows, and shared desktop screen-spaces seen by all participants in a meeting. These tools are not AI tools per se, but they supported Englbart's objective of amplifying human intelligence by enabling teams of humans and machines to work together.

This is not the only successful example of human-machine agent teams. After IBM Deep Blue beat him in 1997, Garry Kasparov invented a new kind of chess that he called Advanced Chess, where a "player" is a team consisting of a human augmented by a computer. It was soon found that the teams of competent players and good chess programs were able to defeat the best machines.

Another example of successful teaming can be found of all places in high school robotics competitions, where teams of human navigators augmented with autonomous function agents are the most frequent winners.

Another example is web surfing, the navigational process of using web searches to find answers to questions. Humans interact via a browser with a network of servers in the Internet. The servers locate web pages that may answer the question posed by the human, and the human selects the best response and follows up with more searches to refine the answer.

Although it is possible that a cacophony of interactions among agents could overwhelm the network connecting them, the successful examples of human-computer teaming show that a well-designed system of human and machine agents can perform their functions more effectively than any human or machine acting alone.

These examples show that human-machine teaming is a rich area and can often be achieved with simple interfaces that do not rely on AI tools.

The success of human-machine teams has exposed a rift among AI researchers. Some want machines that are intelligent on their own, with no human assistance. Others believe that a team of a machine and humans can outsmart the same machine operating alone.

Level 5—Creative AI

This level is intermediate between machines that support creative teams and machines that demonstrate general intelligence. The question is: can there be a machine that is creative on its own without the benefit of a team? At the current stage of the technology, there are no working machines that demonstrate either this level or the next. We can discuss proposals, but they are speculations about machines that have not been built.

Creativity means to open new possibilities in social space that did not exist before. It is not the same as surprise, which means that something unexpected happened – we can be blind to a possibility that already exists. Humans exhibit creativity all the time. Can machines do it? This is a controversial topic.

Jeff Dean, head of AI division at Google, says, echoing Englebart, "We want to use AI to augment the abilities of people, to enable us to accomplish more and to allow us to spend more time on our creative endeavors." He sees AI as the source of ever more powerful tools that enable greater heights of human creativity. He has not commented on whether a machine can be creative on its own.

Some AI researchers have speculated that creativity is recombination of existing ideas and have experimented with machines that do that. An early example is the TRIZ algorithm, invented around 1946 by Genrich Altshuller, Soviet inventor and science-fiction writer. TRIZ searched a patent database and proposed new combinations of patent claims, many of which led to new patents. Another example is the genetic algorithm, popularized around 1975 by John Holland at University of Michigan to find near-optimal solutions to problems by simulating genetic mutation and cross-combination. An early example was a US Navy robot that could find its way through a mine field without being blown up. The robot programs started as random strings. Each program was rated with a fitness value based on its demonstrated ability to guide the robot safely. Through generations of refinement in the genetic algorithm, they evolved into programs that succeeded in navigating safely through mine fields.

Artists and musicians have experimented with AI tools to produce new art forms. The Prisma app, which transforms photographs into art images in the style of famous painters, is an early example. Ahmed Elgammal of Rutgers University demonstrated works of art generated by a neural network machine called AICAN, which generated new images of paintings that resembled some existing paintings but differed in significant, novel ways. Some of those images fetched handsome prices at auction. He cautioned that AICAN produces new art in comparison to a database of art images; it is not sensitive to context and cannot create art that helps people find meanings in the issues and concerns lurking in the unstated background of their experience. His conclusion was that while the AI appeared artistically creative, it was not as creative as an artist armed with an AI tool. In effect the artists working with AI tools are human-machine teams of greater artistic creativity than machines alone. A similar conclusion applies for music and, indeed, for other domains of art.

These examples of creative machines appear as unsupervised learning, as for genetic algorithms and other recombinant machines, or as human-machine teaming. We do not know of an example of a creation machine that is more powerful than unsupervised learning or human-machine teaming.

If creativity could be interpreted as a mechanical process, it could certainly be realized by a machine. However, creativity seems to be a deeply social process involving many human assessments about new possibilities and contexts. It may not be possible to build a machine that rises, on its own, to this kind of creativity.

Level 6—Aspirational AI

This level includes a variety of speculative machines that represent the dreams of many AI researchers. The most ambitious dreams feature machines that think, reason, understand, and are self-aware, conscious, self-reflective, compassionate, and sentient. No such machines have ever been built and no one knows whether they can be built.

Early on, AI researchers realized that AI machines lacked common sense. Early medical expert systems, for example, were prone to make mistakes no doctor would make. Researchers thought that the solution was to gather a large compendium of common-sense facts and rules in a very large database for use by the expert system. In 1984 Douglas Lenat, CEO of Cyc Corporation, set out to build such a machine, which he called Cyc. His project continues to this day. The machine, which now contains millions of common-sense facts, has never succeeded in helping an expert system behave like a human. Dreyfus was not surprised. He wrote, “Al researchers have been working to solve the problem of getting computers, which are syntactic engines sensitive only to the form or shape of their input, to behave like human beings who are sensitive to semantics or meaning.” For example, we know from experience that we cannot chew gum and whistle at the same time. There is no rule that states this. We know it because we can imagine trying to do it. Dreyfus believed that it is an impossible task to formalize for machines the understanding of the world we have by virtue of being able to imagine what our bodies can do.

Much of AI research has been predicated on the assumption that the brain is a computer and the mind is its software. It follows that creating an artificial intelligence is a programming problem. As early as the 1970s, some AI researchers challenged this assumption. Cognitive scientists now believe that the structure itself of the brain—intricate folds, creases, and cross connections—gives rises to consciousness as a statistical phenomenon of brain activity. Therefore, the “mind” cannot be separated from a particular brain or transported to another kind of computer. Commenting on this, brain neuroscientist Christof Koch says: “To create human-level consciousness in a machine, the intrinsic causal powers of the brain must be instantiated at the level of the transistors and wiring making up the hardware. The intrinsic causal power of contemporary computers is puny compared to those of brains. Thus artificial consciousness calls either for computer architectures radically different from today's machines or for a merging of neural and silicon circuits as envisioned by transhumanists."

The obvious implication is that intelligence is biological, not just a computation. But even this explanation is insufficient. It appears that much of what we think we know is actually distributed in the social networks of which we are a part; we "recall" it by interacting with others. Most of what we "know" is not in our heads, but is in our surrounding social communities. Chilean Biologists Humberto Maturana and Francisco Varela argued that biological structure determines how organisms can interact and that consciousness and thought arise in human networks of coordination of actions. A conclusion is that software and biologically constructed machines will not be sufficient to generate machine intelligence. In ways we still do not understand, our social communities and interactions in language are essential for general intelligence.

Unsolved Questions

By considering a hierarchy of AI machines focused on demonstrated capacities to learn to solve problems, we have sought to create a framework to understand the powers and limitations of the AI machines we now have. The greatest successes have come from neural networks, Bayesian learning machines, and reinforcement learning machines at Levels 2 and 3. Human-machine teaming at level 4 shows great promise. Automatic recombination at level 5 is useful but in the known examples it appears as unsupervised learning or human-machine teaming, and does not reproduce human creativity.

Level 6 expresses the fictionalized ideals of AI. Its speculations about future machines are often enchanting and engaging. But there are serious problems with the assumptions behind these speculations. For example, the assumption that human behavior is ultimately rule- and fact-based is false. The assumption that structuring a machine like a human neural network will generate human behavior is false. The assumption that human intelligent behavior arises in the mind associated with a brain has been called into question partly because a mind does not behave like software and partly because many aspects of mind arise from social interactions and conversations. We can expect continued advances in the tasks that machines can do as long as we do not ask the machines to understand relevance or meaning. These problems may bring on another bust cycle for general artificial intelligence.

New applications of AI are announced every day. This does not mean AI technology is yet advancing toward Levels 5 and 6 as described here. AI is getting better at Levels 2 through 4. We need to separate excitement over new or improved applications from true advancement in the power of an algorithm to solve a certain class of problems. The AI hierarchy described here makes this possible by ranking according to learning power demonstrated in applications. This is a way to measure true progress.

At the start of this essay we suggested there are dangers in the growing dependence on AI when its operations and trustworthiness are not well understood. The hierarchy gives a way to understand the principles of operation and limits of existing AI technologies. It gives you a way to recognize hype and resist its lures. Recent examples where unquestioning faith in claims about AI has led to serious problems include healthcare, fire insurance, and even deciding jail times for convictions in court.

The hierarchy leads to the tantalizing – though likely unpopular – conclusion that human intelligence is not computable. Three aspects support this conclusion. First, nothing remotely resembling an intelligent machine has ever been built with electronics despite 70 years of trying, or with mechanical automata despite 250 years of trying. Second, machines are designed to work locally – each move depends only on the local configurations and forms of signals and symbols. The sacrifice of sensitivity to context is rewarded with extreme speed of calculation, but speed is not intelligence. Third, machines do not have flesh-and-bone bodies and therefore cannot have any understanding that comes through our bodies such as language, imagination, history, concerns, anxiety over health, or relationships. It may be that the epitome of machine AI is to support human-machine teaming. And that is perfectly OK.

Bibliography

Bostrum, N. 2014. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 328 pp.

Cheeseman, P. et al. 1988. AUTOCLASS: a Bayesian classification system. In *Proc. Fifth Machine Learning Workshop*, Morgan-Kaufmann: 54-64.

Dreyfus, H. 1972. *What machines (still) cannot do.* MIT Press. [Updated in 1978 and 1992.]

Dreyfus, H. 2001. *On the Internet*. Routledge.

Elgammal, A. 2019. AI is blurring the definition of artist. *American Scientist 107*, 1 (Jan-Feb): 1. Available: https://www.americanscientist.org/article/ai-is-blurring-the-definition-of-artist

Kanerva, P. 1988. *Sparse Distributed Memory.* (2nd edition 2003). MIT Press.

Koch, Christof. 2019. *The Feeling of Life Itself: Why Consciousness is Widespread But Can't Be Computed.* MIT Press.

Maturana, H., and F. Varela. 1988. *The Tree of Knowledge*. Shambala.

Minsky, M. 1986. *The Society of Mind.* Simon and Schuster.

Silver, D. et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science 362*:1140-1144.

Sloman, S., and P. Fernback. 2017. *The Knowledge Illusion: Why We Never Think Alone.* Penguin.

Winograd, T., and F. Flores. 1987. *Understanding Computers and Cognition*. Addison-Wesley.

Zador, A. M. 2019. A critique of pure learning and what artificial neural networks can learn from animal brains. *Nature Communications*, an online journal, https://doi.org/10.1038/s41467-019-11786-6