The Natural Logic of Artificial Intelligence or, what genetic algorithms really do*

Mario Carpo

*This essay derives from the text of two papers I presented in Turin in the summer and fall of 2018, first at the PhD course of excellence Schema. *Towards a philosophical-architectural dictionary* organized by the Doctoral School of the Politecnico di Torino, then at the conference *Scienza Nuova: Humanities 4.0* organized by the Jacques Derrida / Law and Culture International Chair of Philosophy, Università di Torino, on October 15, 2018. It refers to arguments I discuss in The Second Digital Turn (Cambridge, MA: The MIT Press, 2017: see in particular pages 9-54) and in "The Alternative Science of Computation," *E-Flux*, New York (electronic publication, June 2017: http://www.e-flux.com/architecture/artificial-labor/142274/the-alternative-science-of-computation/). See now also Carpo, "Particlised. Computational Discretism, or the Rise of the Digital Discrete," in Gilles Retsin, ed., "Discrete: Reappraising the Digital in Architecture," *AD, Architectural Design* 89, no. 2 (2019): 86-93.

Computers are machines; but they are machines that are different in spirit, and in their own technical logic, from any other machine we have known before of after the industrial revolution. Computers are not mechanical machines--and they are certainly not organic systems, either; they are something else and new and unprecedented. If in doubt, let's just look at what computers do--at the way they work: *a fructibus eorum cognoscetis eos*. If we use computers to make things--to produce physical stuff--computers make things more or less the way pre-industrial artisans did; not the way any modern engineer would. Gilles Deleuze would have loved that, had he lived to see it. And if we use computers to think, or something akin to that, computers think more or less the way a child could, not the way any modern scientist would. But computers think like children that never grow up, as they do not have to: thanks to their immense memory and processing power, their childish way of thinking is so effective that computers never need to grow out of it.

Let me tackle the two topics--making, and thinking-- separately, as these are two different stories. First, physical production. The technical logic of mechanical mass-production is well-known: starting with the archetypal technology of modernity, Gutenberg's press, most technologies of mechanical reproduction typically used casts, molds, dies, or stamps to replicate identical copies. Molds or casts cost money and once they are made, it makes sense to use them many times over and for as long as possible, to amortize their cost by spreading it over many copies. This is the logic of mass-production, which achieves economies of scale through standardization, and the reproduction of identical copies: its iron logic is well-known, too: *the more identical copies we make, the cheaper each copy will be.*

But digital fabrication, as we know it today, does not use casts or molds; when digitally made, each piece is individually carved or milled out of a block of pre-existing material, or printed out of nothing, almost, by today's 3D printing tools. As there are no mechanical matrixes to begin with, there is no need to amortize their cost, hence there is no economic incentive to make more copies of the same item: likewise, when needed, each piece can be different from all others, at the same cost per piece--just like in traditional artisan production. The curve of marginal costs, which is asymptotic in all matrix-based mechanical making, is theoretically flat in most digital fabrication processes. In digital making, *making more of the same will not make anything cheaper.* As there are no economies of scale in digital fabrication, digital making is in no need to ever "scale up"; for the first time since the end of the Middle Ages, bigger today no longer means cheaper: bigger factories and bigger markets no longer mean cheaper goods. This new technical logic, which digital designers have been advocating since the 1990s, has a name: *digital mass-customization,* or non-standard seriality; and it means, literally: the serial reproduction of non-identical parts; the mass production of variations at no extra cost.

In more recent times the same logic has been spreading from production to commerce, and from commerce to finance, with similar results. Extrapolating from this technical logic, some have even imagined a future "zero marginal cost society," where some, or even many products and services will cost nothing; and, an even longer shot, a society where all human labor has been eliminated--a society of universal plenty where all laws of modern economics, both socialist and capitalist, will simply cease to exist.

But, leaving that matter stand for the time being, let's move to my

Philosophy Kitchen Extra #3 — Anno 6 — Luglio 2019 — ISBN: 978-88-941631-2-4 — Schema. Verso un dizionario architettonico-filosofico

second topic: computers as machines for thinking (as opposed to machines for making). As we all know, for the last 3 or 4 years or so everyone has been talking about Artificial Intelligence, and this may be surprising for the older among us, as the term "Artificial Intelligence" itself (or AI) is not new, and we may remember the time when it was already very popular, in the 1960s and early 1970s. Indeed, the term was introduced by computers scientists in 1956 as an alternative to Norbert Wiener's cybernetics, as Wiener's cybernetics had a strong emphasis on neurophysiology that many computer scientists back then found suspicious, or worse. Remarkably, in spite of colossal investments, particularly by the military in the age of the Cold War, both Wiener's cybernetics, and early Artificial Intelligence never produced any usable results. When it became evident that even the best mainframe computers of the time could not do much more than high school arithmetic, military funding dried up, and most AI projects were abandoned. This was the beginning of what computer scientists to this day call "the long winter of Artificial Intelligence." In short, the history of Artificial Intelligence is the history of a fiasco. Whereas what many call AI today very often seems to works: nobody knows for sure what AI stands for today, but most commercial computers already carry out increasingly intelligent tasks, and today's computers, unlike those of the 60s, easily win at games of checkers, chess, and Go, against the best human champions. Today's computers can even, almost, drive cars. Why then should we call all this AI, if AI is a technology that was already tried, and that famously failed in the past?

One reason may simply be that the use of this vintage term today may be a misnomer--i.e., wrong, and likely misleading. Another reasons is that many of the AI tools and strategies that work today are not very different from those that did not work in the 1950s and 1960s; but today's computers are so much more powerful than those of yesterday that computer programs that could be conceived, but could not work back then, do work right now, due to mere technical progress. And this, I am told by experts, is indeed part of the story. But another part of the story is that the extraordinary power of today's computers, when compared to those of 50 years ago, has brought about an actual methodological shift, which goes beyond mere quantitative progress. What computer scientists today call Brute Force AI, or Dataism, or Big Data Computation, is not more of the same old game; it is an entirely different game, a game with a different spirit, and a different logic: and this new game is nurturing what appears to be an entirely new scientific method--or perhaps we should call it, a new post-scientific method; or even *a new kind of science*.

Let me try to explain that in brief. Vintage AI aimed at the imitation of well-established human processes: one school of AI favored the imitation of the deductive methods of the mathematical sciences, based on some formalized rules for any given task or discipline (for example, the rules of grammar for a program meant to translate between languages). To the contrary, another school of thought favored inductive processes, based on iterative trial and error and on various optimizing strategies meant to reduce the number of trials needed for the extrapolation of more general statements. The first of these two methods imitated human science; the second, human learning; both followed established patterns of Western science (induction, formalization, deduction, rationalism or empiricism); and by the way neither imitated the physiology of the human brain, in spite of some fancy terminologies then adopted and still in use ("Neural Networks", for example). Most of these strategies were invented in the

50s, developed in the 60s--and abandoned in the 70s. The former of these two schools of computational thought favored so-called knowledge-based systems, aka expert systems, or rule-based systems; the latter is often called, by contrast, the connectionist school.

Fast forward to today. One generation after the invention of the PC and the rise of the Internet (which by the way no cybernetician nor AI guru of the 1950s, 60s, and 70s ever anticipated), the novelty of today's Big Data computation (or Dataism) is that, for the first time ever in the history of humankind, there seems to be no practical limit to the amount of data we can capture, store, and process. This is an unprecedented, almost anthropological change in the history of the human condition. Since the invention of the alphabet till a few years ago, *we always needed more data than we had*; today, for the first time ever, *we seem to always have more data than we need*. Humankind has shifted, almost overnight, from ancestral data penury to a new and untested state of data affluence. One of the first techno-scientific consequences of this truly Copernican upheaval is that many traditional cultural technologies and social practices predicated on our supposedly permanent shortage of data are now, all of a sudden, unnecessary and obsolete.

This is a subject I discussed at length elsewhere, but to make a long story short, big data computation today no longer needs to replicate the small data logic of either human science or human learning. Modern science used to compress the infinite variability and complexity of the world into short and memorable mathematical formulas--small formulas that were made to measure for human thinking; formulas we can comprehend within our mind; formula we can work with. The human mind needs small formulas, instead of big data, because the human brain cannot easily work with big numbers: the human brain was never hard-wired for big data. But today's electronic computers are. *What today we call big data simply means data that are too big for us; but which computers can work with just fine*. Not surprisingly, this is where human science, and Big Data computation, start to follow two different paths, or methods, and to function in two very different ways.

Scientific induction, or inference, is the capacity we have to construe general statements that go beyond our recorded experience. We need this capacity because our recorded experience is limited. But: let's assume, *per absurdum*, and to the limit, that we can now build a machine with almost unlimited, searchable data storage. *Such a machine would not need to construe general statements that go beyond its recorded experience, because its recorded experience could be almost infinite.* Consequently, this machine would have perfect predictive skills without any need for mathematical formulas, or laws of causation--in fact, without any need for what we call science. Such machine could predict the future by simply retrieving the past. The search for a precedent could then replace all predictive science; a Universal Google Machine would replace all science, past or future. The motto of this new science would be: Don't calculate; Search. Or, to be more precise: Don't calculate; Search for a precedent, because *whatever has happened before, if it has been recorded, if it can be retrieved, will simply happen again, whenever the same conditions reoccur*.

Of course here one would need a lot of small print to define what "the same conditions" means--which would bring us back to some core tropes and problems of the modern scientific method; and indeed many traditional scientific tricks and trades and shortcuts of all kinds still apply at all steps of the new computational science of Big Data. Conceptually, however, and ideally, this is this is the main difference between yesterday's AI and today's; the main reason why AI works today and in the past it didn't. AI today does not even try to imitate the abstractions and generalizations of human thinking; instead, AI today solves problems by storing a huge amount of precedents, in the raw--as they come; as found; and then searching this immense data-base--looking for the right precedent--whenever needed. No human could work this way, because it would take forever, which is why we mostly don't work this way. But computers do. This is why we humans invented a scientific method, based on comparison selection formalization generalization and abstraction, based on rules formulas axioms and laws of causation; which post-human computers don't need, and don't use.

This is how computers today can translate between languages--not by applying the rules of grammar, as Noam Chomsky thought long ago, but looking for the record of pre-existing translations validated by use. This is how computers win chess matches: not by applying the rules of the game, but by searching for a suitable precedent in a universal archive of all games already played. Indeed, in this instance, as in structural engineering, computers can do more than that: they can simulate all kinds of fake precedents on demand, playing against themselves; and these simulated precedents will be just as good, for predictive purposes, as real historical ones. Among so many reliable precedents, either real or simulated, it does not take any degree of intelligence, either natural or artificial, to find a good solution for any given problem. This is how computers today win a game of chess; and this is how in engineering we can already use post-human, Big Data computation to solve problems we could not solve in any other way.

Let me show that, to conclude, with a real-life example, the 2012 ICD/ITKE Research Pavilion, built in Stuttgart, Germany, by Achim Menges's team a few years ago. How do you think this structure was calculated--using the kind of science that all engineers of my generation studied at school? No; that would have been impossible; because that pavilion was made of millions of different filaments; and calculating each one of them in the traditional way would take forever. Instead, the authors of this building started with a random, perfectly arbitrary geometrical and material layout, in this instance inspired by biological models; then calculated the structural behavior of this first model using computational FEA. FEA is a design method that subdivides a continuous structure into a huge amount of discrete particles, then calculates the equilibrium and interactions among all of these very small parts. FEA a is conceptually simple method, but it results in so many calculations and with so many big numbers that only recent electronic computers can solve its equations and pull some usable results out of it.

But this is only the first step; as the initial structure calculated this way was only meant as a random sample--a shot in the dark, so to speak. Designers were expected to work on it and improve it, based on the results of the first verifications they had carried out. And this is how they did it: again randomly, and blindly, they tweaked some aspects of the geometry of the shell and of the internal layout of the fibers; then they reran the FEA calculation on this second model, and so on: the process was repeated many times over, until the authors were pleased with the results. In this process of optimization by trial and error, every simulated model that was tested and discarded corresponds to a physical model that a traditional artisan--an artisan making a chair, for example--would have made, tested, and likely broken in real life. Using digital simulations of structural performance, however, *today we can make and break on the screen in a few hours more full-size trials than a traditional craftsman would have made and broken in a lifetime.* Good artisans of old could learn from their trials, and errors, over time, and intuit some shortcuts, fixes, or strategies; and so could we today, theoretically, using computational simulation; but in fact even that is no longer necessary. As there is no limit to the number of trials we can run, we can simply keep making and breaking (in simulation) all possible variations, randomly, until at some point we shall find one that does not break; and that will be the good one.

One may object that even that may take too long, and of course we have a solution for that, too: instead of doing many trials ourselves, one by one, we have programs that will run many trials in a sequence, then will look for the best results in that sequence, sort them based on some parameters we have chosen, then restart from that, *ad libitum atque ad infinitum*. This is what some call gradient-based optimization, also known as machine learning, deep learning, artificial neural networks, etc.; all of which more or less derive from, or relate to, the theory of genetic (generative, evolutionary) algorithms that was developed by John Holland in the mid 1970s. From a more general point of view, however, this is little more than massive, automated trial and error. No human would calculate anything that way, because it would take forever, by definition; and because it seems a bit dumb--but computers can do so many trials so fast, that, using advanced computation, massive trial and error becomes a viable computational strategy. In fact, that's the best computational strategy, because that's the only thing that computers really do.

Evidently, this is a far cry from how a modern engineer would have designed that structure we started this--which is one reason why no modern engineer ever designed it. A modern engineer would have started with a set of formulas establishing causal relationships between loads, forms, and stresses in the structure. By the causality they express, these formulas interpret and provide some understanding of the physical phenomena they describe.

But this is exactly what today's computers don't do. Computational optimization, as I described it, does not depend upon formulas, laws, or rules of causation: all validation comes from the authority of precedent--either real, or simulated.

This is how computers today predict things that modern science cannot calculate, and our mind cannot understand. In some cases, computers can already tell us what is going to happen, but they won't tell us why--because computers don't do that. Through computational form searching, we can already design new structures of unimaginable complexity. But precisely because it is unimaginable, this post-human complexity belies interpretation, and transcends the small-data logic of causality and determinism we have invented over time to simplify nature and convert it into reassuring, transparent, human-friendly models of causality. Why does one unimaginably complex structure stand up, and thousands very similar ones we just run through computational simulation don't? Who knows. Nobody knows it; least of all, its designers. And yet it does stands up; using digital simulations, we know in advance it will, which is why we can build it.

Prediction without causation means prediction without explanation. Not long ago, this would have been seen as black magic--and people doing that would have been burnt at the stake. Today, that is just the way computers work--and the way we must let them work, whether we like that or not, if we want to take advantage of their power. This is what I think we should call, at this point, a new kind of science. Would this be the second coming of the reactionary post-modern science that po-mo thinkers envisaged one generation ago, driven by their own anti-modern ideologies and anti-technical furor, and which would now be vindicated, ironically, by a new technological revolution that none of them had seen coming? Or would this be a last chance for redemption that is unexpectedly being given to modern rationality--assuming that any of it be left? Time will tell. A digital Sturm und Drang may not be around the corner, but there is thunder on the horizon, as well as dawn.