Artificial Intelligence: Historical Context and State of the Art

Arlindo L. Oliveira and Mário A. T. Figueiredo

Abstract The idea that intelligence is the result of a computational process and can, therefore, be automated, is centuries old. We review the historical origins of the idea that machines can be intelligent, and the most significant contributions made by Thomas Hobbes, Charles Babbage, Ada Lovelace, Alan Turing, Norbert Wiener, and others. Objections to the idea that machines can become intelligent have been raised and addressed many times, and we provide a brief survey of the arguments and counter-arguments presented over time. Intelligence was first viewed as symbol manipulation, leading to approaches that had some successes in specific problems, but did not generalize well to real-world problems. To address the difficulties faced by the early systems, which were brittle and unable to handle unforeseen complexities, machine learning techniques were increasingly adopted. Recently, a sub-field of machine learning known as deep learning has led to the design of systems that can successfully learn to address difficult problems in natural language processing, vision, and (yet to a lesser extent) interaction with the real world. These systems have found applications in countless domains and are one of the central technologies behind the fourth industrial revolution, also known as *Industry 4.0*. Applications in analytics enable artificial intelligence systems to exploit and extract economic value from data and are the main source of income for many of today's largest companies. Artificial intelligence can also be used in automation, enabling robots and computers to replace humans in many tasks. We conclude by providing some pointers to possible future developments, including the possibility of the development of artificial general intelligence, and provide leads to the potential implications of this technology in the future of humanity.

A. L. Oliveira (✉)
University of Lisbon, Instituto Superior Técnico, Lisbon, Portugal
e-mail: arlindo.oliveira@tecnico.ulisboa.pt

M. A. T. Figueiredo
Instituto de Telecomunicações/Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
e-mail: mario.figueiredo@tecnico.ulisboa.pt

H. Sousa Antunes et al. (eds.), *Multidisciplinary Perspectives on Artificial Intelligence and the Law*, Law, Governance and Technology Series 58,
https://doi.org/10.1007/978-3-031-41264-6_1

1 Historical Origins

The idea that intelligence can be automated has ancient roots. References to non-human thinking machines exist in Homer's Iliad and Thomas Hobbes clearly stated, in the Leviathan (Hobbes 1651), that human thought is no more than arithmetic computation. Both Pascal and Leibnitz, among many others, designed machines to automate arithmetic computations, which can be considered the precursors of modern calculators. But it was not until the mid-nineteenth century that the first proposal of a truly general computer appeared, created by Charles Babbage.

The original objectives of Babbage were to build an advanced tabulating device, which he called the *Difference Engine*. As envisaged by Babbage, this was a mechanical device that could be programmed to perform a series of computations specified in advance, by a complex arrangement of cogs, dented wheels, and levers. Although he managed to build only some parts of the Difference Engine, Babbage conceived of an even more powerful machine, the Analytical Engine. Had it been built, the engine would have had the ability to perform general computations in much the same way as a modern computer, although at a much slower speed imposed by its mechanical parts.

Although Babbage conceived the engine, it was Ada Lovelace, a friend mathematician, who wrote the most insightful analyses of the power of the engine (Menabrea and Lovelace 1843), arguing that it could do much more than just perform numeric computations. In particular, she observed that the machine might act upon things other than numbers if those things satisfied well-defined mathematical rules. She argued that the machine could write songs or perform abstract algebra, as long as those tasks could be expressed using symbolic languages. However, Lovelace also argued that the machine could not create anything new, but only perform exactly the tasks it was programmed for, ruling out the possibility that intelligent behavior could, somehow, be programmed into the machine. This argument was analyzed much later by an even more influential mathematician, Alan Turing.

2 Can Machines Think?

About a century later, Alan Turing, one of the most profound and creative mathematicians of all time, developed some of the central ideas of modern computing and came to different conclusions than those reached by Lovelace. Turing, who became known for having played an important role in the Allied World War II effort to decode the enemy messages encoded by the German Enigma cipher machines, achieved some of the most significant results in mathematics, namely in the mathematical foundations of computer science, results that are as important today as they were at the time they were obtained.

In a very important paper (Turing 1937), Turing showed that any digital computer with a large enough memory, which handles symbols and meets a few simple conditions, can perform the same calculations and compute the same set of functions as any other digital computer, a concept that became known as *Turing universality*. He described a particular type of computer, known today as a *Turing machine*, which uses a tape to write and read symbols as memory, and demonstrated that this type of computer can (in principle, assuming an unbounded tape) perform the same operations, do the same calculations, as any other computer that manipulates symbols. In the same year, Alonzo Church published a description of the so-called *lambda calculus* (Church 1936), a formal system for expressing computation based on function abstraction and application, which is also a universal model of computation with the same expressive power as Turing machines.

The combination of these two results lead to what became known as the *Church-Turing thesis*, which can be stated, informally, as follows: any result that can be actually calculated can be computed by a Turing machine, or by any another computer that manipulates symbols and has enough memory. This theoretical and purely mathematical result has important philosophical consequences. Note that there is a somewhat circular definition in this formulation: what exactly does the sentence "a result that can be actually calculated" mean? Are there any numerical results that are not in this category? The work of Alonzo Church, Alan Turing, and Kurt Gödel demonstrated that there are results that, although perfectly well defined, cannot be calculated. In 1931, Gödel proved that no consistent system of axioms is sufficient to prove all truths about the arithmetic of natural numbers and that, for any such consistent formal system, there will always be statements about natural numbers that are true, but that are unprovable within the system (Gödel 1931).

There is a close connection between Gödel's result and the problem that Turing addressed in his 1937 paper, and which can be stated in a simple way (that became known as the *halting problem*): is it possible to determine whether the execution of a computer program with a given input will terminate? Turing demonstrated that it is not possible to answer this question in the general case. It may be possible to answer the question in particular cases, but there are programs for which it is not possible to determine whether or not their execution will terminate.

Armed with these important insights on the nature and power of digital computers, Turing moved forward to analyze another important question, which is at the philosophical core of the field of artificial intelligence: can a computer behave intelligently?

Before describing Turings's work of 1950, in which he proposes an answer to this question, it is important to understand the consequences of considering the mechanistic ideas of Thomas Hobbes and the Church-Turing thesis together. Hobbes argued that the reasoning carried out by the human brain is nothing more than mathematical symbol manipulation. Church and Turing demonstrated that all machines that manipulate symbols are equivalent to each other, as long as they satisfy certain minimum requirements and are not limited in the time they are allowed to take to perform a given task, neither in the available memory. The result of these two ideas may arguably lead to the conclusion that a computer, in the

broadest sense of the term, should be able to carry out the same manipulation of symbols as a human brain and, therefore, be as intelligent as a human. There is, however, some disagreement on the scientific community about this conclusion, as stated. Some people believe that the type of substrate where the computations are carried out (biological or digital) may be important, while others argue that the conclusion may be true in principle but irrelevant in practice due to several types of difficulties.

In his famous paper (Turing 1950), Turing asked exactly this question: can machines think? To avoid the difficulties inherent in defining what "thinking" means, Turing proposed to reformulate the question into a different and better-defined problem. In particular, he proposed to analyze a hypothetical imitation game, a thought experiment that led to the now well-known *Turing test*. In the game proposed by Turing, an interrogator, in a separate room, communicates with a man and a woman, through typed text. The interrogator's objective is to distinguish the man from the woman by asking them questions.

Turing wondered if someday in the future a computer that is in the man's place can make the interrogator make a mistake as frequently as he would in the case where a man and a woman are present. Variations of this test were proposed by Turing himself in later texts, but the essence of the test remains the same: is it possible for an interrogator to distinguish between answers given by a computer and answers given by a human being? Turing argues that this question is, in essence, equivalent to the original question "Can machines think?" and it has the advantage of avoiding anthropomorphic prejudices that could condition the response. In fact, given our individual experience and human history, it is only natural to assume that only human beings can *think*.[1] This prejudice could stop us from obtaining an objective answer to the original question. It could happen that the interrogator would decide that the computer cannot think for the simple reason that the computer would not be like us since it doesn't have a head, arms, and legs, like humans. The use of the imitation game reduces the probability that prejudices rooted in our previous experience prevent us from recognizing a machine as a thinking being, even in cases where this could be true.

Turing not only proposes a positive answer to the question "can machines think?", but also indicates an approximate time in the future when this may happen. He argues that within half a century there would be machines with one Gigabyte of memory that would not be distinguishable from humans in a five-minute Turing test. We now know that Turing was somewhat optimistic. By the end of the twentieth century (50 years after Turing's paper), there were indeed machines with 1GB of memory but none of them were likely to pass a five-minute Turing test. Even today, more than 70 years after Turing's article, we still do not have machines like that,

[1] We are referring to thinking at a human level. Although many animals, namely higher vertebrates such as non-human primates, dolphins, and others, can engage in thought processes, such as those underlying action planning and complex social interactions, there is a qualitative difference between the complexity of the thought processes of those animals and that of humans.

although the latest *large language models* (which will be described later), such as ChatGPT, are arguably not far from passing a Turing test with interrogators that are not experts on their weaknesses.

3 Objections to Artificial Intelligence

An interesting part of Turing's 1950 article is where he classifies, analyzes, and responds to a set of objections to his proposal that, sometime in the future, computers might be able to think. The list of objections is instructive and as pertinent today as it was when the article was written.

The first objection is theological, arguing that human intelligence is the result of the immortal soul given by God to every human being, but not to animals or machines. Turing recognizes that he is unable to answer this objection scientifically, but nevertheless tries to provide some sort of answer, using what he considers to be theological reasoning. Turing argues that claiming that God cannot endow an animal or machine with a soul imposes an unacceptable restriction on the powers of the Almighty. Why can't God, Turing asks, endow an animal with a soul, if the animal is endowed with a capacity for thinking similar to that of a human being? A similar argument is valid for machines: won't God have the capacity to endow a machine with a soul, if he so desires and the machine can reason?

The second objection is based on the idea that the consequences of a machine being able to think would be so dire that it is better to hope that this will never happen. Turing feels that this argument is not strong enough to even merit an explicit rebuttal. However, seven decades after his article, there are proposals to stop the development of some artificial intelligence technologies, for fear of the possible negative consequences. Therefore, the objection that Turing analyzed is not entirely irrelevant and human-defined policies may become an obstacle to the development of intelligent machines.

The third objection is mathematical in nature and was later revisited by John Lucas and Roger Penrose. The objection is based on Gödel's theorem (mentioned above), according to which there are mathematical results that cannot be obtained by any machine or procedure. The objection is based on the argument that these limitations do not apply to the human brain. However, as Turing argues, no proof is given that the human brain is not subject to these limitations. Turing gives little credibility to this objection, despite the prestige of some of its advocates.

The fourth objection is based on the idea that only consciousness can lead to intelligence and that it will never be possible to demonstrate that a machine is conscious. Even if a machine can write a poem, only when the machine becomes aware of the meaning of the poem, can the machine be considered intelligent, the argument goes. Turing notes that, in the most radical version of this objection, only by becoming the machine could we be sure that the machine is conscious. But once we were the machine, it would be useless to describe the feelings or the sensation of consciousness, as we would be ignored by the rest of the world, which would not be

experiencing these sensations in person. Taken to an extreme, this objection reflects a solipsistic position, denying conscious behavior not only to machines but also to all other human beings since the existence of consciousness in other humans cannot be demonstrated beyond any doubt. While acknowledging that the phenomenon of consciousness remains unexplained (something true to this day), Turing does not regard this objection as decisive.

The fifth objection results from diverse and unsubstantiated arguments about behaviors that no machine can have. This category contains arguments like "no machine will have a sense of humor", "no machine will fall in love", "no machine will like strawberries and cream", and "no machine will be the object of its own thought". One curious (and perplexing) argument in this category is that "machines do not make errors", whereas humans do. As Turing points out, no justification is explicitly given for any of these limitations, which are supposed to be common to all machines. According to Turing, the objections arise, perhaps, from the wrong application of the principle of induction: so far, no machine has been in love, so no machine will ever be in love. A popular objection in this category is that no machine will ever have genuine feelings. Like the others, there is no scientific basis for this objection, it just reflects the limited view we have, based on our knowledge of the machines that currently exist.

The sixth objection is due to Ada Lovelace and was already referred to in the previous section. Although she realized that the analytic engine could process many other types of information besides numbers, Lovelace argued that the engine could never create anything new, as it only performed the operations for which it was programmed beforehand. Turing does not disagree with Lovelace's claim, but argues that it did not occur to Lovelace that the instructions could be so complex that they would lead the machine to actually create something new.

The seventh objection, perhaps even more popular today than it was in Turing's day, is based on the idea that the brain is not equivalent to a machine that manipulates symbols. We now know that the brain does not work in any way like a traditional computer, as the working principles of brains and digital computers are very different. Furthermore, the brain does not directly manipulate discrete symbols but physical variables with continuous values and theoretical results from mathematics tell us that a machine that manipulates continuous (real) values is necessarily more powerful than a machine that manipulates only discrete symbols. Turing's answer is that any machine that manipulates symbols, if properly programmed, will be able to give answers sufficiently close to the answers given by the machine that manipulates continuous values. Despite this response, this argument has held some weight over the decades. Many philosophers and scientists still believe that no machine that manipulates symbols can accurately emulate the behavior of the human brain and pass a Turing test.

The eighth objection is based on the argument that no set of rules is sufficient to describe the richness of human behavior. Since a machine always follows a set of rules, no machine can reproduce human behavior, which will always be unpredictable. Turing argues that it is not difficult for a machine to behave

unpredictably and it is not possible to demonstrate that, deep down, our brain does not function according to a set of rules.

The ninth, and final, objection, curiously the one to which Turing seems to give more weight, is based on the (supposed) existence of extrasensory perception and telepathic powers. Although ESP and telepathy progressively fell into disrepute in scientific circles, Turing apparently believed the evidence known at the time, which seemed to point to the existence of this phenomenon. As Turing very well argues, if there is extrasensory perception, the Turing test would have to be modified to avoid the possibility of telepathic communication. We now know, however, that there is no such thing as ESP, which makes the ninth objection irrelevant to the present discussion.

More than half a century after Turing's work, the objections to the possibility of thinking machines remain essentially the same and the answers given by Turing remain as valid now as they were then. None of the objections presented seem strong enough to convince us that machines cannot think, although of course this does not in any way prove that machines can think.

In the last decades, several other objections were raised against the Turing test as a mechanism for identifying intelligence. The first, and most important, of these objections, is that the test does not really assess intelligence, but whether the tested subject has an intelligence analogous to human intelligence. An intelligent computer (or even a hypothetical individual of a non-human intelligent species) would not pass the Turing test unless it could convince the examiner that it behaves as a human would. A system could even be much smarter than a human and still fail the test, for example, because it fails to disguise a superhuman ability for mathematics.

A second objection is that the test does not address all abilities by which human intelligence can express itself, but only those abilities that can be expressed through written language. Although the test can be generalized to include other forms of communication (e.g. questions could be asked using spoken language), there will still be difficulties in testing human skills that cannot be expressed through the interfaces that are chosen. On the other hand, Turing explicitly proposed a test of limited duration, of a few minutes, which is profoundly different from a test where the interaction is prolonged over hours, days, or even years.

A third objection has to do with the relationship between intelligence and consciousness. Although Turing addressed the question of consciousness when he analyzed the fourth objection on his list, he does not explicitly maintain that a machine that passes the test will necessarily be conscious. Turing avoids explicitly discussing this issue, an attitude that can be considered wise, given that the relationship between intelligence and consciousness remains almost as mysterious nowadays as it was in 1950. Still, very recent proposals address the relationship between computational models and consciousness and this is a field that is being actively studied today.

Despite these objections, and others that we have not analyzed, the Turing test remains important, more as a philosophical instrument that allows us to scrutinize the arguments related to the possibility of the existence of artificial intelligence than as a real mechanism for analyzing the capabilities thereof.

In addition to proposing the Turing test, the article written by Turing in 1950 makes one last suggestion that, prophetically, points in the direction that artificial intelligence would finally take, a couple of decades later. Turing proposed that instead of trying to write a program that would allow a machine to pass the imitation game, it would be simpler to write a program that would enable a machine to learn from experience, just as a baby does. Such a program, Turing argued, would be much easier to write and would allow a machine to learn what it needed to finally pass the Turing test.

This suggestion, so important and prescient, predated in a couple of decades the most successful approach to the problem of creating intelligent systems: machine learning. Instead, the first approaches adopted to try to create intelligent systems were based on the idea that human intelligence, in its most elaborate and evolved forms, consists in the manipulation of symbolic representations of knowledge about the world and the deduction of new knowledge through the manipulation of these symbols.

4 Intelligence as Symbol Manipulation

The idea that intelligent machines could exist, championed by Turing and many others, quickly led to the project of building them. Starting in the 1950s, digital computers became more powerful and increasingly accessible. The first computers were dedicated to scientific and military calculations, but progressively their application spread to other areas of human activity. With the end of the Second World War, the possibility of using computers in activities not related to military applications became a reality. One of the areas that deserved significant attention was the nascent domain of artificial intelligence.

In 1956, a scientific workshop took place in Dartmouth, New Hampshire, bringing together several of the pioneers in the field of artificial intelligence. In fact, it was in the proposal to organize this conference, written by John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon (the famous father of information and communication theory), that the term *artificial intelligence* was coined. Many of those who were present at this meeting went on to create research groups in artificial intelligence at the most important universities in the United States. Those early approaches tried to reproduce parts of human reasoning that at the time seemed the most advanced, such as proving theorems, planning sequences of actions, and playing board games, such as checkers and chess.

Not surprisingly, the first efforts to reproduce human intelligence thus focused precisely on problems requiring the manipulation of symbols and the search for solutions. In that same year, a program written by Allen Newell and Herbert Simon (who also attended the Dartmouth workshop), called the *Logic Theorist*, was able to demonstrate mathematical theorems (Newell and Simon 1956), including some of those in Whitehead and Russell's influential *Principia Mathematica*.

In 1959, Arthur Samuel (who had also attended the Dartmouth workshop) wrote a program that could play checkers well enough to defeat its creator (Samuel 1959). The program incorporated several concepts developed in the field of artificial intelligence, including the ability to look for solutions in very large and complex search spaces. To play checkers well, it is necessary to select, among all the possible moves, those leading to the best results. Since, for each move, the opponent can respond with one of several moves, which must also be answered by the program, this process leads to a very rapid growth in the number of positions that need to be analyzed. This branching of the search process takes the form of a tree, which is thus called a search tree. Developing methods to efficiently explore these search trees became one of the most important instrumental objectives in the field of artificial intelligence.

Efficient search methods are as important today as they were when they were first studied and developed. These methods are also applied in many other areas, namely for planning problems. A robot needs to perform a search to find out how to stack blocks, in a simplified block world, or to find its way from one room to another. Many results of artificial intelligence resulted from studies carried out with simplified environments, where robots were taught to manipulate blocks to achieve certain goals or to move around in controlled environments. One of the first projects to make a robot perform certain tasks, in a simplified block world, led to the development of a system that could manipulate and arrange blocks in specific configurations using vision and natural language processing.

Natural language processing, which aims at making computers process (for example, translate) and even understand written sentences, was another of the problems studied in this first phase of artificial intelligence. Despite the difficulties inherent to this processing, mainly caused by the existence of great ambiguities in the way humans use language, systems that conducted simple conversations, in ordinary English, were designed. The most famous of these early systems, ELIZA (Weizenbaum 1966), was designed by Joseph Weizenbaum and was able to converse with a user, in plain written English. ELIZA used a very simple set of mechanisms to answer questions, using pre-written sentences or simply rephrasing the question in slightly different terms. Although the system had no real understanding of the conversation, many users were tricked into thinking they were talking to a human being. In a way, ELIZA was one of the first systems to pass a Turing test, albeit a test administered under very specific and rather undemanding conditions.

Other projects aimed to create ways to represent human knowledge, so that it could be manipulated and used to generate new knowledge. Through the application of rules of deductive reasoning to knowledge bases, it was possible, for example, to build systems that were able to make medical diagnoses in certain particularly controlled conditions, where knowledge could be expressed symbolically, and combined using rules for the manipulation of symbols. Some so-called *expert systems* were developed based on these techniques and played relevant roles in different areas, mainly in the 1970s and 1980s.

These and other projects demonstrated that some of the capabilities of the human brain that seemed more complex and sophisticated, such as demonstrating

mathematical theorems or playing board games, could be programmed into a computer. These results led to several excessively optimistic predictions about the future evolution of artificial intelligence. In the 1960s, several renowned researchers, including Marvin Minsky and Herbert Simon (who had also attended the Dartmouth workshop), predicted that it would be possible, within three decades, to develop human-like intelligence in computers and to create systems that could perform any function performed by human beings. Those predictions, however, turned out to be unduly optimistic. The research carried out in those decades ended up showing that many tasks easily performed by humans are very difficult to replicate in computers. In particular, it proved exceptionally difficult to translate the results obtained in simplified environments, like the blocks world, to more complex and uncertain environments, such as a bedroom, a kitchen, or a factory. Tasks as simple as recognizing faces or perceiving spoken language proved to be of insurmountable complexity and were never solved by approaches based solely on symbol manipulation.

In fact, almost all of the capabilities of the human brain that have to do with perception and real-world interaction have proved especially difficult to replicate. For example, analyzing a scene captured by a camera and identifying the relevant objects therein is a very difficult task for a computer program, and only now it is finally beginning to be achievable by the most modern artificial intelligence systems. Despite this, we perform it without apparent effort or specific training. Other tasks that we perform easily, such as recognizing a familiar face or understanding a sentence in a noisy environment, are equally difficult to reproduce.

This difficulty contrasts with the relative ease with which it was possible to write computer programs that reproduce the intelligent manipulation of symbols, described in some of the approaches mentioned. This somewhat unexpected difficulty in reproducing behaviors that are trivial for humans and many animals on a computer is called Moravec's paradox: it is easier to reproduce on a computer behaviors that, for humans, require explicit complex mathematical reasoning than it is to recognize a face or perceive natural language, something a child does with great ease and with no specific instructions.

The difficulty in solving most problems involving perception and other characteristics of human intelligence led to several disillusionments with the field of artificial intelligence, the so-called *AI winters*. Despite these negative phases, marked by discouragement and lack of funding for projects in the area, the development of artificial intelligence systems based on symbol manipulation contributed, in different ways, to the creation of many algorithms that are executed by today's computers, in the most varied applications. This area has developed numerous methods of searching and representing knowledge that made it possible to create many programs that perform tasks that we often do not associate with intelligent systems. For example, the optimization of timetables for trains, airplanes, and other transportation systems is often performed by systems based on search and planning algorithms developed by the artificial intelligence community. Similarly, the systems created in the last decade of the twentieth century to play chess use search techniques that are essentially those proposed by this same community.

The methods and algorithms that made it possible to build the search engines that are now one of the central pillars of the Internet, and which allow us to locate, in fractions of a second, the relevant documents on a given topic, are also due to a specific sub-area of artificial intelligence: information retrieval. These systems identify, from the terms that are used in the search, the relevant documents and use different methods to determine which are the most important. The latest versions of these search engines use large language models (which we will describe later) to better understand the users' intentions and provide them with the most meaningful answers possible.

Nevertheless, with few exceptions, the systems based on symbol manipulation are rigid, unadaptable, and brittle. Few, if any, are capable of communicating in natural language, spoken or written, and of understanding the essence of complex questions. They are also uncapable of performing tasks requiring complex, unstructured image processing, or solving challenges that require adaptability to real-world, uncontrolled environments. Although artificial intelligence researchers have developed numerous robotic systems, few of these interact with humans in uncontrolled environments. Robotic systems are used extensively in factories and other industrial environments but, in general, they do so in controlled environments, subject to strict and inflexible rules, which allow them to perform repetitive tasks, based on the manipulation of parts and instruments that always appear in the same positions and the same settings.

Only very recently, after decades of research, have we started to have systems and robots that interact with the real world, with all its complexity and unpredictability. Although they also manipulate symbols, they are based on another idea, the idea that computers could learn from experience, and adapt their behavior intelligently, like children do.

5 Machine Learning

5.1 Basic Concepts

In the article Alan Turing wrote in 1950, he shows a clear awareness of the difficulty inherent in programming a system to behave intelligently. Turing proposed that, instead, it might be easier to build a program that simulates a child's brain. Duly submitted to an educational process, an adult brain would then be obtained, capable of reasoning and of higher intelligence. Turing compares a child's brain to a blank book, in which the experiences of a lifetime are recorded. Turing argued that it would probably be easier to build an adaptive system that uses machine learning to acquire the ability to reason and solve complex problems that we associate with human intelligence.

What exactly is this idea of machine learning, this idea that computers can learn from experience? At first glance, it goes against our intuition of computers, which

we see as machines that blindly obey a certain set of instructions. This was also the idea that Ada Lovelace had, much influenced by the mechanical computer to which she had access, which led her to the conclusion that computers are inflexible and cannot create anything new.

The key concept of machine learning is that it is possible for a system, when correctly configured, to adapt its behavior to approximate the intended results for a given set of inputs. At its core, the concept is easy to explain. Imagine a very simple system that receives as input a single number and generates on its output a single number, which depends on the first one. If this system is shown several examples of the intended correspondence between the input number and the output number, it can learn to guess this correspondence. Suppose the number on the input is the latitude of a city and the number on the output is the average temperature in that city during winter. If the system is given several examples of latitude/temperature pairs, the system can learn an approximate correspondence between the latitude and the temperature. Of course, the match will not, in general, be exact, because of local variations in the environment and location of the cities. But using mathematical techniques that are familiar to many readers, such as regression, it is possible to estimate the average winter temperature from the latitude of the city. Such a system represents perhaps the most basic machine learning system imaginable. This correspondence between the latitude and the temperature was not explicitly programmed by any programmer but inferred from data, using a mathematical formula or algorithm. The machine learning program, however, is very general. It can either infer this correspondence or the correspondence between the average income of a country and its energy usage per capita. Once written, this same program can learn to determine relationships of a certain type (for example, linear) between pairs of numbers, regardless of the concrete problem being analyzed.

In machine learning, the set of examples used to train the system is called the training set and the set of examples later presented to the system to test its performance is called the test set. Once trained, the system can be used many times to predict outputs from new input values, without the need for additional training. In many cases (namely if the training set if very large and the relationship being learned is very complex), the training process can be relatively slow, but using the system to determine the desired match for new examples is quick and efficient.

Returning to the example of latitude and average winter temperature, now imagine that the learning system is given, as input, not only latitude, but also the average winter energy consumption by household, the distance from the sea, and other relevant variables. It is easy to see that the program can now learn, with much more precision, to calculate the relationship between this set of variables and the average winter temperature of a given city. Mathematically, the problem is more complex, but the formulation is the same: given a set of input data, the objective is to obtain a program that generates an estimate of the output. Using the right algorithms, this problem is not much harder than the previous one.

If the output (i.e., the variable being predicted) is a quantitative variable (or collection of quantitative variables), the system obtained through the execution of the learning algorithm is called a regressor, and the problem is known as *regression*.

If the objective is to assign a given class to the object characterized by the input, the system is called a classifier. Let us now consider a much more difficult problem: imagine you are given images with, say, a million pixels. The goal is to learn to classify images in accordance with what is in them; for example, does each image contain, or not, a cat or a dog. Again, we have as input several variables, in this case, three million variables for a color one megapixel image, and the objective is to generate in the output a variable that indicates what is in the photo, a dog, a cat, or a car, for instance. There are, indeed, very large datasets, such as *ImageNet* (Deng et al. 2009) that has more than 14 million images in more than 20,000 categories, which are used to train machine learning systems.

The attentive reader will have noticed that this problem is in essence no different from the first one discussed above. In the present case, it is necessary to calculate the correspondence between three million numbers, the input, and the desired class, the output. Although it has the same formulation, this problem is dramatically more difficult. There is now no direct correspondence, through more or less simple mathematical formulas, between the intended input and output. To get the right class, we need to be able to identify diverse characteristics of the image, such as eyes, wheels, or whiskers, and how these features are spatially related to each other.

Here, too, Alan Turing's pioneering idea works. Although it is very difficult (arguably impossible for a human) to write a program that maps inputs to outputs, images to categories, it is possible to learn it from the labels and descriptions humans created for these images. This idea that a computer can learn from examples was developed from the 1960s onwards by numerous researchers and scientists. The first approaches, which used symbolic representations to learn the correspondences between inputs and outputs, ended up giving way to statistical and/or numerical methods. There are many ways to learn a correspondence between the values in the inputs and the intended outputs, and it is not possible here to describe in a minimally complete way even a small fraction of the algorithms used. However, it is possible to present, in a very brief way, the philosophy underlying most of these approaches.

5.2 Statistical Approaches

A class of approaches that originated in statistics is based on estimating, from the training data, a statistical relationship between the input and the output (Friedman et al. 2001). Some of these statistical approaches (in a subclass usually referred to as *generative*) are based on the famous Bayes law. This law, actually a theorem, computes the probability of occurrence of some event (for example, the class of a given image) from prior knowledge about that event (for example, the probability that any given random image contains a dog, a cat, a person, ...) and about how the input is related to the observations (for example, what are the statistics of images containing dogs). Another class of statistical approaches (usually called *discriminative*) bypasses the application of Bayes' law and, from the training data, estimates directly a model of how the probability of the output values depends on

the input (for example, the probability that a given image contains a dog, given all the pixel values of that image). In complex problems (such as image classification), the discriminative approach is by far more prevalent, because it has two important advantages: it makes a more efficient use of the available data and it is less reliant on assumptions about the form of the underlying statistical relationships being therefore more robust and general-purpose.

5.3 Similarity-Based Approaches

Other approaches focus on assessing the similarities between the different examples of the training set. Imagine we want to determine a person's weight based on their height, sex, age, and waist circumference. And suppose we intend to determine the weight of a new individual never seen before. A simple and pragmatic approach, if there are many examples in the training set, is to look in that set for a person (or set of persons) with characteristics very similar to the individual in question, and guess that the weight of the individual in question is the same as that of the most similar person in the training set. This approach, learning by analogy, is effective if there is a vast training set, and can be carried out in a variety of ways, with algorithms whose designations include the *nearest neighbor* (or k kearest neighbors) method (Fix and Hodges 1989). An extension of this class of method, based on assessing similarities between objects to be classified and those in the training set, led to a class of methods known as kernel machines (of which the most famous member is the *support vector machine*), which was very influential and had significant impact in the last decade of the twentieth century and beginning of this century (Schölkopf and Smola 2002).

5.4 Decision Trees

Yet another class of methods, known as decision trees (Quinlan 1986), work by splitting data into a series of binary decisions. Classifying a new object corresponds to traversing down the tree based on these decisions, moving through the decisions until a leaf node is reached, which will return the prediction (a class or a value). A decision tree is built by making use of a heuristic known as recursive partitioning, which exploits the rationale of divide and conquer. Decision trees have several important advantages, such as being seamlessly applicable to heterogeneous data (with quantitative and categorical variables, which is not true of most other methods), being somewhat analogous to the thought process of human decision making, and, very importantly, being transparent, since the chain of decisions that leads to the final prediction provides an explanation for that prediction. Decision trees can also be combined into so-called *random forests* (Ho 1995), a type of model that uses multiple decision trees, each learned from a different subset of

the data. The prediction of the forest is obtained by averaging those of the trees, which is known to improve its accuracy, at the cost of some loss of transparency and explainability. Random forests are, still today, one of the methods of choice in problems involving heterogenous data and for which the amount of training data available is not large.

5.5 Neural Networks

Neural networks are one of the most flexible and powerful approaches in use today. This approach was pioneered in the 1980s (McClelland et al. 1986) and is heavily inspired by much earlier work on mathematical models of biological neurons, namely by McCulloch and Pitts, in a famous 1943 paper with title "A Logical Calculus of Ideas Immanent in Nervous Activity", which proposed the first mathematical model of biological neurons. Also in the 1940s, Donald Hebb proposed the first biologically plausible learning process for neural networks (the Hebbian rule), which became famously summarized in the sentence "cells that fire together wire together" (Hebb 1949).

Although the functioning of a biological neuron is complex and hard to model, it is possible to build abstract mathematical models of the information processing performed by each of these cells, of which the one by McCulloch and Pitts was the first. In these simplified models, each cell accumulates the excitations received at its inputs and, when the value of this excitation exceeds a certain threshold, it fires, stimulating the neurons connected to its outputs. By implementing or simulating in a computer the interconnection of several of these units (artificial neurons), it is possible to reproduce the basic information processing mechanism used by biological brains. In a real brain, neurons are interconnected through synapses of varying strength. In the neural networks used in machine learning, the strength of the connection between neurons is defined by a number that controls the weight with which it influences the state of the neuron connected to it.

Mathematical methods, called training algorithms, are used to determine the values of these interconnection weights to maximize the accuracy of the correspondence between the computed output and the desired output, over a training set. In fact, these algorithms are essentially a form of feedback, where each error committed on a training sample is fed back to the network to adjust the weights in such a way that this error becomes less likely. The training algorithms that are prevalent in modern machine learning thus have deep roots in the work of Norbert Wiener, one of the greatest mathematicians and scientists of the twentieth century. Wiener's seminal work in cybernetics (a field that he created and baptized) includes the formalization of the notion of feedback, which is one of the cornerstones of much of modern technology. Wiener also influenced the early work on neural networks by bringing McCulloch and Pitts to MIT and creating the first research group where neuropsychologists, mathematicians, and biophysicists joined efforts to try to understand how biological brains work.

The first successful instance of a leaning algorithm for artificial neural networks, following the rationale of error feedback to update the network weights, was proposed by Frank Rosenblatt, in 1958, for a specific type of network, called *perceptron*. The perceptron together with Rosenblatt's algorithm 1958 are the precursors of the modern neural networks and learning algorithms. The early success of perceptrons spawned a large wave of enthusiasm and optimism, which turnec into disappointment when it became obvious that this optimism was very exaggerated. A symbolic moment in the crushing of expectations was the publication of the famous book "Perceptrons", by Minsky and Papert, in 1969. This book provided a mathematical proof of the limitations of perceptrons as well as unsupported statements regarding the challenges of training multi-layer perceptrons (which they recognized would solve those limitations). This disappointment was responsible for a dramatic decrease in the interest and funding for neural network research, which lasted for more than three decades.

Modern neural networks arose of the realization that it is indeed possible to learn/train networks with several layers (currently known as deep neural networks, to which the following section is devoted), which can be mathematically and experimentally shown to be highly flexible structures, capable of solving many prediction problems. At the heart of this possibility is a procedure, known as *backpropagation*, which allows implementing the above-mentioned feedback from prediction errors in training examples to adjustments in the network's weights, aiming at making these errors less likely. The term *backpropagation* and its application in neural networks is due to Rumelhart, Hinton, and Williams, in 1986, but the technique was independently rediscovered many times, and had many predecessors dating back to the 1960s, namely in feedback control theory.

Modern neural networks can have up to many millions of neurons and billions of weights. Sufficiently complex artificial neural networks can be used to process, images, sounds, text, and even videos. For example, when they are used to process images, each of the neurons in one of these networks ends up learning to recognize a certain characteristic of the image. One neuron might recognize a line at a given position, another neuron (deeper, that is, farther from the input) might recognize an outline of a nose, and a third, even deeper, might recognize a particular face. Again, some of the modern work on neural networks for analyzing images has old roots in work from the late 1950s and early 1960s, namely the neural models of the visual cortex of mammals described by Hubel and Wiesel, for which they received the Nobel Prize in 1981.

Although biological brains have inspired artificial neural networks and learning mechanisms, it is important to realize that, in the current state of technology, these networks do not work in the same way as biological brains do. Although networks of this type have been trained to drive vehicles, process texts, recognize faces in videos, and even play champion-level games like chess or Go (a very complex board game popular in Asia), it would be wrong to think that they use the same mechanisms the human brain uses to process information. In most cases, these networks are trained to solve a very specific problem and they are incapable of tackling other problems, let alone making decisions autonomously about which problems should be tackled.

How the human brain organizes itself, through the process of development and learning that takes place during childhood and adolescence, how new memories are kept throughout life, how different goals are pursued over time, by any of us, depend on essentially unknown mechanisms and are not present in artificial neural networks.

However, the last decade has seen the emergence of technologies that enable us to create very complex systems that, at least on the surface, exhibit somewhat more intelligent behavior.

6 The Deep Learning Revolution

In the last decade, machine learning led to remarkable developments in applications where symbolic methods did not perform well, such as computer vision, speech recognition, and natural language processing. These developments are collectively known as *deep learning*. The adjective "deep" refers to the use of multiple layers in neural networks, although other approaches that do not use neural networks also fall into the scope of deep learning (LeCun et al. 2015).

Deep learning commonly resorts to neural network architectures with many layers (essentially the concatenation of many perceptrons in a multilayer structure), leading to new applications and optimized implementations, mainly due to the availability of very large datasets for training these large structures with many paremeters (the weights referred above) and very efficient special-purpose computer processors, such as GPUs (graphics processing units). In deep learning, each neural network layer learns to transform its input into a somewhat more abstract and composite representation. For instance, in image recognition or classification, the raw input may be a matrix of pixels, the first layer may abstract the pixels and encode edges or corners, the second layer may compose and encode arrangements of edges and corners into lines and other shapes, and so on up to semantically meaningful concepts, such as faces or objects, or tumors in medical images. At the very core of the algorithms that learn these deep neural networks is the backpropagation algorithm that was mentioned in the previous section.

Deep learning has had many remarkable successes in recent years. Board games have been popular in artificial intelligence research ever since its inception in the fifties. Simpler games, like checkers and backgammon, have been mastered by machines decades ago, but other more complex games, like chess and Go took longer to solve. IBM's *Deep Blue* was the first chess program to beat a world champion (Campbell et al. 2002), when it defeated Garry Kasparov in 1997 in a rematch. However, *Deep Blue*, like many other chess programs, depended heavily on human-designed playing strategies and relied heavily on brute force search. Starting in 2016, a series of developments by *DeepMind*, a Google-owned company, led to the release of AlphaGo (Silver et al. 2016), a system that beat the best Go players in the world, after learning from expert games and self-play. Posterior developments led to AlphaGo Zero (Silver et al. 2017) and AlphaZero (Schrittwieser et al. 2020),

systems that excelled at Go, chess, and other games, but that did not need to learn from human experts and learned uniquely from self-playing, using deep learning and a technique for sequential decision making known as *reinforcement learning*. The outcomes achieved by these programs were remarkable, as they were able to learn techniques and strategies within a matter of days that had eluded humanity for millennia since the inception of these games. The games these machines play are currently being studied in order to understand the novel strategies and techniques they developed and that are alien to humans.

Another area where deep learning has led to significant advances is computer vision, where the goal is to enable computers to process, analyze, and even understand images and videos. One important feature of the neural network architectures developed for computer vision is that the first layers are *convolutional,* inspired by the architecture of the first layers of the neural visual system of mammals, as discovered by Hubel and Wiesel in the 1950–1960s. Convolutional layers enjoy a certain type of invariance (more precisely, equivariance, but that is beyond the scope of this introduction), which means, in simple terms, that the processing that is performed at each location of the image is the same across all the image. Convolutional layers are an embodiment of a so-called *inductive bias*: a property of the network that is designed rather than learned, based on knowledge of the data that will be processed and the purpose of the network. More specifically, the inductive bias in this case is that to recognize the presence of an object in an image, its location is irrelevant. Another crucial implication of the convolutional nature of these networks is that, due to this invariance, the number of parameters that needs to be learned is dramatically smaller than in an arbitrary network with the same size.

By combining new architectural features, such as convolutional layers (and many other tricks of the trade) with massive datasets and powerful processing engines based on GPUs and TPUs (tensor processing units), deep learning has led to an explosion in the range of applications of computer vision. These applications include face recognition (heavily used in modern smartphones and automated surveillance), image recognition and classification, surveillance, automated facility inspection, medical image reconstruction and analysis, and autonomous driving, among others.

Several modern deep learning architectures, such as transformers (Vaswani et al. 2017), have been developed for natural language processing and used to build large language models. These large language models, which are statistical in nature, such as GPT-3 (Brown et al. 2020), have been trained in corpora with trillions of words, and can accurately answer questions, complete sentences, and write articles and notes about many different topics. They are becoming increasingly useful in the development of customer interaction tools, as shown by the recent release of ChatGPT and GPT-4, mimicking in very impressive ways the behavior of human agents in analyzing and answering requests made in natural language. In some ways, these large language models are approaching Turing's vision of machines that interact in a way that, if based only on text, cannot be distinguished from interaction with humans. The enormous interest raised by the release of ChatGPT shows the potential of these approaches, although this system represents, really, just one more

step forward in the evolution of the technologies used in large language models, which are the result of the deep learning revolution.

7 Applications in Analytics and Automation

Modern artificial intelligence has many applications, in the most diverse fields, and cannot be easily classified using any simple taxonomy. However, they can be roughly clustered into two large, non-exclusive, areas: analytics and automation.

Analytics has a strong connection with other areas with designations such as data science, big data, data mining, or business intelligence. The fundamental goal of analytics is to organize existing data about people, organizations, businesses, or processes, in order to extract economic value from that data, by identifying regularities, propensities, or sensitivities that are susceptible to exploitation. Many of the world's largest companies, including those commonly known as GAFAM (Google, Apple, Facebook, Amazon, and Microsoft) owe much of their value to the ability to organize the data contributed by their users and resell it to support targeted advertising or other sales campaigns. However, the applications of analytics go far beyond advertising and marketing. Properly explored, the data obtained in the most diverse ways can also be used to discover new scientific knowledge and to optimize processes of design, manufacture, distribution, or sales, which represents an important component of another area that has become known as Industry 4.0. Any company that wants to be internationally competitive nowadays or any institution that wants to provide good services to its users must use analytics tools to explore and use the data they have.

Automation, on the other hand, has to do with the partial or total replacement of human beings in tasks that normally require intelligence. This area, whose economic impact is still probably smaller than that of analytics, will grow rapidly in the coming years, as companies and institutions continue to face pressure to become more efficient and reduce costs. Areas as diverse as customer support, legal services, human resources, logistics, distribution, banking, services, and transportation will progressively be transformed as functions previously performed by human employees are progressively automated by machines. This replacement process will be progressive and gradual, giving time for companies to adapt, but, inevitably, tasks such as customer service, facility surveillance, legal document analysis, medical diagnosis, vehicle driving, and many others will be progressively performed by automatic systems based on artificial intelligence. The current state of technology does not yet allow for the complete replacement of professionals in most of these tasks, but technological advance seems inevitable and their consequences, in the long run, indisputable.

Recently, the European Commission proposed two documents for possible adoption by the European Parliament that aim to regulate various aspects of the application of artificial intelligence technologies. These documents are somehow partially aligned with this taxonomy. The Digital Markets Act, proposed in

December 2020, focuses on the need to regulate access to data, for the purposes of analytics, and prevent excessive control of this data by large platforms (the document calls them Gatekeepers), which would lead to situations of overwhelming market dominance. The Artificial Intelligence Act, proposed in April 2021, focuses more on the problems caused by potential high-risk applications that are mainly in the field of automation. Systems considered high-risk by the document include, among many others, those that identify people, operate critical infrastructure, recruit or select candidates for positions or benefits, control access to facilities and countries, or play a role in education or administration of justice.

The European Union's ambition is to regulate artificial intelligence technologies, to maintain the security and privacy of citizens, guarantee competition and preserve the openness of markets while stimulating the development of new, secure, and non-invasive applications. However, the gap between essential and over-regulation is small, and compromises are often difficult. For example, the General Data Protection Regulation (GDPR) is certainly an important piece in the protection of citizens' rights and has placed Europe at the forefront in this field. But it is also a regulation whose compliance poses many challenges, demands, and difficulties for companies and institutions. Concerning the regulation of artificial intelligence, the hope is that Europe will manage to find an appropriate balance, preserving individual rights but also setting up the conditions for the creation of innovation, which will continue to be the engine of increased productivity and economic development.

8 Conclusions

Artificial intelligence, the field that has been developed to realize the idea that machines will one day also be able to *think*, is today an important technology that is behind profound changes that will affect society in the next few decades, globally known as the fourth industrial revolution.

Applications in analytics and automation will expand rapidly in the next decades, leading to changes in the way we live, work, and interact. Although the world seems profoundly changed by the technologies that are already in place, we must be prepared for even deeper changes in years to come, brought by the convergence of diverse technologies, of which artificial intelligence is the most central one.

Although we still do not know how to reproduce human-level intelligent behavior in machines, an objective known as artificial general intelligence, large-scale efforts to develop the technologies that could lead to such a result are being undertaken by all the major economic blocs, companies, research institutes, and universities. It is highly likely that the combined efforts of millions of researchers may eventually shed light on one of the most important questions that humanity has faced: what is intelligence and can it be reproduced in machines?

If artificial general intelligence is indeed possible and becomes a reality sometime in the future, it will raise significant practical, ethical, and social questions, which will have to be discussed and addressed, from a variety of standpoints.[2]

References

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. Adv Neural Inf Proces Syst 33:1877–1901

Campbell M, Hoane AJ, Hsu FH (2002) Deep Blue. Artif Intell 134:57–83

Church A (1936) An unsolvable problem of elementary number theory. Am J Math 58:345–363

Deng J, Dong W, Socher R, Li LJ, Kai L, Li FF (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 20–25

Fix E, Hodges JL (1989) Discriminatory analysis. Nonparametric discrimination: consistency properties. Int Stat Rev 57:238–247

Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning. Springer, New York

Gödel K (1931) Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatsh Math Phys 38-38:173–198

Hebb D (1949) The organization of behavior. Wiley & Sons, New York

Ho T (1995) Random decision forests, Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, pp 278–282

Hobbes T (1651) Leviathan: or, the matter, form, and power of a commonwealth ecclesiastical and civil. G. Routledge and Sons, Manchester and New York

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444

McClelland JL, Rumelhart DE, PDP Research Group (1986) Parallel distributed processing: explorations in the microstructure of cognition. MIT Press

Menabrea L, Lovelace A (1843) Sketch of the analytical engine invented by Charles Babbage. Sci Mem 3:666–731

Newell A, Simon H (1956) The logic theory machine–a complex information processing system. IEEE Trans Inf Theory 2:61–79

Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106

Rosenblatt F (1958) The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psy Rev 65(6):386–408

[2] *See* generally, on the different applications of Machine Learning and AI, in this book I Trancoso, N Mamede, B Martins, H S Pinto and R Ribeiro - The impact of language technologies in the legal domain; J Gonçalves-Sá and F L Pinheiro - Societal Implications of Recommendation Systems - A Technical Perspective; A T Freitas - Data-driven approaches in healthcare - challenges and emerging trends; M Correia and L Rodrigues - Security and Privacy; E Magrani and P G F Silva - The Ethical and Legal Challenges of Recommender Systems Driven by Artificial Intelligence; M Lanz and S Mijic - Risks associated with the use of natural language generation - Swiss civil liability law perspective; M S Fernandes and J R Goldim - Artificial Intelligence and Decision Making in Health - Risks and Opportunities; W Gravett - Judicial Decision-making in the Age of Artificial Intelligence; and D Durães, P M Freitas and P Novais - The Relevance of Deepfakes in the Administration of Criminal Justice. *See* also, on the GDPR, in this book E Magrani and P G F Silva - The Ethical and Legal Challenges of Recommender Systems Driven by Artificial Intelligence.

Samuel AL (1959) Some studies in machine learning using the game of checkers. IBM J Res Dev 3:210–229

Schölkopf B, Smola A (2002) Learning with kernels support vector machines, regularization, optimization, and beyond. MIT Press

Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, Schmitt S, Guez A, Lockhart E, Hassabis D, Graepel T, Lillicrap T, Silver D (2020) Mastering Atari, go, chess and shogi by planning with a learned model. Nature 588:604–609

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D (2016) Mastering the game of go with deep neural networks and tree search. Nature 529:484–489

Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y, Lillicrap T, Hui F, Sifre L, van den Driessche G, Graepel T, Hassabis D (2017) Mastering the game of go without human knowledge. Nature 550:354–359

Turing AM (1937) On computable numbers, with an application to the Entscheidungsproblem. Proc Lond Math Soc 2:230–265

Turing AM (1950) Computing machinery and intelligence. Mind 59:433–460

Vaswani A, Brain G, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Proces Syst 31:5998–6008

Weizenbaum J (1966) ELIZA—a computer program for the study of natural language communication between man and machine. Commun ACM 9:36–45