THE MYTH OF ARTIFICIAL INTELLIGENCE

...

Why Computers Can't Think the Way We Do

...

ERIK J. LARSON

Printed in the United States of America

First printing

Jacket design by Henry Sene Yee
Jacket art courtesy of Shutterstock

9780674259928 (EPUB)
9780674259935 (PDF)

The Library of Congress has cataloged the printed edition as follows:

Names: Larson, Erik J. (Erik John), author.
Title: The myth of artificial intelligence : why computers can't think the way we do / Erik J. Larson.
Description: Cambridge, Massachusetts : The Belknap Press of Harvard University Press, 2021. | Includes bibliographical references and index.
Identifiers: LCCN 2020050249 | ISBN 9780674983519 (cloth)
Subjects: LCSH: Artificial intelligence. | Intellect. | Inference. | Logic. | Natural language processing (Computer science) | Neurosciences.
Classification: LCC Q335 .L37 2021 | DDC 006.3—dc23
LC record available at https://lccn.loc.gov/2020050249

CONTENTS

INTRODUCTION

In the pages of this book you will read about the myth of artificial intelligence. The myth is not that true AI is possible. As to that, the future of AI is a scientific unknown. The myth of artificial intelligence is that its arrival is inevitable, and only a matter of time—that we have already embarked on the path that will lead to human-level AI, and then superintelligence. We have not. The path exists only in our imaginations. Yet the inevitability of AI is so ingrained in popular discussion—promoted by media pundits, thought leaders like Elon Musk, and even many AI scientists (though certainly not all)—that arguing against it is often taken as a form of Luddism, or at the very least a shortsighted view of the future of technology and a dangerous failure to prepare for a world of intelligent machines.

As I will show, the science of AI has uncovered a very large mystery at the heart of intelligence, which no one currently has a clue how to solve. Proponents of AI have huge incentives to minimize its known limitations. After all, AI is big business, and it's increasingly dominant in culture. Yet the possibilities for future AI systems are limited by what we currently know about the nature of intelligence, whether we like it or not. And here we should say it directly: all evidence suggests that human and machine intelligence are radically different. The myth of AI insists that the differences are only temporary, and that more powerful systems will eventually erase them. Futurists like

Ray Kurzweil and philosopher Nick Bostrom, prominent purveyors of the myth, talk not only as if human-level AI were inevitable, but as if, soon after its arrival, superintelligent machines would leave us far behind.

This book explains two important aspects of the AI myth, one scientific and one cultural. The scientific part of the myth assumes that we need only keep "chipping away" at the challenge of general intelligence by making progress on narrow feats of intelligence, like playing games or recognizing images. This is a profound mistake: success on narrow applications gets us not one step closer to general intelligence. The inferences that systems require for general intelligence—to read a newspaper, or hold a basic conversation, or become a helpmeet like Rosie the Robot in *The Jetsons*—cannot be programmed, learned, or engineered with our current knowledge of AI. As we successfully apply simpler, narrow versions of intelligence that benefit from faster computers and lots of data, we are not making incremental progress, but rather picking low-hanging fruit. The jump to general "common sense" is completely different, and there's no known path from the one to the other. No algorithm exists for general intelligence. And we have good reason to be skeptical that such an algorithm will emerge through further efforts on deep learning systems or any other approach popular today. Much more likely, it will require a major scientific breakthrough, and no one currently has the slightest idea what such a breakthrough would even look like, let alone the details of getting to it.

Mythology about AI is bad, then, because it covers up a scientific mystery in endless talk of ongoing progress. The myth props up belief in inevitable success, but genuine respect for science should bring us back to the drawing board. This brings us to the second subject of these pages: the cultural consequences of the myth. Pursuing the myth is not a good way to follow "the smart money," or even a neutral stance. It is bad for science, and it is bad for us. Why? One reason is

that we are unlikely to get innovation if we choose to ignore a core mystery rather than face up to it. A healthy culture for innovation emphasizes exploring unknowns, not hyping extensions of existing methods—especially when these methods have been shown to be inadequate to take us much further. Mythology about inevitable success in AI tends to extinguish the very culture of invention necessary for real progress—with or without human-level AI. The myth also encourages resignation to the creep of a machine-land, where genuine invention is sidelined in favor of futuristic talk advocating current approaches, often from entrenched interests.

Who should read this book? Certainly, anyone should who is excited about AI but wonders why it is always ten or twenty years away. There is a scientific reason for this, which I explain. You should also read this book if you think AI's advance toward superintelligence is inevitable and worry about what to do when it arrives. While I cannot prove that AI overlords will not one day appear, I can give you reason to seriously discount the prospects of that scenario. Most generally, you should read this book if you are simply curious yet confused about the widespread hype surrounding AI in our society. I will explain the origins of the myth of AI, what we know and don't know about the prospects of actually achieving human-level AI, and why we need to better appreciate the only true intelligence we know—our own.

IN THIS BOOK

In Part One, The Simplified World, I explain how our AI culture has simplified ideas about people, while expanding ideas about technology. This began with AI's founder, Alan Turing, and involved understandable but unfortunate simplifications I call "intelligence errors." Initial errors were magnified into an ideology by Turing's friend and statistician, I. J. Good, who introduced the idea of "ultraintelligence" as the predictable result once human-level AI had been achieved.

Between Turing and Good, we see the modern myth of AI take shape. Its development has landed us in an era of what I call technological kitsch—cheap imitations of deeper ideas that cut off intelligent engagement and weaken our culture. Kitsch tells us how to think and how to feel. The purveyors of kitsch benefit, while the consumers of kitsch experience a loss. They—we—end up in a shallow world.

In Part Two, The Problem of Inference, I argue that the only type of inference—thinking, in other words—that will work for human-level AI (or anything even close to it) is the one we don't have a clue how to program or engineer. The problem of inference goes to the heart of the AI debate because it deals directly with intelligence, in people or machines. Our knowledge of the various types of inference dates back to Aristotle and other ancient Greeks, and has been developed in the fields of logic and mathematics. Inference is already described using formal, symbolic systems like computer programs, so a very clear view of the project of engineering intelligence can be gained by exploring inference. There are three types. Classic AI explored one (deduction), modern AI explores another (induction). The third type (abduction) makes for general intelligence, and, surprise, no one is working on it—at all.[1] Finally, since each type of inference is distinct—meaning, one type cannot be reduced to another—we know that failure to build AI systems using the type of inference undergirding general intelligence will result in failure to make progress toward artificial general intelligence, or AGI.

In Part Three, The Future of the Myth, I argue that the myth has very bad consequences if taken seriously, because it subverts science. In particular, it erodes a culture of human intelligence and invention, which is necessary for the very breakthroughs we will need to understand our own future. Data science (the application of AI to "big data") is at best a prosthetic for human ingenuity, which if used correctly can help us deal with our modern "data deluge." If used as a replacement for individual intelligence, it tends to chew up invest-

Part I

THE SIMPLIFIED WORLD

Chapter 1

• • •

THE INTELLIGENCE ERROR

The story of artificial intelligence starts with the ideas of someone who had immense human intelligence: the computer pioneer Alan Turing.

In 1950 Turing published a provocative paper, "Computing Machinery and Intelligence," about the possibility of intelligent machines.[1] The paper was bold, coming at a time when computers were new and unimpressive by today's standards. Slow, heavy pieces of hardware sped up scientific calculations like code breaking. After much preparation, they could be fed physical equations and initial conditions and crank out the radius of a nuclear blast. IBM quickly grasped their potential for replacing humans doing calculations for businesses, like updating spreadsheets. But viewing computers as "thinking" took imagination.

Turing's proposal was based on a popular entertainment called the "imitation game." In the original game, a man and a woman are hidden from view. A third person, the interrogator, relays questions to one of them at a time and, by reading the answers, attempts to determine which is the man and which the woman. The twist is that the man has to try to deceive the interrogator while the woman tries to assist him—making replies from either side suspect. Turing replaced the man and woman with a computer and a human. Thus began what we now call the Turing test: a computer and a human receive typed

questions from a human judge, and if the judge can't accurately identify which is the computer, the computer wins. Turing argued that with such an outcome, we have no good reason to define the machine as unintelligent, regardless of whether it is human or not. Thus, the question of whether a machine has intelligence replaces the question of whether it can truly think.

The Turing test is actually very difficult—no computer has ever passed it. Turing, of course, didn't know this long-term result in 1950; however, by replacing pesky philosophical questions about "consciousness" and "thinking" with a test of observable output, he encouraged the view of AI as a legitimate science with a well-defined aim. As AI took shape in the 1950s, many of its pioneers and supporters agreed with Turing: any computer holding a sustained and convincing conversation with a person would be, most of us would grant, doing something that requires thinking (whatever that is).

TURING'S INTUITION / INGENUITY DISTINCTION

Turing had made his reputation as a mathematician long before he began writing about AI. In 1936, he published a short mathematical paper on the precise meaning of "computer," which at the time referred to a person working through a sequence of steps to get a definite result (like performing a calculation).[2] In this paper, he replaced the human computer with the idea of a machine doing the same work. The paper ventured into difficult mathematics. But in its treatment of machines it made no reference to human thinking or the mind. Machines can run automatically, Turing said, and the problems they solve do not require any "external" help, or intelligence. This external intelligence—the human factor—is what mathematicians sometimes call "intuition."

Turing's 1936 work on computing machines helped launch computer science as a discipline and was an important contribution to mathematical logic. Still, Turing apparently thought that his early definition missed something essential. In fact, the same idea of the mind or human faculties assisting problem-solving appeared two years later in his PhD thesis, a clever but ultimately unsuccessful attempt to bypass a result from the Austrian-born mathematical logician Kurt Gödel (more on this later). Turing's thesis contains this curious passage about intuition, which he compares with another mental capability he calls ingenuity:

> Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two faculties, which we may call intuition and ingenuity. The activity of the intuition consists in making spontaneous judgments which are not the result of conscious trains of reasoning. These judgments are often but by no means invariably correct (leaving aside the question as to what is meant by "correct"). Often it is possible to find some other way of verifying the correctness of an intuitive judgment. One may for instance judge that all positive integers are uniquely factorable into primes; a detailed mathematical argument leads to the same result. It will also involve intuitive judgments, but they will be ones less open to criticism than the original judgment about factorization. I shall not attempt to explain this idea of "intuition" any more explicitly.

Turing then moves on to explain ingenuity: "The exercise of ingenuity in mathematics consists in aiding the intuition through suitable arrangements of propositions, and perhaps geometrical figures or drawings. It is intended that when these are really well arranged the validity of the intuitive steps which are required cannot seriously be doubted."[3]

system has limits, at any rate. It cannot prove in its own language something that is true. In other words, we can see something that the computer cannot.[5]

Gödel's result dealt a massive blow to a popular idea at the time, that all of mathematics could be converted into rule-based operations, cranking out mathematical truths one by one. The zeitgeist was formalism—not talk of minds, spirits, souls, and the like. The formalist movement in mathematics signaled a broader turn by intellectuals toward scientific materialism, and in particular, logical positivism—a movement dedicated to eradicating traditional metaphysics like Platonism, with its abstract Forms that couldn't be observed with the senses, and traditional notions in religion like the existence of God. The world was turning to the idea of precision machines, in effect. And no one took up the formalist cause as vigorously as the German mathematician David Hilbert.

HILBERT'S CHALLENGE

At the outset of the twentieth century (before Gödel), David Hilbert had issued a challenge to the mathematical world: show that all of mathematics rested on a secure foundation. Hilbert's worry was understandable. If the purely formal rules of mathematics can't prove any and all truths, it's at least theoretically possible for mathematics to disguise contradictions and nonsense. A contradiction buried somewhere in mathematics ruins everything, because from a contradiction anything can be proven. Formalism then becomes useless.

Hilbert expressed the dream of all formalists, to prove finally that mathematics is a closed system governed only by rules. Truth is just "proof." We acquire knowledge by simply tracing the "code" of a proof and confirming no rules were violated. The larger dream, thinly disguised, was really a worldview, a picture of the universe as itself a mechanism. AI began taking shape as an idea, a philosophical posi-

tion that might also be proven. Formalism treated intelligence as a rule-based process. A machine.

Hilbert issued his challenge at the Second International Congress of Mathematicians in Paris in 1900. The intellectual world was listening. His challenge had three main parts: to prove that mathematics was complete; to prove that mathematics was consistent; and to prove that mathematics was decidable.

Gödel dealt the first and second parts of Hilbert's challenge a death blow with the publication of his incompleteness theorems in 1931. The question of decidability was left unanswered. A system is decidable if there is a definite procedure (a proof, or sequence of deterministic, obvious steps) to establish whether any statement constructed using the rules of the system is true or false. The statement $2+2=4$ must be True, and $2+2=5$ must be False. And so for all statements that one can validly make using the symbols and rules of the system. Since arithmetic was thought to be the foundation of mathematics, proving mathematics was decidable amounted to proving the result for arithmetic and its extensions. This would amount to saying that mathematicians, playing a "game" with rules and symbols (the formalist idea), were in fact playing a valid game that never led to contradiction or absurdity.

Turing was fascinated with Gödel's result, which demonstrated not the power of formal systems but rather their limitations. He took up work on the remaining part of Hilbert's challenge, and began thinking in earnest about whether a decision procedure for formal systems might exist. By 1936, in his paper "Computable Numbers," he proved that it must not. Turing realized that Gödel's use of self-reference also applied to questions about decision procedures or, in effect, computer programs. In particular, he realized that there must exist (real) numbers that *no* definite method could "calculate," by writing out their decimal expansion, digit by digit. He imported a result from the nineteenth-century mathematician Georg Cantor, who

proved that real numbers (those with a decimal expansion) were more numerous than the integers, even though real numbers and integers are both infinite. Turing stood on the shoulders of giants, perhaps. But in the end, his work in "Computable Numbers" proved again a negative. It was a limiting result: no universal decision procedure was possible. In other words, rules—even in mathematics—aren't enough. Hilbert was wrong.[6]

IMPLICATIONS FOR AI

What is important to AI here is this: Turing disproved that mathematics was decidable by inventing a machine, a deterministic machine, requiring no insight or intelligence to solve problems. Today, we refer to his abstract formulation of a machine as a Turing machine. I am typing on one right now. Turing machines are computers. It is one of the great ironies of intellectual history that the theoretical framework for computation was put in place as a side-thought, a means to another end. While working to disprove that mathematics itself was decidable, Turing first invented something precise and mechanical, the computer.

In his 1938 PhD thesis, Turing hoped that formal systems might be extended by including additional rules (then sets of rules, and sets of sets of rules) that could handle the "Gödel problem." He discovered, rather, that the new, more powerful system would have a new, more complicated Gödel problem. There was no way around Gödel's incompleteness. Buried in the complexities of Turing's discussion of formal systems, however, is an odd suggestion, relevant to the possibility of AI. Perhaps the faculty of intuition cannot be reduced to an algorithm, to the rules of a system?

Turing wanted to find a way out of Gödel's limiting result in his 1938 thesis, but he discovered that this was impossible. Instead, he switched gears, exploring how, as he put it, to "greatly reduce" the re-

quirement of human intuition when doing calculations. His thesis considered the powers of ingenuity, by creating ever more complicated systems of rules. (Ingenuity, it turned out, could become universal—there are machines that can take as input other machines, and thus run all the machines that can be built. This insight, technically a universal Turing machine and not a simple Turing machine, was to become the digital computer.) But in his formal work on computing, Turing had (perhaps inadvertently) let the cat out of the bag. By allowing for intuition as distinct from and outside of the operations of a purely formal system like a computer, Turing in effect suggested that there may be differences between computer programs that do math and mathematicians.

It was a curious turn, therefore, that Turing made from his early work in the 1930s to the more wide-ranging speculation about the possibility of intelligent computers in "Computing Machinery and Intelligence," published a little over a decade later. By 1950, discussion of intuition disappeared from Turing's writings about the implications of Gödel. His interests turned, in effect, to the possibility that computers might become "intuition-machines" themselves. In essence, he decided that Gödel's result didn't apply to the question of AI: if we humans are highly advanced computers, Gödel's result means only that there are some statements that we cannot understand or see to be true, just as with less complicated computers. The statements might be fantastically complicated and interesting. Or, possibly, they might be banal yet overwhelmingly complex. Gödel's result left open the question of whether minds were just very complicated machines, with very complicated limitations.

Intuition, in other words, had become part of Turing's ideas about machines and their powers. Gödel's result couldn't say (to Turing, anyway) whether minds were machines or not. On the one hand, incompleteness says that some statements can be seen to be true using intuition, but cannot be proved by a computer using ingenuity. On

the other hand, a more powerful computer can use more axioms (or more bits of relevant code) and prove the result—thus showing that intuition is not beyond computation for that problem. This becomes an arms race: more and more powerful ingenuity substituting for intuition on more and more complicated problems. No one can say who wins the race, so no one can make a case—using the incompleteness result—about the inherent differences between intuition (mind) and ingenuity (machine). But as Turing no doubt knew, if this were true, then so too was at least the possibility of artificial intelligence.

Thus, between 1938 and 1950, Turing had a change of heart about ingenuity and intuition. In 1938, intuition was the mysterious "power of selection" that helped mathematicians decide which systems to work with and what problems to solve. Intuition was not something in the computer. It was something that decided things about the computer. In 1938, Turing thought intuition wasn't part of any system, which suggested not only that minds and machines were fundamentally different but that AI-as-human-thinking was well-nigh impossible.

Yet by 1950 he had reversed his position. With the Turing test, he offered a challenge for skeptics and a sort of defense of intuition in machines, asking in effect: Why not? This was a radical about-face. A new view of intelligence, it seemed, was taking shape.

Why the shift? Something outside the world of strict mathematics and logic and formal systems had happened to Turing between 1938 and 1950. It had happened, in fact, to all of Great Britain, and indeed to most of the world. What happened was the Second World War.

The codes were generated by a typewriter-looking device known as the Enigma, a kind of machine that had been in commercial use since the 1920s but that the Germans had strengthened significantly for use in the war. Modified Enigmas were used for all strategic communications in the Nazi war effort. The Luftwaffe, for instance, used the Enigma machine in its conduct of the air war, as did the Kriegsmarine in its naval operations. Messages encrypted with the modified Enigma were widely thought to be undecipherable.

Turing's role in Bletchley and his subsequent rise to national hero after the war is a story that has been told many times. (In 2014, the major motion picture *The Imitation Game* dramatized his work at Bletchley, as well as his subsequent role in developing computers.) Turing's major breakthrough was, by pure mathematical standards, relatively uninteresting because it exploited an old idea from deductive logic. The method that he and others half-jokingly referred to as "Turingismus" involved eliminating large numbers of possible solutions to Enigma codes by finding combinations with contradictions. Contradictory combinations are impossible; we cannot have both "A" and "not-A" in some logical system, just as we cannot be both "at the store" and "at home" at the same time. Turingismus was a winning idea, and became a huge success at Bletchley. It did what was required of the "boy geniuses" sequestered in the think tank by speeding up the task of decrypting Enigma messages. Other scientists devised different strategies for cracking the codes at Bletchley.[4] Ideas were tested on a machine called a Bombe—its tongue-in-cheek name borrowed from a predecessor machine in Poland, the Bomba, and possibly inspired by the small noise made when a calculation was finished. Think of the Bombe as a proto-computer, capable of running different programs.

The advantage in war swung from the Axis to Allied powers by 1943 or thereabouts, in no small part because of the sustained effort of the Bletchley code-crackers. The team was a celebrated success, and its members became war heroes. Careers were made. Bletchley,

meanwhile, also proved a haven for thinking about computation: Bombes were machines, and they ran programs to solve problems that humans, by themselves, could not.

INTUITIVE MACHINES? NO.

For Turing, Bletchley played a major role in crystallizing his ideas about the possibility of intelligent machines. Like his colleagues Jack Good and Claude Shannon, Turing saw the power and utility of their "brain games" as cryptanalysts during the war: they could decipher messages that were otherwise completely opaque to the military. The new methods of computation were not just interesting for considering automated chess-playing. Computation could, quite literally, sink warships.

Turing was thinking about an abstraction (yet again): minds and machines, or the general idea of intelligence. But there was something odd about his view of what it meant. In the 1940s, intelligence was a trait not typically attributed to formal systems like the purely mechanical code-breaking Bombes of Bletchley. Gödel had demonstrated that, in general, truth cannot be reduced to formality, as in playing a formal game with a set sequence of rules—but recall that his proof left open the question of whether specific machines might actually incorporate the intuition that minds use to make choices about rules to follow, even if no supreme system could exist that could prove everything (which Gödel had shown so definitely in 1931).

After Bletchley, Turing turned increasingly to the question of whether powerful machines could be built that used intuition and ingenuity. The vast number of possible combinations to check to decipher German codes swamped human intuition. But systems with the right programs could accomplish the task by simplifying such vast mathematical possibilities. To Turing, this suggested that intuition could be embodied in machines. In other words, the success at Bletchley implied that perhaps an artificial intelligence could be built.

To make sense of his line of thought, however, some particular idea about "intelligence" had to be settled on. Intelligence as displayed by humans needed to be reducible—analyzable—in terms of the powers of a machine. In essence, intelligence had to be reducible to problem-solving. That is what playing chess is, after all, and that is what breaking a code is, as well.

And here we have it: Turing's great genius, and his great error, was in thinking that human intelligence reduces to problem-solving. Whether or not the ideas about intelligent machines in his 1950 "Computing Machinery and Intelligence" became explicit in the war years, it is clear that his experience at Bletchley crystallized his later view of AI, and it is clear that AI in turn followed closely and without necessary self-analysis precisely in his path.

But a closer look at the Bletchley code-cracking success immediately reveals a dangerous simplification in the philosophical ideas about man and machine. Bletchley was an intelligent system—a coordination of military efforts (including spying and espionage, as well as capture of enemy vessels), social intelligence between the military and the various scientists and engineers at Bletchley, and (as with all of life) sometimes sheer dumb luck. In truth, as a practical reality, the German-modified Enigma was unbreakable by purely mechanical means. The Germans knew this based on mathematical arguments about the difficulty of mechanical deciphering. Part of Bletchley's success was, ironically, the stubborn confidence of Nazi commanders in the impregnability of the Enigma ciphers—thus they fail at crucial times to modify or strengthen the machines after discovering certain ciphers had been cracked, blaming covert spying operations rather than scientific defeat. But the fog of war mixes together not just new technologies but new forms of human and social intelligence. War is not chess.

Early in the war, for instance, Polish forces had recovered important fragments of Enigma communications that later provided invaluable clues to Bletchley efforts. The Poles had used these fragments

(along with others from Russian sources) to develop their own, simpler Bomba as early as 1938. Turing's much improved version in the early months of 1940—the Bombe using his "Turingismus"—relied on the early work the Poles made possible by events on the battlefield. Turing, too, would see his own design improved in response to improvements in the Enigma by his colleague Gordon Welchman, by which a "diagonal board" was added to further simplify the search for contradictions.[5] Here were two human minds, using intuition, working together socially.

More events in the theater of war proved vitally important. Off the shores of Norway, a British aircraft carrier was sunk on June 8, 1940. The attack provided the location of German U-boats, albeit at the heavy cost of many sailors left at the bottom of the sea. Just weeks before, in late April 1940, the German patrol boat *VP2623*, a particularly devastating member of the fleet, was captured with a trove of Enigma evidence inside. The necessary pieces of the Enigma puzzle were getting into Allied hands, and finding their way to the Bletchley group.

These bits and pieces by themselves were grossly inadequate for quick deciphering of future German communications, amounting to what one Turing biographer called "guesswork" for Bletchley cryptanalysts. But they facilitated an all-important first step in figuring out how to program the Bombes. Turing and colleagues called it the "weight of evidence," borrowing a term coined by the American scientist and logician C. S. Peirce (who is prominently featured in Part Two of this book).[6]

Weight of evidence can be understood by mathematicians in different ways, but for Bletchley's success (and for larger issues regarding AI) it amounts to the application of informed guesses, or intuition, to give direction to ingenuity, or machines. A scrap of deciphered text recovered from a captured U-boat could mean anything, just as a white ball found near a bag of white balls could mean anything, but in each case, we can make intelligent guesses to understand what hap-

pened. We think the ball is very likely from the bag, even though we didn't see it taken out. Still, it's a guess. Guesses of this sort can't be proven true, but the better human intuition does at setting initial conditions for devising mechanical procedures, the better chance those procedures have of terminating on desired outcomes, rather than, say, running on aimlessly in false or misleading directions. Weight of evidence—guessing—made Bombes work.

Bletchley scientists were not merely feeding information into Bombes, leaving them to do the tireless and important work of eliminating millions of incorrect codes or ciphers. To be sure, the Bombes were necessary—this is what Turing saw so clearly, and what no doubt suffused his imagination with the possibility that his "mechanical procedures" could reproduce or supersede human intelligence. But the fact is that the Bletchley group was first and foremost engaged in guesswork. They were forming hypotheses by recognizing the clues hidden in the patchwork of scraps of instructions, ciphers, and messages coming in from the battlefield. Guessing is known in science as forming hypotheses (a term Charles Sanders Peirce also used), and it is absolutely fundamental to the advancement of human knowledge. Small wonder then that the Bletchley effort amounted to a system of guessing well. Its sine qua non was not mechanical but rather what we might call initial intelligent observation. The Bombes had to be pointed at something, and then set on their course.

In line with a theme we will explore in Part Two, Peirce had recognized early on, by the late nineteenth century, that every observation that shapes the complex ideas and judgments of intelligence begins with a guess, or what he called an abduction:

> Looking out of my window this lovely spring morning I see an azalea in full bloom. No, no! I do not see that; though that is the only way I can describe what I see. That is a proposition, a sentence, a fact; but what I perceive is not proposition, sentence,

error and, further, one that has been passed down through generations of AI scientists, right up to the present day.

TURING'S INTELLIGENCE ERROR AND NARROW AI

The problem-solving view of intelligence helps explain the production of invariably narrow applications of AI throughout its history. Game playing, for instance, has been a source of constant inspiration for the development of advanced AI techniques, but games are simplifications of life that reward simplified views of intelligence. A chess program plays chess, but does rather poorly driving a car. IBM's Watson system plays *Jeopardy!,* but not chess or Go, and massive programming or "porting" efforts are required to use the Watson platform to perform other data mining and natural language processing functions, as with recent (and largely unsuccessful) forays into health care.

Treating intelligence as problem-solving thus gives us narrow applications. Turing no doubt knew this, and speculated in his 1950 paper that perhaps machines could be made to learn, thus overcoming the constraints that are a natural consequence of designing a computer system narrowly to solve a problem. If machines could learn to become general, we would witness a smooth transition from specific applications to general thinking beings. We would have AI.

What we now know, however, argues strongly against the learning approach suggested early on by Turing. To accomplish their goals, what are now called machine learning systems must each learn something specific. Researchers call this giving the machine a "bias." (This doesn't carry the negative connotation it does in the broader social world; it doesn't mean that the machine is pigheaded or difficult to argue with, or has an agenda in the usual sense of the word.) A bias in machine learning means that the system is designed and tuned to learn something. But this is, of course, just the problem of producing

narrow problem-solving applications. (This is why, for example, the deep learning systems used by Facebook to recognize human faces haven't also learned to calculate your taxes.)

Even worse, researchers have realized that giving a machine learning system a bias to learn a particular application or task means it will do more poorly on other tasks. There is an inverse correlation between a machine's success in learning some one thing, and its success in learning some other thing. Even seemingly similar tasks are inversely related in terms of performance. A computer system that learns to play championship-level Go won't also learn to play championship-level chess. The Go system has been specifically designed, with a particular bias toward learning the rules of Go. Its learning curve, as they call it, thus follows the known scoring of that particular game. Its learning curve regarding some other game, say *Jeopardy!* or chess, is useless—in fact, nonexistent.

Machine learning bias is typically understood as a source of learning error, a technical problem. (It has also taken on the secondary meaning, hewing to ordinary language usage, of producing results that are unintentionally and unacceptably weighted by, say, race or gender.) Machine learning bias can introduce error simply because the system doesn't "look" for certain solutions in the first place. But bias is actually necessary in machine learning—it's part of learning itself.

A well-known theorem called the "no free lunch" theorem proves exactly what we anecdotally witness when designing and building learning systems. The theorem states that any bias-free learning system will perform no better than chance when applied to arbitrary problems. This is a fancy way of stating that designers of systems must give the system a bias deliberately, so it learns what's intended. As the theorem states, a truly bias-free system is useless. There are complicated techniques, like "pre-training" on data using unsupervised methods that expose the features of the data to be learned. All of this is part and parcel of successful machine learning. What's left out

of the discussion, however, is that tuning a system to learn what's intended by imparting to it a desired bias generally means causing it to become narrow, in the sense that it won't then generalize to other domains. Part of what it means to build and deploy a successful machine learning system is that the system is not bias-free and general but focused on a particular learning problem. Viewed this way, narrowness is to some extent baked in to such approaches. Success and narrowness are two sides of the same coin.

This fact alone casts serious doubt on any expectation of a smooth progression from today's AI to tomorrow's human-level AI. People who assume that extensions of modern machine learning methods like deep learning will somehow "train up," or learn to be intelligent like humans, do not understand the fundamental limitations that are already known. Admitting the necessity of supplying a bias to learning systems is tantamount to Turing's observing that insights about mathematics must be supplied by human minds from outside formal methods, since machine learning bias is determined, prior to learning, by human designers.[10]

TURING'S LEGACY

To sum up the argument, the problem-solving view of intelligence necessarily produces narrow applications, and is therefore inadequate for the broader goals of AI. We inherited this view of intelligence from Alan Turing. (Why, for instance, do we even use the term artificial intelligence, rather than, perhaps, speaking of "human-task-simulation"?)[11] Turing's great genius was to clear away theoretical obstacles and objections to the possibility of engineering an autonomous machine, but in so doing he narrowed the scope and definition of intelligence itself. It is no wonder, then, that AI began producing narrow problem-solving applications, and it is still doing so to this day.

Turing, again, disliked viewing thinking or intelligence as something social or situational. Yet despite his proclivities to see human intelligence as an individual mechanical process—ushering in untold media references to the "mechanical brain" as early computers appeared in the 1940s—it is obvious that talk of intelligence always involves, as it must necessarily involve, situating it in a broader context. General (non-narrow) intelligence of the sort we all display daily is not an algorithm running in our heads, but calls on the entire cultural, historical, and social context within which we think and act in the world. AI would hardly have moved forward if developers had embraced such a large and complicated understanding of intelligence—that is true enough. At the same time, as a result of Turing's simplification, we've ended up with narrow applications, and we have no reason to expect general ones without a radical reconceptualization of what we mean by AI.

Turing anticipated some of these difficulties in his 1950 paper by suggesting that machines might be made to learn. What we now know, however (contra recent excitement about machine learning), is that learning itself is a kind of problem-solving, made possible only by introducing a bias into the learner that simultaneously makes possible the learning of a particular application, while reducing performance on other applications. Learning systems are actually just narrow problem-solving systems, too. Given that there is no known theoretical bridge from such narrow systems to general intelligence of the sort displayed by humans, AI has fallen into a trap. Early errors in understanding intelligence have led, by degrees and inexorably, to a theoretical impasse at the heart of AI.

Consider again Turing's original distinction between intuition and ingenuity. The question of AI for him was whether intuition—that which is supplied by the designer of a system—could in fact be "pulled into" the formal part of the system (the ingenuity machine),

thus making a system capable of escaping the curse of narrowness by using intuition to choose its own problems—to grow smarter and to learn. So far, no one has done this with any computer. No one even has the slightest clue how this would work. We do know that designers use intuition outside AI systems to tell such systems what specific problems to solve (or to learn to solve). The question of systems using intuition autonomously goes straight to the core of what I will call the Problem of Inference, to which we will turn in Part Two.

There will also be much more to say about the "narrowness trap" of AI in Part Two. First, however, there is more ground to cover in this part. We will turn next to superintelligence, another intelligence error, and a natural extension of the first.

sible and therefore not requiring further explanation. But it does; we do need to understand the "how."

If we suppose a simple enhancement like superior hardware, the proposal is too trivial and silly to entertain further. Even a stalwart believer in inexorable progress like Ray Kurzweil isn't likely to reduce intelligence that far—we don't think adding RAM to a MacBook makes it (really and truly) more intelligent. The device is now faster, and can load bigger applications, and so on. But if intelligence means anything interesting, it must be more complicated than loading applications faster. This harder part of intelligence is left unsaid.

Or suppose we borrow language from the biological world (as AI so often does), and then confidently declare that computational capability doesn't devolve, it evolves. Looking deeper, we see that this argument is plagued once again by an inadequate and naive view of intelligence. The problem—a glaring omission—is that we have no evidence in the biological world of anything intelligent ever designing a more intelligent version of itself. Humans are intelligent, but in the span of human history we have never constructed more intelligent versions of ourselves.

A precondition for building a smarter brain is to first understand how the ones we have are cognitive, in the sense that we can imagine scenarios, entertain thoughts and their connections, find solutions, and discover new problems. Things occur to us; we reason through our observations and what we already know; answers pop into our heads. All of this buzz of biological magic remains opaque, its "processing" still vastly uncharted. And yet, we have been contemplating and investigating our thinking processes and brain functions for millennia.

Why should a generally intelligent machine suddenly have insight into its own global cognitive capacities, when we clearly do not? And even if it did, how could the machine use this knowledge to make itself smarter?

This is not a matter of self-improvement. We can, for instance, make ourselves more intelligent by reading books or going to school; educating ourselves makes possible further intellectual development, and so on. All of this is uncontroversial. And none of it is the point. One major problem with assumptions about increases in intelligence in AI circles is the problem of circularity: it takes (seemingly general) intelligence to increase general intelligence. A closer look reveals no linear progression, but only mystery.

VON NEUMANN AND SELF-REPRODUCING MACHINES

Good introduced the idea of self-improving AIs leading to ultraintelligence in the mid-1960s, but nearly two decades earlier John Von Neumann had considered the idea and rejected it. In a 1948 talk at the Institute for Advanced Studies at Princeton, Von Neumann explained that, while human reproduction often improves on prior "designs," it's clear that machines tasked with designing new and better machines face a fundamental stumbling block, since any design for a new machine must be specified in the parent machine. The parent machine would then necessarily be more complex, not less, than its creation: "An organization which synthesizes something is necessarily more complicated, of a higher order, than the organization it synthesizes," he said.[3]

In other words, Von Neumann pointed to a fundamental difference between organic life as we know it, and the machines we build. Jack Good's prediction of ultraintelligence was a bit of science fiction.

Von Neumann theorized that a self-reproducing machine would need, at minimum, eight parts, including a "stimulus organ," a "fusing organ" to connect parts together, a "cutting organ" to sever connections, and a "muscle" for motion. He then sketched plausible mechanisms for cognitive improvements including a randomizing element, akin to biological mutation, to allow for the necessary modifications. But Von Neumann thought that, rather than advance the machine's

thinking, such random mutations were more likely to "devolve" desired functions and capacities. The most probable outcome was nonfunction, the equivalent of a lethal change: "So, while this system is exceedingly primitive, it has the trait of an inheritable mutation, even to the point that a mutation made at random is probably lethal, but may be non-lethal and inheritable."

For the machines to get something better, essentially smarter, from their designs they would need a creative element added to their stimulus and fusing organs. Unlike biological evolution, the idea wasn't to wait around millions of years, but to require of parent systems the necessary Promethean spark in themselves, leading more or less directly to better designs. This was fiction, thought Von Neumann. As he explained to his colleagues at Princeton, no science or engineering theories could make sense of it. Von Neumann, no Luddite, was exploding the "intelligence explosion."

One obvious flaw in predictions of an intelligence explosion leading to superintelligence is that we already have human-level intelligence—we are human. By Good's logic, we should then be capable of designing something better than human. This is just a restatement of the goals of the field of AI, so we are getting trapped in a circle. The humans who are AI researchers already know it's a mystery how to design smarter artifacts, just as Von Neumann explained. Transferring this mystery from our own intelligence to an envisioned machine's doesn't help. To unpack this more, consider a genius AI researcher we'll call Alice.

INTELLIGENCE EXPLOSIONS, THE VERY IDEA

Let's suppose Alice is an AI scientist who has a dull neighbor, Bob. Bob has common sense, can read the newspaper, and can carry on a conversation (although perhaps it's boring), so he's worlds ahead of the best AI systems coming out of Google's DeepMind.

Alice works for an amazing new startup (soon to be acquired by Google), and wants to build an AI that is as smart as Bob. She's sketched out two systems, in the spirit of Daniel Kahneman's well-known System 1 and System 2.[4] They are intuition pumps or metaphors that give a rough blueprint for the types of problems needing to be solved to get to artificial general intelligence. In Alice's context, we'll call these System X, for competence on well-defined tasks like game play (as in chess or Go) and System Y, for general intelligence. The latter system includes Bob's competence at reading and conversation, but also the murkier area of novel ideas and insights.

Bob is terrible at chess, and in fact his X system is pathetic compared not only to a system like AlphaGo but also to many other humans. His short-term memory is worse than most people's; he scores poorly on IQ tests; and he struggles with crossword puzzles. As for his Y system, his general intelligence shows a conspicuous lack of interest in or ability at novel or insightful thinking. Bob is not the kind of neighbor that gets many invitations to dinner parties.

Alice's strategy is first to design a Bob-Machine that matches Bob's intelligence. She reasons that if she succeeds in creating a Bob-Machine, that machine can design a smarter version of itself, leading eventually to an intelligence explosion. Now, again, keep in mind that designing a Bob-Machine is no easy task, because Bob has a System Y—which means he has solved the problem of commonsense reasoning and has general cognitive abilities. He can pass a Turing test, for one. And he can read children's stories and the sports section and summarize them. Bob therefore blows away Google's best natural language understanding systems, like Ray Kurzweil's Talk to Books semantic search tool. This is why Alice is excited about her Bob-Machine project; it would be a huge advance in AI.

The question is: how to get there? Alice's first approach is to maximize the Bob-Machine's System X capabilities. She gives it a computer memory and access to the web via Google. Unfortunately, this ver-

sion of the Bob-Machine quickly proves Stuart Russell's point that supercomputers without real intelligence just get to wrong answers more quickly.[5] The Bob-Machine remembers the wrong things and fails to ask the right questions. All the improvements on the System X side just make the machine more competent at recalling and coughing up crackpot theories and making pronouncements about the world with more facts, all misused and poorly understood from a System Y perspective. Sure, the Bob-Machine plays flawless chess, but its chess competence makes it less interesting to Alice, who realizes that the machine she has created has no hope of designing a "more intelligent" version of itself.

In an *aha!* moment, though, Alice realizes that Bob himself couldn't design a smarter version of himself. So how could the Bob-Machine? The problem, she thinks, is that System X optimization does not supply resources to System Y of the necessary kind. The Bob-Machine (like Bob himself) has to see its own intelligence as something of a certain quantity, assess how it is limited and to what extent, and then actively redesign itself so as to become smarter in the important and relevant ways. But this is precisely the way in which the Bob-Machine (like Bob) is unintelligent! The Bob-Machine *can't* do this, because it lacks these System Y capabilities for insight, discovery, and innovation. Alice must go back to the drawing board.

Alice then decides that the Bob-Machine is just too stupid to be part of a bootstrapping process to superintelligence. (In a moment of sheer panic, it occurs to her that this logic jeopardizes the entire enterprise of getting to superintelligence, but she manages to suppress this concern quickly.) Alice decides, in deference to AI's founder and to the eager-beaver marketing department of her company, Ultra++, that she'll instead focus on designing a machine as intelligent as Alan Turing, called the Turing-Machine.

Now, assuming that Turing was smarter than Alice (though who is to say?), she can't just design a Turing-Machine directly, and anyway

scientists from the responsibility of needing to make scientific breakthroughs or develop revolutionary ideas. Artificial intelligence just evolves, like we did. We can call the futurists and AI believers in this camp evolutionary technologists, or ETs.

The ET view is popular among new age technologists like *Wired* cofounder Kevin Kelly, who argues in his 2010 book *What Technology Wants* that AI won't come about as the work of a "mad scientist," but simply as an evolutionary process on the planet, much like natural evolution.[6] According to this view, the world is becoming "intelligenized" (Kelly's word), and more and more complex and intelligent forms of technology are emerging without explicit human design.[7] Such thinkers also might envision the World Wide Web as a giant, growing brain. Humans, in this view, become a link in a cosmic historical chain that reaches into the future to true AI, where we get left behind or assimilated.

Organic life evolves extremely slowly, but ETs view technological progress as accelerating. As Kurzweil famously argues, technology is getting more complicated on an accelerating curve, according to a law he thinks is discernible in history, the Law of Accelerating Returns. Thus, human-level intelligence and then superintelligence will emerge on the planet in drastically short timeframes, as compared to organic evolution. In decades or even years, we will be confronted with them.

This is a simple, tidy story of humanity. We are transitioning to something else, which will be smarter and better.

Notice that the story is not testable; we just have to wait around and see. If the predicted year of true AI's coming is false, too, another one can be forecast, a few decades into the future. AI in this sense is unfalsifiable and thus—according to the accepted rules of the scientific method—unscientific.

Note that I'm not saying that true AI is impossible. As Stuart Russell and other AI researchers like to point out, twentieth-century

scientists such as Ernest Rutherford thought that building an atomic bomb was impossible, but Leo Szilard figured out how nuclear chain reactions work—a mere twenty-four hours after Rutherford pronounced the idea dead.[8] It's a good reminder not to bet against science. But note that nuclear chain reactions grew from scientific theories that were testable. Theories about mind power evolving out of technology aren't testable.

The claims of Good and Bostrom, presented as scientific inevitability, are more like imagination pumps: just think if this could be! And there's no doubt, it would be amazing. Perhaps dangerous. But imagining a what-if scenario stops far short of serious discussion about what's up ahead.

For starters, a general superintelligence capability must be connected to the broader world in such a way that it can observe and "guess" more productively than we do. And if intelligence is also social and situational, as it seems it must be, then an immense amount of contextual knowledge is required to engineer something more intelligent. Good's problem is not narrow and mechanical, but rather pulls into its orbit the whole of culture and society. Where is the barest, even remotely plausible blueprint for this?

Good's proposal, in other words, is based once again on an inadequate and simplified view of intelligence. It presupposes the original intelligence error, and adds to it yet another reductive sleight of hand: that an individual mechanical intelligence can design and construct a greater one. That a machine would be situated at such an Archimedean point of creation seems implausible, to put it mildly. The idea of superintelligence is in reality a multiplication of errors, and it represents in barest form the extension of the fantasy about the rise of AI.

To dig deeper into all of this, we should push further into this fantasy. It's called the *Singularity*, and we turn to it next.

Chapter 4

• • •

THE SINGULARITY, THEN AND NOW

In the 1950s, the mathematician Stanislaw Ulam recalled an old conversation with John Von Neumann, in which Von Neumann discussed the possibility of a technological turning point for humanity: "the ever accelerating progress of technology . . . gives the appearance of approaching some essential singularity in the history of the race beyond which human affairs, as we know them, could not continue."[1]

Von Neumann likely made this comment as digital computers were arriving on the technological scene. But digital computers were the latest innovation in a long and seemingly unbroken sequence of technologies.[2] By the 1940s, it had become clear that the scientific and industrial revolutions of the past three hundred years had set in motion forces of immense, symbiotic power: the fruits of new science seeded the development of new technology, which in turn made possible more scientific discovery. For example, science gave us the telescope, which in turn improved astronomy.

Inextricably tied to changes in science and technology was social change—rapid, chaotic at times, and seemingly irreversible. City populations exploded (with considerable doses of squalor and injustice), and entirely new forms of social and economic organization emerged, seemingly overnight. Steam engines revolutionized transportation, as did, later, internal combustion engines. Trains, trolleys,