

## Prologue: Terrified

Computers seem to be getting smarter at an alarming rate, but one thing they still can't do is appreciate irony. That's what was on my mind a few years ago, when, on my way to a discussion about artificial intelligence (AI), I got lost in the capital of searching and finding—the Googleplex, Google's world headquarters in Mountain View, California. What's more, I was lost inside the Google Maps building. Irony squared.

The Maps building itself had been easy to find. A Google Street View car was parked by the front door, a hulking appendage crowned by a red-and-black soccer ball of a camera sticking up from its roof. However, once inside, with my prominent "Visitor" badge assigned by security, I wandered, embarrassed, among warrens of cubicles occupied by packs of Google workers, headphones over ears, intently typing on Apple desktops. After some (map-less) random search, I finally found the conference room assigned for the daylong meeting and joined the group gathered there.

The meeting, in May 2014, had been organized by Blaise Agüera y Arcas, a young computer scientist who had recently left a top position at Microsoft to help lead Google's machine intelligence effort. Google started out in 1998 with one "product": a website that used a novel, extraordinarily successful method for searching the web. Over the years, Google has evolved into the world's most important tech company and now offers a vast array of products and services, including Gmail, Google Docs, Google Translate, YouTube, Android, many more that you might use every day, and some that you've likely never heard of.

Google's founders, Larry Page and Sergey Brin, have long been motivated by the idea of creating artificial intelligence in computers, and this quest has become a major focus at Google. In the last decade, the company has hired a profusion of AI experts, most notably Ray Kurzweil, a well-known inventor and a controversial futurist who promotes the idea of an AI Singularity, a time in the near future when computers will become smarter than humans. Google hired Kurzweil to help realize this vision. In 2011, Google created an internal AI research group called Google Brain; since then, the company has also acquired an impressive array of AI start-up companies with equally optimistic names: Applied Semantics, DeepMind, and Vision Factory, among others.

In short, Google is no longer merely a web-search portal—not by a long shot. It is rapidly becoming an applied AI company. AI is the glue that unifies the diverse products, services, and blue-sky research efforts offered by Google and its parent company, Alphabet. The company's ultimate aspiration is reflected in the original mission statement of its DeepMind group: "Solve intelligence and use it to solve everything."<sup>1</sup>

### AI and *GEB*

I was pretty excited to attend an AI meeting at Google. I had been working on various aspects of AI since graduate school in the 1980s and had been tremendously impressed by what Google had accomplished. I also thought I had some good ideas to contribute. But I have to admit that I was there only as a tagalong. The meeting was happening so that a group of select Google AI researchers could hear from and converse with Douglas Hofstadter, a legend in AI and the author of a famous book cryptically titled *Gödel, Escher, Bach: an Eternal Golden Braid*, or more succinctly, *GEB* (pronounced "gee-ee-bee"). If you're a computer scientist, or a computer enthusiast, it's likely you've heard of it, or read it, or tried to read it.

Written in the 1970s, *GEB* was an outpouring of Hofstadter's many intellectual passions—mathematics, art, music, language, humor, and wordplay, all brought together to address the deep questions of how intelligence, consciousness, and the sense of self-awareness that each human experiences so fundamentally can emerge from the non-intelligent, nonconscious substrate of biological cells. It's also about how intelligence and self-awareness might eventually be attained by computers. It's a unique book; I don't know of any other book remotely like it. It's not an easy read, and yet it became a bestseller and won both the Pulitzer Prize and the National Book Award. Without a doubt, *GEB* inspired more young people to pursue AI than any other book. I was one of those young people.

In the early 1980s, after graduating from college with a math degree, I was living in New York City, teaching math in a prep school, unhappy, and casting about for what I really wanted to do in life. I discovered *GEB* after reading a rave review in *Scientific American*. I went out and bought the book immediately. Over the next several weeks, I devoured it, becoming increasingly convinced that not only did I want to become an AI researcher but I specifically wanted to work with Douglas Hofstadter. I had never before felt so strongly about a book, or a career choice.

At the time, Hofstadter was a professor in computer science at Indiana University, and my quixotic plan was to apply to the computer science PhD program there, arrive, and then persuade Hofstadter to accept me as a student. One minor problem was that I had never taken even one computer science course. I had grown up with computers; my father was a hardware engineer at a 1960s tech start-up company, and as a hobby he built a mainframe computer in our family's den. The refrigerator-sized Sigma 2 machine wore a magnetic button proclaiming "I pray in FORTRAN," and as a child I was half-convinced it did quietly at night, while the rest of the family was asleep. Growing up in the 1960s and '70s, I learned a bit of each of the popular languages of the day: FORTRAN, then BASIC, then Pascal, but I knew next to nothing about proper programming techniques, not to mention anything else an incoming computer science graduate student needs to know.

To speed along my plan, I quit my teaching job at the end of the school year, moved to Boston, and started taking introductory computer science courses to prepare for my new career. A few months into my new life, I was on the campus of the Massachusetts

Institute of Technology, waiting for a class to begin, and I caught sight of a poster advertising a lecture by Douglas Hofstadter, to take place in two days on that very campus. I did a double take; I couldn't believe my good fortune. I went to the lecture, and after a long wait for my turn in a crowd of admirers I managed to speak to Hofstadter. It turned out he was in the middle of a yearlong sabbatical at MIT, after which he was moving from Indiana to the University of Michigan in Ann Arbor.

To make a long story short, after some persistent pursuit on my part, I persuaded Hofstadter to take me on as a research assistant, first for a summer, and then for the next six years as a graduate student, after which I graduated with a doctorate in computer science from Michigan. Hofstadter and I have kept in close touch over the years and have had many discussions about AI. He knew of my interest in Google's AI research and was nice enough to invite me to accompany him to the Google meeting.

### Chess and the First Seed of Doubt

The group in the hard-to-locate conference room consisted of about twenty Google engineers (plus Douglas Hofstadter and myself), all of whom were members of various Google AI teams. The meeting started with the usual going around the room and having people introduce themselves. Several noted that their own careers in AI had been spurred by reading *GEB* at a young age. They were all excited and curious to hear what the legendary Hofstadter would say about AI. Then Hofstadter got up to speak. "I have some remarks about AI research in general, and here at Google in particular." His voice became passionate. "I am terrified. Terrified."

Hofstadter went on.<sup>3</sup> He described how, when he first started working on AI in the 1970s, it was an exciting prospect but seemed so far from being realized that there was no "danger on the horizon, no sense of it actually happening." Creating machines with humanlike intelligence was a profound intellectual adventure, a long-term research project whose fruition, it had been said, lay at least "one hundred Nobel prizes away."<sup>4</sup> Hofstadter believed AI was possible in principle: "The 'enemy' were people like John Searle, Hubert Dreyfus, and other skeptics, who were saying it was impossible. They did not understand that brain is a hunk of matter that obeys physical law and the computer can simulate anything ... the level of neurons, neurotransmitters, et cetera. In theory, it can be done." Indeed, Hofstadter's ideas about simulating intelligence at various levels—from neurons to consciousness—were discussed at length in *GEB* and had been the focus of his own research for decades. But in practice, until recently, it seemed to Hofstadter that general "human-level" AI had no chance of occurring in his (or even his children's) lifetime, so he didn't worry much about it.

Near the end of *GEB*, Hofstadter had listed "Ten Questions and Speculations" about artificial intelligence. Here's one of them: "Will there be chess programs that can beat anyone?" Hofstadter's speculation was "no." "There may be programs which can beat anyone at chess, but they will not be exclusively chess players. They will be programs of *general* intelligence."<sup>4</sup>

At the Google meeting in 2014, Hofstadter admitted that he had been "dead wrong." The rapid improvement in chess programs in the 1980s and '90s had sown the first seed of doubt in his appraisal of AI's short-term prospects. Although the AI pioneer Herbert Simon had predicted in 1957 that a chess program would be world champion "within 10 years," by the mid-1970s, when Hofstadter was writing *GEB*, the best computer chess programs played only at the level of a good (but not great) amateur. Hofstadter had befriended Eliot Hearst, a chess champion and psychology professor who had written extensively on how human chess experts differ from computer chess programs. Experiments showed that expert human players rely on quick recognition of patterns on the chessboard to decide on a move rather than the extensive brute-force look-ahead search that chess programs use. During a game, the best human players can perceive a configuration of pieces as a particular "kind of position" that requires a certain "kind of strategy." That is, these players can quickly recognize particular configurations and strategies as instances of higher-level concepts. Hearst argued that without such a general ability to perceive patterns and recognize abstract concepts, chess programs would never reach the level of the best humans. Hofstadter was persuaded by Hearst's arguments.

However, in the 1980s and '90s, computer chess saw a big jump in improvement, mostly due to the steep increase in computer speed. The best programs still played in a very unhuman way: performing extensive look-ahead to decide on the next move. By the mid-1990s, IBM's Deep Blue machine, with specialized hardware for playing chess, had reached the Grandmaster level, and in 1997 the program defeated the reigning world chess champion, Garry Kasparov, in a six-game match. Chess mastery, once seen as a pinnacle of human intelligence, had succumbed to a brute-force approach.

### Music: The Bastion of Humanity

Although Deep Blue's win generated a lot of hand-wringing in the press about the rise of intelligent machines, "true" AI still seemed quite distant. Deep Blue could play chess, but it couldn't do anything else. Hofstadter had been wrong about chess, but he still stood by the other speculations in *GEB*, especially the one he had listed first:

QUESTION: Will a computer ever write beautiful music?  
SPECULATION: Yes but not soon.

Hofstadter continued,

Music is a language of emotions, and until programs have emotions as complex as ours, there is no way a program will write anything beautiful. There can be "forgeries"—shallow imitations of the syntax of earlier music—but despite what one might think at first, there is much more to musical expression than can be captured in syntactic rules.... To think ... that we might soon be able to command a preprogrammed mass-produced mail-order twenty-dollar desk-model "music box" to bring forth from its sterile cir-

cuiry pieces which Chopin or Bach might have written had they lived longer is a grotesque and shameful misestimation of the depth of the human spirit.<sup>5</sup>

Hofstadter described this speculation as “one of the most important parts of *GEB*—I would have staked my life on it.”

In the mid-1990s, Hofstadter’s confidence in his assessment of AI was again shaken, this time quite profoundly, when he encountered a program written by a musician, David Cope. The program was called Experiments in Musical Intelligence, or EMI (pronounced “Emmy”). Cope, a composer and music professor, had originally developed EMI to aid him in his own composing process by automatically creating pieces in Cope’s specific style. However, EMI became famous for creating pieces in the style of classical composers such as Bach and Chopin. EMI composes by following a large set of rules, developed by Cope, that are meant to capture a general syntax of composition. These rules are applied to copious examples from a particular composer’s opus in order to produce a new piece “in the style” of that composer.

Back at our Google meeting, Hofstadter spoke with extraordinary emotion about his encounters with EMI:

I sat down at my piano and I played one of EMI’s mazurkas “in the style of Chopin.” It didn’t sound exactly like Chopin, but it sounded enough like Chopin, and like coherent music, that I just felt *deeply* troubled.

Ever since I was a child, music has thrilled me and moved me to the very core. And every piece that I love feels like it’s a direct message from the emotional heart of the human being who composed it. It feels like it is giving me access to their innermost soul. And it feels like there is *nothing* more human in the world than that expression of music. Nothing. The idea that pattern manipulation of the most superficial sort can yield things that sound as if they are coming from a human being’s heart is very, very troubling. I was just completely thrown by this.

Hofstadter then recounted a lecture he gave at the prestigious Eastman School of Music, in Rochester, New York. After describing EMI, Hofstadter had asked the Eastman audience—including several music theory and composition faculty—to guess which of two pieces a pianist played for them was a (little-known) mazurka by Chopin and which had been composed by EMI. As one audience member described later, “The first mazurka had grace and charm, but not ‘true-Chopin’ degrees of invention and large-scale fluidity... The second was clearly the genuine Chopin, with a lyrical melody; large-scale, graceful chromatic modulations; and a natural, balanced form.”<sup>6</sup> Many of the faculty agreed and, to Hofstadter’s shock, voted EMI for the first piece and “real-Chopin” for the second piece. The correct answers were the reverse.

In the Google conference room, Hofstadter paused, peering into our faces. No one said a word. At last he went on. “I was terrified by EMI. Terrified. I hated it, and was extremely threatened by it. It was threatening to destroy what I most cherished about humanity. I think EMI was the most quintessential example of the fears that I have about artificial intelligence.”

## Google and the Singularity

Hofstadter then spoke of his deep ambivalence about what Google itself was trying to accomplish in AI—self-driving cars, speech recognition, natural-language understanding, translation between languages, computer-generated art, music composition, and more. Hofstadter’s worries were underlined by Google’s embrace of Ray Kurzweil and his vision of the Singularity, in which AI, empowered by its ability to improve itself and learn on its own, will quickly reach, and then exceed, human-level intelligence. Google, it seemed, was doing everything it could to accelerate that vision. While Hofstadter strongly doubted the premise of the Singularity, he admitted that Kurzweil’s predictions still disturbed him. “I was terrified by the scenarios. Very skeptical, but at the same time, I thought, maybe their timescale is off, but maybe they’re right. We’ll be completely caught off guard. We’ll think nothing is happening and all of a sudden, before we know it, computers will be smarter than us.”

If this actually happens, “we will be superseded. We will be relics. We will be left in the dust.

“Maybe this is going to happen, but I don’t want it to happen *soon*. I don’t want my children to be left in the dust.”

Hofstadter ended his talk with a direct reference to the very Google engineers in that room, all listening intently: “I find it very scary, very troubling, very sad, and I find it terrible, horrifying, bizarre, baffling, bewildering, that people are rushing ahead blindly and deliriously in creating these things.”

## Why Is Hofstadter Terrified?

I looked around the room. The audience appeared mystified, embarrassed even. To these Google AI researchers, none of this was the least bit terrifying. In fact, it was old news. When Deep Blue beat Kasparov, when EMI started composing Chopin-like mazurkas, and when Kurzweil wrote his first book on the Singularity, many of these engineers had been in high school, probably reading *GEB* and loving it, even though its AI prognostications were a bit out of date. The reason they were working at Google was precisely to make AI happen—not in a hundred years, but now, as soon as possible. They didn’t understand what Hofstadter was so stressed out about.

People who work in AI are used to encountering the fears of people outside the field, who have presumably been influenced by the many science fiction movies depicting superintelligent machines that turn evil. AI researchers are also familiar with the worries that increasingly sophisticated AI will replace humans in some jobs, that AI applied to big data sets could subvert privacy and enable subtle discrimination, and that ill-understood AI systems allowed to make autonomous decisions have the potential to cause havoc.

Hofstadter’s terror was in response to something entirely different. It was not about AI becoming too smart, too invasive, too malicious, or even too useful. Instead, he was terrified that intelligence, creativity, emotions, and maybe even consciousness

itself would be too *easy* to produce—that what he valued most in humanity would end up being nothing more than a “bag of tricks,” that a superficial set of brute-force algorithms could explain the human spirit.

As *GEB* made abundantly clear, Hofstadter firmly believes that the mind and all its characteristics emerge wholly from the physical substrate of the brain and the rest of the body, along with the body’s interaction with the physical world. There is nothing immaterial or incorporeal lurking there. The issue that worries him is really one of complexity. He fears that AI might show us that the human qualities we most value are disappointingly simple to mechanize. As Hofstadter explained to me after the meeting, here referring to Chopin, Bach, and other paragons of humanity, “If such minds of infinite subtlety and complexity and emotional depth could be trivialized by a small chip, it would destroy my sense of what humanity is about.”

### I Am Confused

Following Hofstadter’s remarks, there was a short discussion, in which the nonplussed audience prodded Hofstadter to further explain his fears about AI and about Google in particular. But a communication barrier remained. The meeting continued, with project presentations, group discussion, coffee breaks, the usual—none of it really touching on Hofstadter’s comments. Close to the end of the meeting, Hofstadter asked the participants for their thoughts about the near-term future of AI. Several of the Google researchers predicted that general human-level AI would likely emerge within the next thirty years, in large part due to Google’s own advances on the brain-inspired method of “deep learning.”

I left the meeting scratching my head in confusion. I knew that Hofstadter had been troubled by some of Kurzweil’s Singularity writings, but I had never before appreciated the degree of his emotion and anxiety. I also had known that Google was pushing hard on AI research, but I was startled by the optimism several people there expressed about how soon AI would reach a general “human” level. My own view had been that AI had progressed a lot in some narrow areas but was still nowhere close to having the broad, general intelligence of humans, and it would not get there in a century, let alone thirty years. And I had thought that people who believed otherwise were vastly underestimating the complexity of human intelligence. I had read Kurzweil’s books and had found them largely ridiculous. However, listening to all the comments at the meeting, from people I respected and admired, forced me to critically examine my own views. While assuming that these AI researchers underestimated humans, had I in turn underestimated the power and promise of current-day AI?

Over the months that followed, I started paying more attention to the discussion surrounding these questions. I started to notice the slew of articles, blog posts, and entire books by prominent people suddenly telling us we should start worrying, right now, about the perils of “superhuman” AI. In 2014, the physicist Stephen Hawking proclaimed, “The development of full artificial intelligence could spell the end of the human race.”<sup>7</sup> In the same year, the entrepreneur Elon Musk, founder of the Tesla and

SpaceX companies, said that artificial intelligence is probably “our biggest existential threat” and that “with artificial intelligence we are summoning the demon.”<sup>8</sup> Microsoft’s cofounder Bill Gates concurred: “I agree with Elon Musk and some others on this and don’t understand why some people are not concerned.”<sup>9</sup> The philosopher Nick Bostrom’s book *Superintelligence*, on the potential dangers of machines becoming smarter than humans, became a surprise bestseller, despite its dry and ponderous style.

Other prominent thinkers were pushing back. Yes, they said, we should make sure that AI programs are safe and don’t risk harming humans, but any reports of near-term superhuman AI are greatly exaggerated. The entrepreneur and activist Mitchell Kapor advised, “Human intelligence is a marvelous, subtle, and poorly understood phenomenon. There is no danger of duplicating it anytime soon.”<sup>10</sup> The roboticist (and former director of MIT’s AI Lab) Rodney Brooks agreed, stating that we “grossly overestimate the capabilities of machines—those of today and of the next few decades.”<sup>11</sup> The psychologist and AI researcher Gary Marcus went so far as to assert that in the quest to create “strong AI”—that is, *general* human-level AI—“there has been almost no progress.”<sup>12</sup>

I could go on and on with dueling quotations. In short, what I found is that the field of AI is in turmoil. Either a huge amount of progress has been made, or almost none at all. Either we are within spitting distance of “true” AI, or it is centuries away. AI will solve all our problems, put us all out of a job, destroy the human race, or cheapen our humanity. It’s either a noble quest or “summoning the demon.”

### What This Book Is About

This book arose from my attempt to understand the true state of affairs in artificial intelligence—what computers can do now, and what we can expect from them over the next decades. Hofstadter’s provocative comments at the Google meeting were something of a wake-up call for me, as were the Google researchers’ confident responses about AI’s near-term future. In the chapters that follow, I try to sort out how far artificial intelligence has come, as well as elucidate its disparate—and sometimes conflicting—goals. In doing so, I consider how some of the most prominent AI systems actually work, and investigate how successful they are and where their limitations lie. I look at the extent to which computers can now do things that we believe to require high levels of intelligence—beating humans at the most intellectually demanding games, translating between languages, answering complex questions, navigating vehicles in challenging terrain. And I examine how they fare at the things we take for granted, the everyday tasks we humans perform without conscious thought: recognizing faces and objects in images, understanding spoken language and written text, and using the most basic common sense.

I also try to make sense of the broader questions that have fueled debates about AI since its inception: What do we actually mean by “general human” or even “superhuman” intelligence? Is current AI close to this level, or even on a trajectory to get there?

What are the dangers? What aspects of our intelligence do we most cherish, and to what extent would human-level AI challenge how we think about our own humanness? To use Hofstadter's terms, how terrified should we be?

This book is not a general survey or history of artificial intelligence. Rather, it is an in-depth exploration of some of the AI methods that probably affect your life, or will soon, as well as the AI efforts that perhaps go furthest in challenging our sense of human uniqueness. My aim is for you to share in my own exploration and, like me, to come away with a clearer sense of what the field has accomplished and how much further there is to go before our machines can argue for their own humanity.

## Part I

### Background

# 1

## The Roots of Artificial Intelligence

### Two Months and Ten Men at Dartmouth

The dream of creating an intelligent machine—one that is as smart as or smarter than humans—is centuries old but became part of modern science with the rise of digital computers. In fact, the ideas that led to the first programmable computers came out of mathematicians' attempts to understand human thought—particularly logic—as a mechanical process of "symbol manipulation." Digital computers are essentially symbol manipulators, pushing around combinations of the symbols 0 and 1. To pioneers of computing like Alan Turing and John von Neumann, there were strong analogies between computers and the human brain, and it seemed obvious to them that human intelligence could be replicated in computer programs.

Most people in artificial intelligence trace the field's official founding to a small workshop in 1956 at Dartmouth College organized by a young mathematician named John McCarthy.

In 1955, McCarthy, aged twenty-eight, joined the mathematics faculty at Dartmouth. As an undergraduate, he had learned a bit about both psychology and the nascent field of "automata theory" (later to become computer science) and had become intrigued with the idea of creating a thinking machine. In graduate school in the mathematics department at Princeton, McCarthy had met a fellow student, Marvin Minsky, who shared his fascination with the potential of intelligent computers. After graduating, McCarthy had short-lived stints at Bell Labs and IBM, where he collaborated, respectively, with Claude Shannon, the inventor of information theory, and Nathaniel Rochester, a pioneering electrical engineer. Once at Dartmouth, McCarthy persuaded Minsky, Shannon, and Rochester to help him organize "a 2 month, 10 man study of artificial intelligence to be carried out during the summer of 1956."<sup>1</sup> The term *artificial intelligence* was McCarthy's invention; he wanted to distinguish this field from a related effort called cybernetics.<sup>2</sup> McCarthy later admitted that no one really liked the name—after all, the goal was *genuine*, not "artificial," intelligence—but "I had to call it something, so I called it 'Artificial Intelligence.'"<sup>3</sup>

The four organizers submitted a proposal to the Rockefeller Foundation asking for funding for the summer workshop. The proposed study was, they wrote, based on "the

conjecture that every aspect of learning or any other feature of intelligence can be in principle so precisely described that a machine can be made to simulate it."<sup>4</sup> The proposal listed a set of topics to be discussed—natural-language processing, neural networks, machine learning, abstract concepts and reasoning, creativity—that have continued to define the field to the present day.

Even though the most advanced computers in 1956 were about a million times slower than today's smartphones, McCarthy and colleagues were optimistic that AI was in close reach: "We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer."<sup>5</sup>

Obstacles soon arose that would be familiar to anyone organizing a scientific workshop today. The Rockefeller Foundation came through with only half the requested amount of funding. And it turned out to be harder than McCarthy had thought to persuade the participants to actually come and then stay, not to mention agree on anything. There were lots of interesting discussions but not a lot of coherence. As usual in such meetings, "Everyone had a different idea, a hearty ego, and much enthusiasm for their own plan."<sup>6</sup> However, the Dartmouth summer of AI did produce a few very important outcomes. The field itself was named, and its general goals were outlined. The soon-to-be "big four" pioneers of the field—McCarthy, Minsky, Allen Newell, and Herbert Simon—met and did some planning for the future. And for whatever reason, these four came out of the meeting with tremendous optimism for the field. In the early 1960s, McCarthy founded the Stanford Artificial Intelligence Project, with the "goal of building a fully intelligent machine in a decade."<sup>7</sup> Around the same time, the future Nobel laureate Herbert Simon predicted, "Machines will be capable, within twenty years, of doing any work that a man can do."<sup>8</sup> Soon after, Marvin Minsky, founder of the MIT AI Lab, forecasted that "within a generation ... the problems of creating 'artificial intelligence' will be substantially solved."<sup>9</sup>

### Definitions, and Getting On with It

None of these predicted events have yet come to pass. So how far do we remain from the goal of building a "fully intelligent machine"? Would such a machine require us to reverse engineer the human brain in all its complexity, or is there a shortcut, a clever set of yet-unknown algorithms, that can produce what we recognize as full intelligence? What does "full intelligence" even mean?

"Define your terms ... or we shall never understand one another."<sup>10</sup> This admonition from the eighteenth-century philosopher Voltaire is a challenge for anyone talking about artificial intelligence, because its central notion—intelligence—remains so ill-defined. Marvin Minsky himself coined the phrase "suitcase word"<sup>11</sup> for terms like *intelligence* and its many cousins, such as *thinking*, *cognition*, *consciousness*, and *emotion*. Each is packed like a suitcase with a jumble of different meanings. *Artificial intelligence* inherits this packing problem, sporting different meanings in different contexts.

Most people would agree that humans are intelligent and specks of dust are not. Likewise, we generally believe that humans are more intelligent than worms. As for human intelligence, IQ is measured on a single scale, but we also talk about the different dimensions of intelligence: emotional, verbal, spatial, logical, artistic, social, and so forth. Thus, intelligence can be binary (something is or is not intelligent), on a continuum (one thing is more intelligent than another thing), or multidimensional (someone can have high verbal intelligence but low emotional intelligence). Indeed, the word *intelligence* is an over-packed suitcase, zipper on the verge of breaking.

For better or worse, the field of AI has largely ignored these various distinctions. Instead, it has focused on two efforts: one scientific and one practical. On the scientific side, AI researchers are investigating the mechanisms of “natural” (that is, biological) intelligence by trying to embed it in computers. On the practical side, AI proponents simply want to create computer programs that perform tasks as well as or better than humans, without worrying about whether these programs are actually *thinking* in the way humans think. When asked if their motivations are practical or scientific, many AI people joke that it depends on where their funding currently comes from.

In a recent report on the current state of AI, a committee of prominent researchers defined the field as “a branch of computer science that studies the properties of intelligence by synthesizing intelligence.”<sup>12</sup> A bit circular, yes. But the same committee also admitted that it’s hard to define the field, and that may be a good thing: “The lack of a precise, universally accepted definition of AI probably has helped the field to grow, blossom, and advance at an ever-accelerating pace.”<sup>13</sup> Furthermore, the committee notes, “Practitioners, researchers, and developers of AI are instead guided by a rough sense of direction and an imperative to ‘get on with it.’”

### An Anarchy of Methods

At the 1956 Dartmouth workshop, different participants espoused divergent opinions about the correct approach to take to develop AI. Some people—generally mathematicians—promoted mathematical logic and deductive reasoning as the language of rational thought. Others championed inductive methods in which programs extract statistics from data and use probabilities to deal with uncertainty. Still others believed firmly in taking inspiration from biology and psychology to create brain-like programs. What you may find surprising is that the arguments among proponents of these various approaches persist to this day. And each approach has generated its own panoply of principles and techniques, fortified by specialty conferences and journals, with little communication among the subspecialties. A recent AI survey paper summed it up: “Because we don’t deeply understand intelligence or know how to produce general AI, rather than cutting off any avenues of exploration, to truly make progress we should embrace AI’s ‘anarchy of methods.’”<sup>14</sup>

But since the 2010s, one family of AI methods—collectively called deep learning (or deep neural networks)—has risen above the anarchy to become the dominant AI paradigm. In fact, in much of the popular media, the term *artificial intelligence* itself has

come to mean “deep learning.” This is an unfortunate inaccuracy, and I need to clarify the distinction. AI is a field that includes a broad set of approaches, with the goal of creating machines with intelligence. Deep learning is only one such approach. Deep learning is itself one method among many in the field of *machine learning*, a subfield of AI in which machines “learn” from data or from their own “experiences.” To better understand these various distinctions, it’s important to understand a philosophical split that occurred early in the AI research community: the split between so-called symbolic and subsymbolic AI.

### Symbolic AI

First let’s look at *symbolic AI*. A symbolic AI program’s knowledge consists of words or phrases (the “symbols”), typically understandable to a human, along with rules by which the program can combine and process these symbols in order to perform its assigned task.

I’ll give you an example. One early AI program was confidently called the General Problem Solver<sup>15</sup> or GPS for short. (Sorry about the confusing acronym; the General Problem Solver predated the Global Positioning System.) GPS could solve problems such as the “Missionaries and Cannibals” puzzle, which you might have tackled yourself as a child. In this well-known conundrum, three missionaries and three cannibals all need to cross a river, but their boat holds only two people. If at any time the (hungry) cannibals outnumber the (tasty-looking) missionaries on one side of the river ... well, you probably know what happens. How do all six get across the river intact?

The creators of the General Problem Solver, the cognitive scientists Herbert Simon and Allen Newell, had recorded several students “thinking out loud” while solving this and other logic puzzles. Simon and Newell then designed their program to mimic what they believed were the students’ thought processes.

I won’t go into the details of how GPS worked, but its symbolic nature can be seen by the way the program’s instructions were encoded. To set up the problem, a human would write code for GPS that looked something like this:

```
CURRENT STATE:  
LEFT-BANK = [3 MISSIONARIES, 3 CANNIBALS, 1 BOAT]  
RIGHT-BANK = [EMPTY]
```

```
DESIRED STATE:  
LEFT-BANK = [EMPTY]  
RIGHT-BANK = [3 MISSIONARIES, 3 CANNIBALS, 1 BOAT]
```

In English, these lines represent the fact that initially the left bank of the river “contains” three missionaries, three cannibals, and one boat, whereas the right bank doesn’t contain any of these. The desired state represents the goal of the program—get everyone to the right bank of the river.

At each step in its procedure, GPS attempts to change its current state to make it more similar to the desired state. In its code, the program has “operators” (in the form of subprograms) that can transform the current state into a new state and “rules” that encode the constraints of the task. For example, there is an operator that moves some number of missionaries and cannibals from one side of the river to the other:

```
MOVE (#MISSIONARIES, #CANNIBALS, FROM-SIDE, TO-SIDE)
```

The words inside the parentheses are called arguments, and when the program runs, it replaces these words with numbers or other words. That is, #MISSIONARIES is replaced with the number of missionaries to move, #CANNIBALS with the number of cannibals to move, and FROM-SIDE and TO-SIDE are replaced with “LEFT-BANK” or “RIGHT-BANK,” depending on which riverbank the missionaries and cannibals are to be moved from. Encoded into the program is the knowledge that the boat is moved along with the missionaries and cannibals.

Before being able to apply this operator with specific values replacing the arguments, the program must check its encoded rules; for example, the maximum number of people that can move at a time is two, and the operator cannot be used if it will result in cannibals outnumbering missionaries on a riverbank.

While these symbols represent human-interpretable concepts such as *missionaries*, *cannibals*, *boat*, and *left bank*, the computer running this program of course has no knowledge of the meaning of these symbols. You could replace all occurrences of “MISSIONARIES” with “Z372B” or any other nonsense string, and the program would work in exactly the same way. This is part of what the term *General* refers to in General Problem Solver. To the computer, the “meaning” of the symbols derives from the ways in which they can be combined, related to one another, and operated on.

Advocates of the symbolic approach to AI argued that to attain intelligence in computers, it would not be necessary to build programs that mimic the brain. Instead, the argument goes, general intelligence can be captured entirely by the right kind of symbol-processing program. Agreed, the workings of such a program would be vastly more complex than the Missionaries and Cannibals example, but it would still consist of symbols, combinations of symbols, and rules and operations on symbols. Symbolic AI of the kind illustrated by GPS ended up dominating the field for its first three decades, most notably in the form of *expert systems*, in which human experts devised rules for computer programs to use in tasks such as medical diagnosis and legal decision-making. There are several active branches of AI that still employ symbolic AI; I’ll describe examples of it later, particularly in discussions of AI approaches to reasoning and common sense.

### Subsymbolic AI: Perceptrons

Symbolic AI was originally inspired by mathematical logic as well as by the way people described their conscious thought processes. In contrast, *subsymbolic* approaches to AI

took inspiration from neuroscience and sought to capture the sometimes-*unconscious* thought processes underlying what some have called fast perception, such as recognizing faces or identifying spoken words. Subsymbolic AI programs do not contain the kind of human-understandable language we saw in the Missionaries and Cannibals example above. Instead, a subsymbolic program is essentially a stack of equations—a thicket of often hard-to-interpret operations on numbers. As I’ll explain shortly, such systems are designed to learn from data how to perform a task.

An early example of a subsymbolic, brain-inspired AI program was the perceptron, invented in the late 1950s by the psychologist Frank Rosenblatt.<sup>16</sup> The term *perceptron* may sound a bit 1950s science-fiction-y to our modern ears (as we’ll see, it was soon followed by the “cognitron” and the “neocognitron”), but the perceptron was an important milestone in AI and was the influential great-grandparent of modern AI’s most successful tool, deep neural networks.

Rosenblatt’s invention of perceptrons was inspired by the way in which neurons process information. A neuron is a cell in the brain that receives electrical or chemical input from other neurons that connect to it. Roughly speaking, a neuron sums up all the inputs it receives from other neurons, and if the total sum reaches a certain threshold level, the neuron fires. Importantly, different connections (*synapses*) from other neurons to a given neuron have different strengths; in calculating the sum of its inputs, the given neuron gives more weight to inputs from stronger connections than inputs from weaker connections. Neuroscientists believe that adjustments to the strength of connections between neurons is a key part of how learning takes place in the brain.

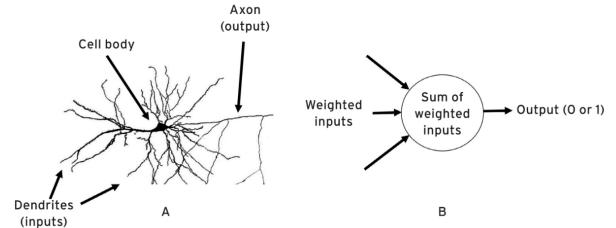


FIGURE 1: A, a neuron in the brain; B, a simple perceptron

To a computer scientist (or, in Rosenblatt’s case, a psychologist), information processing in neurons can be simulated by a computer program—a perceptron—that has multiple numerical inputs and one output. The analogy between a neuron and a perceptron is illustrated in figure 1. Figure 1A shows a neuron, with its branching dendrites (fibers that carry inputs to the cell), cell body, and axon (that is, output channel) labeled. Figure 1B shows a simple perceptron. Analogous to the neuron, the perceptron adds up its inputs, and if the resulting sum is equal to or greater than the perceptron’s

*threshold*, the perceptron outputs the value 1 (it “fires”); otherwise it outputs the value 0 (it “does not fire”). To simulate the different strengths of connections to a neuron, Rosenblatt proposed that a numerical *weight* be assigned to each of a perceptron’s inputs; each input is multiplied by its weight before being added to the sum. A perceptron’s *threshold* is simply a number set by the programmer (or, as we’ll see, learned by the perceptron itself).

In short, a perceptron is a simple program that makes a yes-or-no (1 or 0) decision based on whether the sum of its weighted inputs meets a threshold value. You probably make some decisions like this in your life. For example, you might get input from several friends on how much they liked a particular movie, but you trust some of those friends’ taste in movies more than others. If the total amount of “friend enthusiasm”—giving more weight to your more trusted friends—is high enough (that is, greater than some unconscious threshold), you decide to go to the movie. This is how a perceptron would decide about movies, if only it had friends.

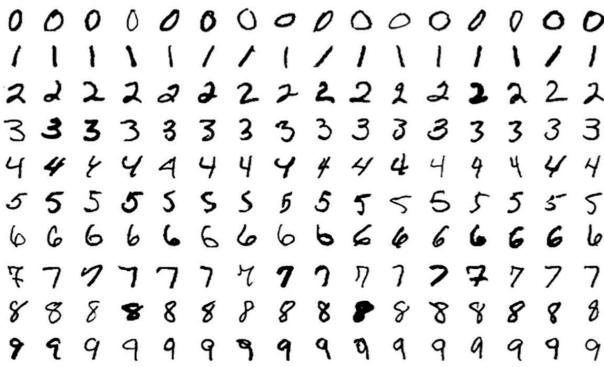


FIGURE 2: Examples of handwritten digits

Inspired by networks of neurons in the brain, Rosenblatt proposed that networks of perceptrons could perform visual tasks such as recognizing faces and objects. To get a flavor of how that might work, let’s explore how a perceptron might be used for a particular visual task: recognizing handwritten digits like those in figure 2.

In particular, let’s design a perceptron to be an 8 detector—that is, to output a 1 if its inputs are from an image depicting an 8, and to output a 0 if the image depicts some other digit. Designing such a detector requires us to (1) figure out how to turn an image into a set of numerical inputs, and (2) determine numbers to use for the perceptron’s

weights and threshold, so that it will give the correct output (1 for 8s, 0 for other digits). I’ll go into some detail here because many of the same ideas will arise later in my discussions of neural networks and their applications in computer vision.

### Our Perceptron’s Inputs

Figure 3A shows an enlarged handwritten 8. Each grid square is a pixel with a numerical “intensity” value: white squares have an intensity of 0, black squares have an intensity of 1, and gray squares are in between. Let’s assume that the images we give to our perceptron have been adjusted to be the same size as this one:  $18 \times 18$  pixels. Figure 3B illustrates a perceptron for recognizing 8s. This perceptron has 324 (that is,  $18 \times 18$ ) inputs, each of which corresponds to one of the pixels in the  $18 \times 18$  grid. Given an image like the one in figure 3A, each of the perceptron’s inputs is set to the corresponding pixel’s intensity. Each of the inputs would have its own weight value (not shown in the figure).

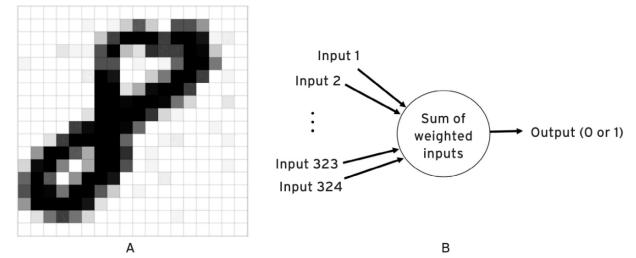


FIGURE 3: An illustration of a perceptron that recognizes handwritten 8s. Each pixel in the  $18 \times 18$ -pixel image corresponds to an input for the perceptron, yielding 324 ( $= 18 \times 18$ ) inputs.

### Learning the Perceptron’s Weights and Threshold

Unlike the symbolic General Problem Solver system that I described earlier, a perceptron doesn’t have any explicit rules for performing its task; all of its “knowledge” is encoded in the numbers making up its weights and threshold. In his various papers, Rosenblatt showed that given the correct weight and threshold values, a perceptron like the one in figure 3B can perform fairly well on perceptual tasks such as recognizing simple handwritten digits. But how, exactly, can we determine the correct weights and threshold for a given task? Again, Rosenblatt proposed a brain-inspired answer: the perceptron should *learn* these values on its own. And how is it supposed to learn the correct values? Like the behavioral psychology theories popular at the time, Rosen-

blatt's idea was that perceptrons should learn via *conditioning*. Inspired in part by the behaviorist psychologist B. F. Skinner, who trained rats and pigeons to perform tasks by giving them positive and negative reinforcement, Rosenblatt's idea was that the perceptron should similarly be *trained* on examples: it should be rewarded when it fires correctly and punished when it errs. This form of conditioning is now known in AI as supervised learning. During training, the learning system is given an example, it produces an output, and it is then given a "supervision signal," which tells how much the system's output differs from the correct output. The system then uses this signal to adjust its weights and threshold.

The concept of supervised learning is a key part of modern AI, so it's worth discussing in more detail. Supervised learning typically requires a large set of positive examples (for instance, a collection of 8s written by different people) and negative examples (for instance, a collection of other handwritten digits, not including 8s). Each example is *labeled* by a human with its category—here, 8 or not-8. This label will be used as the supervision signal. Some of the positive and negative examples are used to *train* the system; these are called the *training set*. The remainder—the *test set*—is used to evaluate the system's performance after it has been trained, to see how well it has learned to answer correctly in general, not just on the training examples.

Perhaps the most important term in computer science is *algorithm*, which refers to a "recipe" of steps a computer can take in order to solve a particular problem. Frank Rosenblatt's primary contribution to AI was his design of a specific algorithm, called the perceptron-learning algorithm, by which a perceptron could be trained from examples to determine the weights and threshold that would produce correct answers. Here's how it works: Initially, the weights and threshold are set to random values between -1 and 1. In our example, the weight on the first input might be set to 0.2, the weight on the second input set to -0.6, and so on, and the threshold set to 0.7. A computer program called a random-number generator can easily generate these initial values.

Now we can start the training process. The first training example is given to the perceptron; at this point, the perceptron doesn't see the correct category label. The perceptron multiplies each input by its weight, sums up all the results, compares the sum with the threshold, and outputs either 1 or 0. Here, the output 1 means a guess of 8, and the output 0 means a guess of not-8. Now, the training process compares the perceptron's output with the correct answer given by the human-provided label (that is, 8 or not-8). If the perceptron is correct, the weights and threshold don't change. But if the perceptron is wrong, the weights and threshold are changed a little bit, making the perceptron's sum on this training example closer to producing the right answer. Moreover, the amount each weight is changed depends on its associated input value; that is, the blame for the error is meted out depending on which inputs had the most impact. For example, in the 8 of figure 3A, the higher-intensity (here, black) pixels would have the most impact, and the pixels with 0 intensity (here, white) would have no impact. (For interested readers, I have included some mathematical details in the notes.<sup>17</sup>)

The whole process is repeated for the next training example. The training process goes through all the training examples multiple times, modifying the weights and threshold a little bit each time the perceptron makes an error. Just as the psychologist B. F. Skinner found when training pigeons, it's better to learn gradually over many trials; if the weights and threshold are changed too much on any one trial, then the system might end up learning the wrong thing (such as an overgeneralization that "the bottom and top halves of an 8 are always equal in size"). After many repetitions on each training example, the system eventually (we hope) settles on a set of weights and a threshold that result in correct answers for all the training examples. At that point, we can evaluate the perceptron on the test examples to see how it performs on images it hasn't been trained on.

An 8 detector is useful if you care only about 8s. But what about recognizing other digits? It's fairly straightforward to extend our perceptron to have ten outputs, one for each digit. Given an example handwritten digit, the output corresponding to that digit should be 1, and all the other outputs should be 0. This extended perceptron can learn all of its weights and thresholds using the perceptron-learning algorithm; the system just needs enough examples.

Rosenblatt and others showed that networks of perceptrons could learn to perform relatively simple perceptual tasks; moreover, Rosenblatt proved mathematically that for a certain, albeit very limited, class of tasks, perceptrons with sufficient training could, in principle, learn to perform these tasks without error. What wasn't clear was how well perceptrons could perform on more general AI tasks. This uncertainty didn't seem to stop Rosenblatt and his funders at the Office of Naval Research from making ridiculously optimistic predictions about their algorithm. Reporting on a press conference Rosenblatt held in July 1958, *The New York Times* featured this recap:

The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself, and be conscious of its existence. Later perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech and writing in another language, it was predicted.<sup>18</sup>

Yes, even at its beginning, AI suffered from a hype problem. I'll talk more about the unhappy results of such hype shortly. But for now, I want to use perceptrons to highlight a major difference between symbolic and subsymbolic approaches to AI.

The fact that a perceptron's "knowledge" consists of a set of numbers—namely, the weights and threshold it has learned—means that it is hard to uncover the rules the perceptron is using in performing its recognition task. The perceptron's rules are not symbolic; unlike the General Problem Solver's symbols, such as LEFT-BANK, #MISSIONARIES, and MOVE, a perceptron's weights and threshold don't stand for particular concepts. It's not easy to translate these numbers into rules that are understandable by humans. The situation gets much worse with modern neural networks that have millions of weights.

One might make a rough analogy between perceptrons and the human brain. If I could open up your head and watch some subset of your hundred billion neurons firing, I would likely not get any insight into what you were thinking or the “rules” you used to make a particular decision. However, the human brain has given rise to language, which allows you to use symbols (words and phrases) to tell me—often imperfectly—what your thoughts are about or why you did a certain thing. In this sense, our neural firings can be considered *subsymbolic*, in that they underlie the symbols our brains somehow create. Perceptrons, as well as more complicated networks of simulated neurons, have been dubbed “*subsymbolic*” in analogy to the brain. Their advocates believe that to achieve artificial intelligence, language-like symbols and the rules that govern symbol processing cannot be programmed directly, as was done in the General Problem Solver, but must emerge from neural-like architectures similar to the way that intelligent symbol processing emerges from the brain.

### The Limitations of Perceptrons

After the 1956 Dartmouth meeting, the symbolic camp dominated the AI landscape. In the early 1960s, while Rosenblatt was working avidly on the perceptron, the big four “founders” of AI, all strong devotees of the symbolic camp, had created influential—and well-funded—AI laboratories: Marvin Minsky at MIT, John McCarthy at Stanford, and Herbert Simon and Allen Newell at Carnegie Mellon. (Remarkably, these three universities remain to this day among the most prestigious places to study AI.) Minsky, in particular, felt that Rosenblatt’s brain-inspired approach to AI was a dead end, and moreover was stealing away research dollars from more worthy symbolic AI efforts.<sup>19</sup> In 1969, Minsky and his MIT colleague Seymour Papert published a book, *Perceptrons*,<sup>20</sup> in which they gave a mathematical proof showing that the types of problems a perceptron could solve *perfectly* were very limited and that the perceptron-learning algorithm would not do well in scaling up to tasks requiring a large number of weights and thresholds.

Minsky and Papert pointed out that if a perceptron is augmented by adding a “layer” of simulated neurons, the types of problems that the device can solve is, in principle, much broader.<sup>21</sup> A perceptron with such an added layer is called a multilayer neural network. Such networks form the foundations of much of modern AI; I’ll describe them in detail in the next chapter. But for now, I’ll note that at the time of Minsky and Papert’s book, multilayer neural networks were not broadly studied, largely because there was no general algorithm, analogous to the perceptron-learning algorithm, for learning weights and thresholds.

The limitations Minsky and Papert proved for simple perceptrons were already known to people working in this area.<sup>22</sup> Frank Rosenblatt himself had done extensive work on multilayer perceptrons and recognized the difficulty of training them.<sup>23</sup> It wasn’t Minsky and Papert’s mathematics that put the final nail in the perceptron’s coffin; rather, it was their speculation on multilayer neural networks:

[The perceptron] has many features to attract attention: its linearity; its intriguing learning theorem; its clear paradigmatic simplicity as a kind of parallel computation. There is no reason to suppose that any of these virtues carry over to the many-layered version. Nevertheless, we consider it to be an important research problem to elucidate (or reject) our intuitive judgment that the extension is sterile.<sup>24</sup>

Ouch. In today’s vernacular that final sentence might be termed “passive-aggressive.” Such negative speculations were at least part of the reason that funding for neural network research dried up in the late 1960s, at the same time that symbolic AI was flush with government dollars. In 1971, at the age of forty-three, Frank Rosenblatt died in a boating accident. Without its most prominent proponent, and without much government funding, research on perceptrons and other subsymbolic AI methods largely halted, except in a few isolated academic groups.

### AI Winter

In the meantime, proponents of symbolic AI were writing grant proposals promising impending breakthroughs in areas such as speech and language understanding, commonsense reasoning, robot navigation, and autonomous vehicles. By the mid-1970s, while some very narrowly focused expert systems were successfully deployed, the more general AI breakthroughs that had been promised had not materialized.

The funding agencies noticed. Two reports, solicited respectively by the Science Research Council in the U.K. and the Department of Defense in the United States, reported very negatively on the progress and prospects for AI research. The U.K. report in particular acknowledged that there was promise in the area of specialized expert systems—“programs written to perform in highly specialised problem domains, when the programming takes very full account of the results of human experience and human intelligence within the relevant domain”—but concluded that the results to date were “wholly discouraging about general-purpose programs seeking to mimic the problem-solving aspects of human [brain] activity over a rather wide field. Such a general-purpose program, the coveted long-term goal of AI activity, seems as remote as ever.”<sup>25</sup> This report led to a sharp decrease in government funding for AI research in the U.K.; similarly, the Department of Defense drastically cut funding for basic AI research in the United States.

This was an early example of a repeating cycle of bubbles and crashes in the field of AI. The two-part cycle goes like this. Phase 1: New ideas create a lot of optimism in the research community. Results of imminent AI breakthroughs are promised, and often hyped in the news media. Money pours in from government funders and venture capitalists for both academic research and commercial start-ups. Phase 2: The promised breakthroughs don’t occur, or are much less impressive than promised. Government funding and venture capital dry up. Start-up companies fold, and AI research slows. This pattern became familiar to the AI community: “AI spring,” followed by over-promising and media hype, followed by “AI winter.” This has happened, to various

degrees, in cycles of five to ten years. When I got out of graduate school in 1990, the field was in one of its winters and had garnered such a bad image that I was even advised to leave the term “artificial intelligence” off my job applications.

### Easy Things Are Hard

The cold AI winters taught practitioners some important lessons. The simplest lesson was noted by John McCarthy, fifty years after the Dartmouth conference: “AI was harder than we thought.”<sup>26</sup> Marvin Minsky pointed out that in fact AI research had uncovered a paradox: “Easy things are hard.” The original goals of AI—computers that could converse with us in natural language, describe what they saw through their camera eyes, learn new concepts after seeing only a few examples—are things that young children can easily do, but, surprisingly, these “easy things” have turned out to be harder for AI to achieve than diagnosing complex diseases, beating human champions at chess and Go, and solving complex algebraic problems. As Minsky went on, “In general, we’re least aware of what our minds do best.”<sup>27</sup> The attempt to create artificial intelligence has, at the very least, helped elucidate how complex and subtle are our own minds.