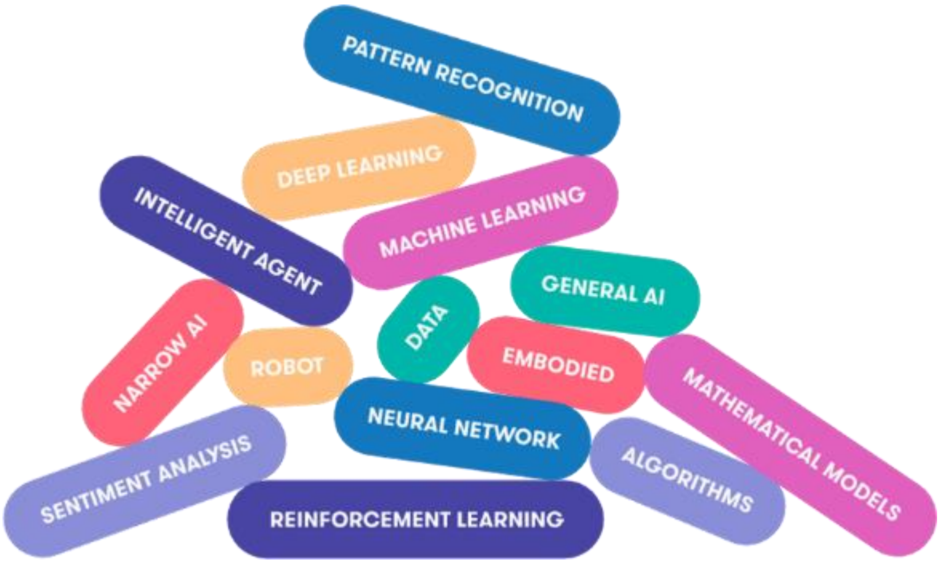
Module 6 – An Introduction to Artificial Intelligence

Will a robot take my job? How is Artificial Intelligence (AI) likely to change my job in the next ten years? Where are AI technologies being used right now and where will they come next? First, let's cover the following topics: How should we define AI; Which fields is AI related to; and the Philosophy behind Artificial Intelligence.



6.1 What is Artificial Intelligence?

In this part, we need to become familiar with the concept of AI by looking into its definition and some examples. As you have probably noticed, AI is currently a hot topic: media coverage and public discussion about AI is almost impossible to avoid. However, you may also have noticed that AI means different things to different people. For some, AI is about artificial life-forms that can surpass human intelligence, and for others, almost any data processing technology can be called AI. To set the scene, we'll discuss what AI is, how it can be defined, and what other fields or technologies are closely related. Before we do so, however, we'll highlight three applications of AI that illustrate different aspects of AI. We'll return to each of them throughout the course to deepen our understanding.

Application 1. Self-driving cars

Self-driving cars require a combination of AI techniques of many kinds: search and planning to find the most convenient route from A to B, computer vision to identify obstacles, and decision making under uncertainty to cope with the complex and dynamic environment. Each of these must work with almost flawless precision in order to avoid accidents. The same technologies are also used in other autonomous systems such as delivery robots, flying drones, and autonomous ships.

Implications: road safety should eventually improve as the reliability of the systems surpasses human level. The efficiency of logistics chains when moving goods should improve. Humans move into a supervisory role, keeping an eye on what's going on while machines take care of the driving. Since transportation is such a crucial element in our daily life, it is likely that there are also some implications that we haven't even thought about yet.

Application 2. Content recommendation

A lot of the information that we encounter in the course of a typical day is personalized. Examples include Meta, Twitter, Instagram, and other social media content; online advertisements; music recommendations on Spotify; movie recommendations on Netflix, HBO, and other streaming services. Many online publishers such as newspapers' and broadcasting companies' websites as well as search engines such as Google also personalize the content they offer. While the frontpage of the printed version of the New York Times or China Daily is the same for all readers, the frontpage of the online version is different for each user. The algorithms that determine the content that you see are based on AI.

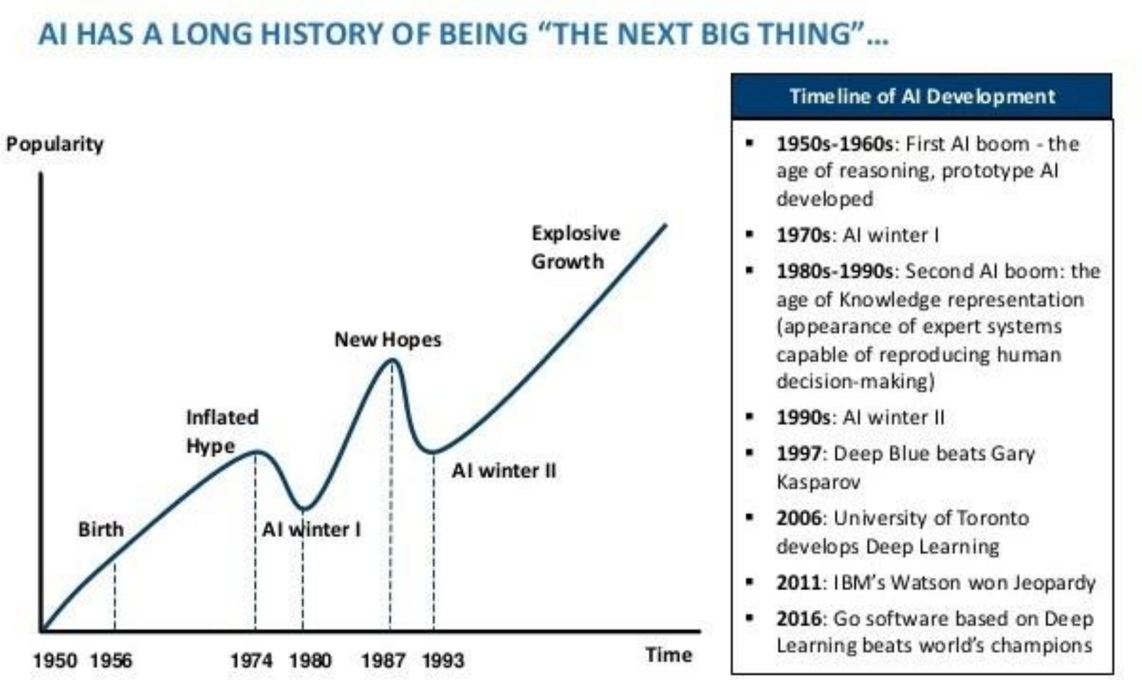
Implications: while many companies don't want to reveal the details of their algorithms, being aware of the basic principles helps you understand the potential implications: these involve so called filter bubbles, echo-chambers, troll factories, fake news, and new forms of propaganda.

Application 3. Image and video processing

Face recognition is already a commodity used in many customer, business, and government applications such as organizing your photos according to people, automatic tagging on social media, and passport control. Similar techniques can be used to recognize other cars and obstacles around an autonomous car, or to estimate wildlife populations, just to name a few examples. AI can also be used to generate or alter visual content. Examples already in use today include style transfer, by which you can adapt your personal photos to look like they were painted by Vincent van Gogh, and computer-generated characters in motion pictures such as Avatar, the Lord of the Rings, and popular Pixar animations where the animated characters replicate gestures made by real human actors.

Implications: when such techniques advance and become more widely available, it will be easy to create natural looking fake videos of events that are impossible to distinguish from real footage. This challenges the notion that “seeing is believing”.

History of AI booms and AI winters



The boom and bust cycle of AI research. Source: Actuaries Digital (2018).

Definition of Artificial Intelligence

What is, and what isn’t AI, is not an easy question. The popularity of AI in the media is in part due to the fact that people have started using the term when they refer to things that used to be called by other names. You can see almost anything from statistics and business analytics to manually encoded if-then rules called AI. Why is this so? Why is the public perception of AI so nebulous? Let’s look at a few reasons.

Firstly, there is no officially agreed definition. Even AI researchers have no exact definition of AI. The field is rather being constantly redefined when some topics are classified as non-AI, and new topics emerge. There's an old (geeky) joke that AI is defined as "cool things that computers can't do". The irony is that under this definition, AI can never make any progress: as soon as we find a way to do something cool with a computer, it stops being an AI problem. However, there is an element of truth in this definition. Fifty years ago, for instance, automatic methods for search and planning were considered to belong to the domain of AI. Nowadays such methods are taught to every computer science student. Similarly, certain methods for processing uncertain information are becoming so well understood that they are likely to be moved from AI to statistics or probability very soon.

The second reason is that AI holds its legacy of science fiction. The confusion about the meaning of AI is made worse by the visions of AI present in various literary and cinematic works of science fiction. Science fiction stories often feature friendly humanoid servants that provide overly-detailed factoids or witty dialogue, but can sometimes follow the steps of Pinocchio and start to wonder if they can become human. Another class of humanoid beings in sci-fi espouse sinister motives and turn against their masters in the vein of old tales of sorcerers' apprentices, going back to the Golem of Prague and beyond. Often the robothood of such creatures is only a thin veneer on top of a very humanlike agent, which is understandable as most fiction — even science fiction — needs to be relatable by human readers who would otherwise be alienated by intelligence that is too different and strange. Most science fiction is thus best read as metaphor for the current human condition, and robots could be seen as stand-ins for repressed sections of society, or perhaps our search for the meaning of life.

And finally, what seems easy is actually hard... and what seems hard is actually easy. Another source of difficulty in understanding AI is that it is hard to know which tasks are easy and which ones are hard. Look around and pick up an object in your hand, then think about what you did: you used your eyes to scan your surroundings, figured out where are some suitable objects for picking up, chose one of them and planned a trajectory for your hand to reach that one, then moved your hand by contracting various muscles in sequence and managed to squeeze the object with just the right amount of force to keep it between your fingers. It can be hard to appreciate how complicated all this is, but sometimes it becomes visible when something goes wrong: the object you pick is much heavier or lighter than you expected, or someone else opens a door just as you are reaching for the handle, and then you can find yourself seriously out of balance. Usually, these kinds of tasks feel effortless, but that feeling belies millions of years of evolution and several years of childhood practice. While easy for you, grasping objects by a robot is extremely hard, and it is an area of active study. Recent examples include Google's robotic grasping project, and a cauliflower picking robot.

By contrast, the tasks of playing chess and solving mathematical exercises can seem to be very difficult, requiring years of practice to master and involving our higher faculties and concentrated conscious thought. No wonder that some initial AI research concentrated on these kinds of tasks, and it may have seemed at the time that they encapsulate the essence of intelligence. It has since turned out that playing chess is very well suited to computers, which can follow fairly simple rules and compute many alternative move sequences at a rate of billions of computations a second. Computers beat the reigning human world champion in chess in the famous Deep Blue vs Kasparov matches in 1997. Could you have imagined that the harder problem turned out to be grabbing the pieces and moving them on the board without knocking it over? We will study the techniques that are used in playing games like chess or tic-tac-toe in the next section. Similarly, while in-depth mastery of mathematics requires (what seems like) human intuition and ingenuity, many (but not all) exercises of a typical high-school or college course can be solved by applying a calculator and simple set of rules.

An attempt at a definition more useful than the "what computers can't do yet" joke would be to list properties that are characteristic to AI, in this case autonomy and adaptivity.

1. Autonomy: the ability to perform tasks in complex environments without constant guidance by a user.
2. Adaptivity: the ability to improve performance by learning from experience.

When defining and talking about AI we have to be cautious as many of the words that we use can be quite misleading. Common examples are learning, understanding, and intelligence. You may well say, for example, that a system is intelligent, perhaps because it delivers accurate navigation instructions or detects signs of melanoma in photographs of skin lesions. When we hear something like this, the word "intelligent" easily suggests that the system is capable of performing any task an intelligent person is able to perform: going to the grocery store and cooking dinner, washing and folding laundry, and so on. Likewise, when we say that a computer vision system understands images because it is able to segment an image into distinct objects such as other cars, pedestrians, buildings, the road, and so on, the word "understand" easily suggest that the system also understands that even if a person is wearing a T-shirt that has a photo of a road printed on it, it is not okay to drive on that road (and over the person). In both of the above cases, we'd be wrong.

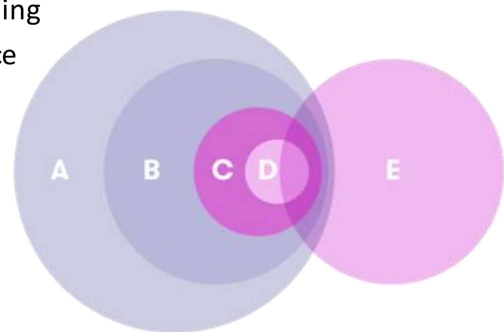
Marvin Minsky, a cognitive scientist and one of the greatest pioneers in AI, coined the term "suitcase word" for terms that carry a whole bunch of different meanings that come along even if we intend only one of them. Using such terms increases the risk of misinterpretations such as the ones above. It is important to realize that intelligence is not a single dimension like temperature. You can compare today's temperature to yesterday's, or the temperature in Helsinki to that in Rome, and tell which one is higher and which is lower. We even have a tendency to think that it is possible to rank people with respect to their intelligence — that's what the intelligence quotient (IQ) is supposed to do. However, in the context of AI, it is obvious that different AI systems cannot be compared on a single axis or dimension in terms of their intelligence. Is a chess-playing algorithm more intelligent than a spam filter, or is a music recommendation system more intelligent than a self-driving car? These questions make no sense. This is because artificial intelligence is narrow: being able to solve one problem tells us nothing about the ability to solve another, different problem.

Related fields of study

In addition to Artificial Intelligence, there are several other closely related topics that are good to know at least by name. These include machine learning, data science, and deep learning. Machine learning can be said to be a subfield of AI, which itself is a subfield of computer science. Machine learning enables AI solutions that are adaptive.

Taxonomy of AI

A. Computer Science
B. Artificial Intelligence
C. Machine Learning
D. Deep Learning
E. Data science



Concise definitions can be given as follows:

Machine learning refers to systems that improve their performance in a given task with more and more experience or data.

Deep learning is a subfield of machine learning, which itself is a subfield of AI, which itself is a subfield of computer science. The depth of deep learning refers to the complexity of a mathematical model. The increased computing power of modern computers has allowed researchers to increase this complexity to reach levels that appear not only quantitatively but also qualitatively different from before. Science often involves a number of progressively more special subfields, subfields of subfields, and so on. This enables researchers to zoom into a particular topic so that it is possible to catch up with the ever-increasing amount of knowledge accrued over the years, and produce new knowledge on the topic — or sometimes, correct earlier knowledge to be more accurate.

Data science is a recent umbrella term that includes machine learning and statistics, certain aspects of computer science including algorithms, data storage, and web application development. Data science is also a practical discipline that requires understanding of the domain in which it is applied in, for example, business or science: its purpose (what "added value" means), basic assumptions, and constraints. Data science solutions often involve at least a pinch of AI (but usually not as much as one would expect from the media headlines).

Robotics means building and programming robots so that they can operate in complex, real-world scenarios. In a way, robotics is the ultimate challenge of AI since it requires a combination of virtually all areas of AI. For example, computer vision and speech recognition for sensing the environment; natural language processing, information retrieval, and reasoning under uncertainty for processing instructions and predicting consequences of potential actions; cognitive modelling and affective computing (systems that respond to expressions of human feelings or that mimic feelings) for interacting and working together with humans. Many of the robotics-related AI problems are best approached by machine learning, which makes machine learning a central branch of AI for robotics.

What is a robot? In brief, a robot is a machine comprising sensors (which sense the environment) and actuators (which act on the environment) that can be programmed to perform sequences of actions. People used to science-fictional depictions of robots will usually think of humanoid machines walking with an awkward gait and speaking in a metallic monotone. Most real-world robots currently in use look very different as they are designed according to the application. Most applications would not benefit from the robot having human shape, just like we don’t have humanoid robots to do our dishwashing but machines in which we place the dishes to be washed by jets of water. It may not be obvious at first sight, but any kind of vehicles that have at least some level of autonomy and include sensors and actuators are also counted as robotics. On the other hand, software-based solutions such as a customer service chatbot, even if they are sometimes called software robots, aren’t counted as (real) robotics.

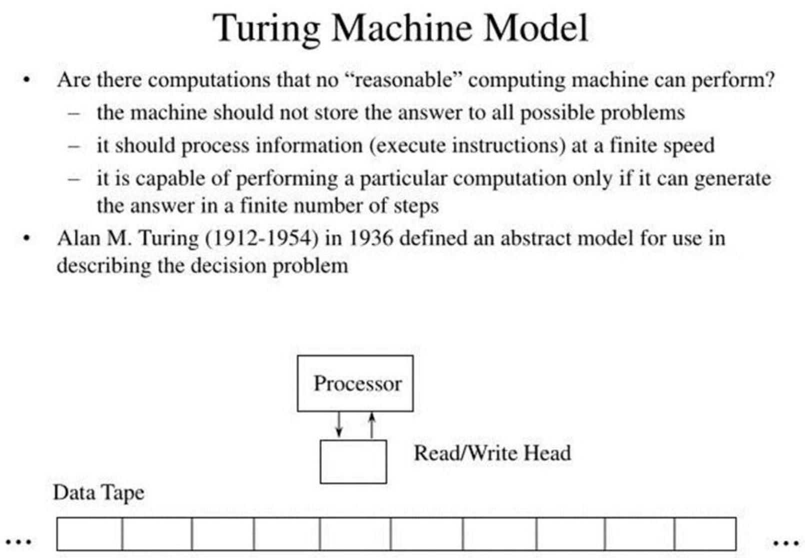
Philosophy behind Artificial Intelligence

The very nature of the term Artificial Intelligence brings up philosophical questions whether intelligent behavior implies or requires the existence of a mind, and to what extent is consciousness replicable as computation.

The Turing test

Alan Turing (1912-1954) was an English mathematician and logician. He is rightfully considered to be the father of computer science. Turing was fascinated by intelligence and thinking, and the possibility of simulating them by machines. Turing's most prominent contribution to Artificial Intelligence is his "Imitation Game", which later became known as the Turing test. In the test, a human interrogator interacts with two players, A and B, by exchanging written messages in a chat. If the interrogator cannot determine which player, A or B, is a computer and which is a human, the computer is said to pass the test. The argument is that if a computer is indistinguishable from a human in a general natural language conversation, then it must have reached human-level intelligence.

What Turing meant by the test is very much similar to the aphorism by Forrest Gump: "stupid is as stupid does". Turing's version would be "intelligent is as intelligent says". In other words, an entity is intelligent if it cannot be distinguished from another intelligent entity by observing its behavior. Turing just constrained the set of behaviors into discussion so that the interrogator can't base her or his decision on appearances. One philosophical problem this addresses however, is "does being human-like mean you are intelligent?" One criticism of the Turing test as a test for intelligence is that it may actually measure whether the computer behaves like a human more than whether it is intelligent. The test has indeed been passed by computer programs that keep changing the subject, make plenty of spelling errors, and sometimes refuse to respond at all. A famous example is Eugene Goostman, a 13-year-old Ukrainian boy who constantly tried to avoid answering questions by making jokes and changing the subject, even though he was actually a chatbot.



The Chinese room experiment - The Chinese room argument

The idea that intelligence is the same as intelligent behavior has been challenged by some. The best-known counter-argument is John Searle's Chinese Room thought experiment. Searle describes an experiment where a person who doesn't know Chinese is locked in a room. Outside the room is a person who can slip notes written in Chinese inside the room through a mail slot. The person inside the room is given a big manual where she can find detailed instructions for responding to the notes she receives from the outside. Searle argued that even if the person outside the room gets the impression that he is in a conversation with another Chinese-speaking person, the person inside the room does not understand Chinese. Likewise, his argument continues, even if a machine behaves in an intelligent manner, for example, by passing the Turing test, it doesn't follow that it is intelligent or that it has a mind in the way that a human has. The word “intelligent” can also be replaced by the word “conscious” and a similar argument can be made.

Is a self-driving car intelligent? The Chinese Room argument goes against the notion that intelligence can be broken down into small mechanical instructions that can be automated, as stated by Turing. A self-driving car is an example of an element of intelligence (driving a car) that can be automated. The Chinese Room argument suggests that this, however, isn’t really intelligent thinking: it just looks like it. Going back to the above discussion on suitcase words, the AI system in the car doesn’t see or understand its environment, and it doesn’t know how to drive safely, in the way a human being sees, understands, and knows. According to Searle this means that the intelligent behavior of the system is fundamentally different from actually being intelligent.

How much does philosophy matter in practice? The definition of intelligence, natural or artificial, and consciousness appears to be extremely evasive and leads to apparently never-ending discourse. In intellectual company, this discussion can be quite enjoyable. However, as John McCarthy pointed out, the philosophy of AI is “unlikely to have any more effect on the practice of AI research than philosophy of science generally has on the practice of science”. Thus, we’ll continue investigating systems that are helpful in solving practical problems without asking too much whether they are intelligent or just behave as if they were.

When reading the news, you might see the terms “general” and “narrow” AI. So, what do these terms mean? Narrow AI refers to AI that handles one task. General AI, or Artificial General Intelligence (AGI) refers to a machine that can handle any intellectual task. All the AI methods we use today fall under narrow AI, with general AI being in the realm of science fiction. In fact, the ideal of AGI has been all but abandoned by the AI researchers because of lack of progress towards it in more than 50 years despite all the effort. In contrast, narrow AI makes progress in leaps and bounds.

A related dichotomy is “strong” and “weak” AI. This boils down to the above philosophical distinction between being intelligent and acting intelligently, which was emphasized by Searle. Strong AI would amount to a mind that is genuinely intelligent and self-conscious. Weak AI is what we actually have, namely systems that exhibit intelligent behaviors despite being mere computers.

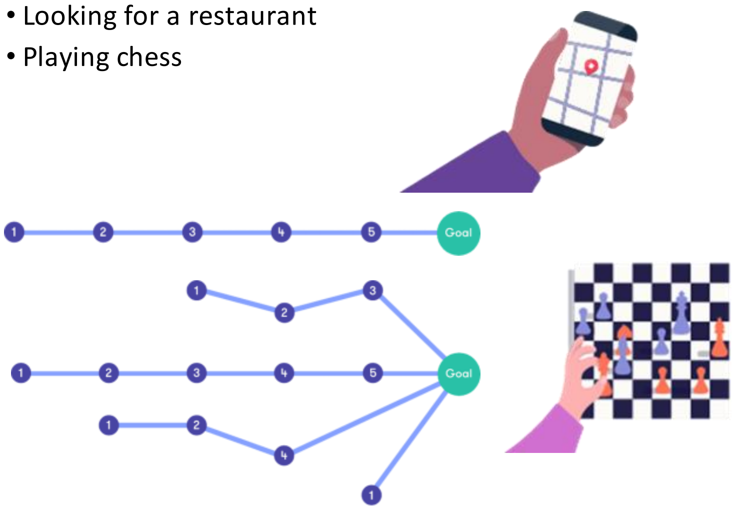
AI is arguably as old as computer science. Long before we had computers, people thought of the possibility of automatic reasoning and intelligence. One of the great thinkers who considered this question was Alan Turing. In addition to the Turing test, his contributions to AI, and more generally to computer science, include the insight that anything that can be computed, i.e., calculated using either numbers or other symbols, can be automated. Turing designed a very simple device that can compute anything that is computable, known as the Turing machine. While it is a theoretical model that isn't practically useful, it led Turing to the invention of programmable computers that can be used to carry out different tasks depending on what they were programmed to do. So instead of having to build a different device for each task, we use the same computer for many tasks. This is the idea of programming. Today this invention sounds trivial but in Turing's days it was far from it. Some of the early programmable computers were used during World War II to crack German secret codes, a project where Turing was also personally involved.

The actual term Artificial Intelligence was coined by John McCarthy (1927-2011) — who is often also referred to as the Father of AI. The term became established when it was chosen as the topic of a summer seminar, known as the Dartmouth conference, which was organized in 1956 at Dartmouth College in New Hampshire. In the proposal to organize the seminar, McCarthy continued with Turing's argument about automated computation. John McCarthy's proposal contains the following crucial statement about AI: "The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

In other words, any element of intelligence can be broken down into small steps so that each of the steps is as such so simple and mechanical that it can be written down as a computer program. This statement was, and is still today, a conjecture, which means a hypothesis on a tentative basis that can't really be proven to be true. Nevertheless, the idea is absolutely fundamental when it comes to the way we think about AI. For example, it shows that McCarthy wanted to bypass any arguments in the spirit of Searle's Chinese Room where "Intelligence is intelligence even if the system that implements it is just a computer that mechanically follows a program".

Search problems and problem solving

• Looking for a restaurant
• Playing chess



6.2 Game Theory

Search algorithms don't necessarily feel like very cool AI methods. However, they can be used to solve tasks that most of us would admit require intelligence like navigation or playing chess. In this section, we will cover the following topics: Search and problem solving; Solving problems with AI; and Search and games.

Search problems and problem solving

Many problems can be phrased as search problems. This requires that we start by formulating the alternative choices and their consequences. For instance, search in practice: getting from A to B. Imagine you're in a foreign city, at some address (say a hotel) and want to use public transport to get to another address (a nice restaurant, perhaps). What do you do? If you are like many people, you pull out your smartphone, type in the destination and start following the instructions. This question belongs to the class of search and planning problems. Similar problems need to be solved by self-driving cars, and for playing games. In the game of chess, for example, the difficulty is not so much in getting a piece from A to B as keeping your pieces safe from the opponent.

Often there are many different ways to solve the problem, some of which may be more preferable in terms of time, effort, cost or other criteria. Different search techniques may lead to different solutions, and developing advanced search algorithms is an established research area. We will not focus on the actual search algorithms. Instead, we emphasize the first stage of the problem-solving process: defining the choices and their consequences, which is often far from trivial and can require careful thinking. We also need to define what our goal is, or in other words, when we can consider the problem solved. After this has been done, we can look for a sequence of actions that leads from the initial state to the goal. In this section, we will discuss two kinds of problems:

- Search and planning in static environments with only one agent;
- Games with two-players (agents) competing against each other.

These categories don't cover all possible real-world scenarios, but they are generic enough to demonstrate the main concepts and techniques. Before we address complex search tasks like navigation or playing chess, let us start from a much-simplified model in order to build up our understanding of how we can solve problems by AI.

The rowboat puzzle

The rowboat puzzle — The fox, chicken, and grain

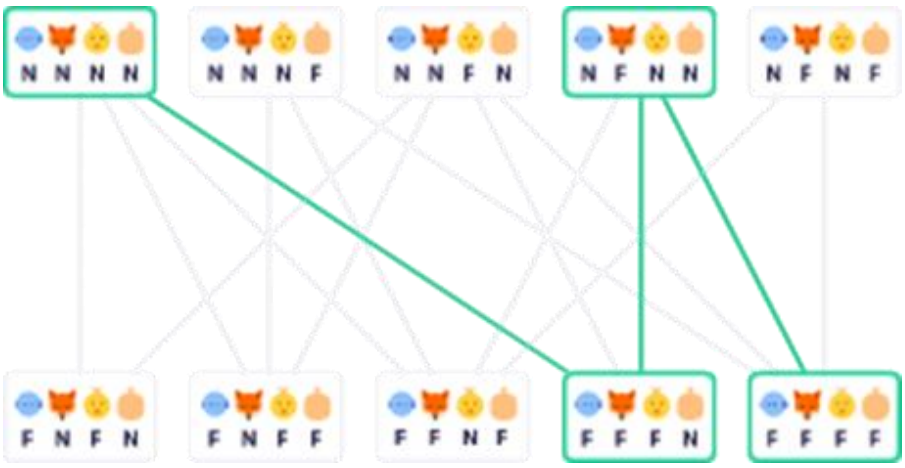
We'll start from a simple puzzle to illustrate the ideas. A robot on a rowboat needs to move three pieces of cargo across a river: a fox, a chicken, and a sack of grain. The fox will eat the chicken if it has the chance, and the chicken will eat the grain if it has the chance, and neither is a desirable outcome. The robot is capable of keeping the animals from doing harm when it is near them, but only the robot can operate the rowboat and only two of the pieces of cargo can fit on the rowboat together with the robot. How can the robot move all of its cargo to the opposite bank of the river?

States of the chicken crossing puzzle

State	Robot	Fox	Chicken	Chicken-feed
NNNN	Near side	Near side	Near side	Near side
NNNF	Near side	Near side	Near side	Far side
NNFN	Near side	Near side	Far side	Near side
NNFF	Near side	Near side	Far side	Far side
NFNN	Near side	Far side	Near side	Near side
NFNF	Near side	Far side	Near side	Far side
NFFN	Near side	Far side	Far side	Near side
NFFF	Near side	Far side	Far side	Far side
FNNN	Far side	Near side	Near side	Near side
FNNF	Far side	Near side	Near side	Far side
FNFN	Far side	Near side	Far side	Near side
FNFF	Far side	Near side	Far side	Far side
FFNN	Far side	Far side	Near side	Near side
FFNF	Far side	Far side	Near side	Far side
FFFN	Far side	Far side	Far side	Near side
FFFF	Far side	Far side	Far side	Far side

After having formulated the alternative states and transitions between them, the rest becomes a mechanical task to find a path from the initial state NNNN to the final state FFFF. Your natural intelligence will surely solve the puzzle: one possible path proceeds from the robot taking the fox and the chicken to the other side i.e., NNNN to FFFN, thence to NFNN, the robot taking the chicken back on the starting side and finally to FFFF, when the robot can now move the chicken and the grain to the other side. In three transitions, the puzzle is solved. There is also an easy version of the rowboat puzzle. If you have heard this riddle before, you might know that it can be solved even with less space on the boat in seven transitions. Intuitively, the strategy is to move the chicken on the other side first, and then go back get either the fox or the feed, and take it to the far side too. The robot then takes the chicken back to the near side to save it from being eaten or from eating the feed, and takes the other remaining object (fox or feed) from the near side to the far side. Finally, the robot goes to fetch the chicken and takes it to the far side to reach the goal.

State	Robot	Fox	Chicken	Chicken-feed
NNNN	Near side	Near side	Near side	Near side
NNNF	Near side	Near side	Near side	Far side
NNFN	Near side	Near side	Far side	Near side
NFNN	Near side	Far side	Near side	Near side
NFNF	Near side	Far side	Near side	Far side
FNFN	Far side	Near side	Far side	Near side
FNFF	Far side	Near side	Far side	Far side
FFNF	Far side	Far side	Near side	Far side
FFFN	Far side	Far side	Far side	Near side
FFFF	Far side	Far side	Far side	Far side



NNNN -> FFFN -> NFNN -> FFFF

Terminology: State– Transition– Path - Cost

For more complex problems, where the number of possible solutions grows in the thousands and in the millions, our systematic or mechanical approach will shine since the hard part will be suitable for a simple computer to do. To formalize a planning problem, we use concepts (key terminology) such as the state space, transitions, and costs.

The state space means the set of possible situations. In the chicken-and-fox crossing puzzle, the state space consisted of ten allowed states NNNN through to FFFF. If the task is to navigate from place A to place B, the state space could be the set of locations defined by their (x,y) coordinates that can be reached from the starting point A. Or we could use a constrained set of locations, for example, different street addresses so that the number of possible states is limited.

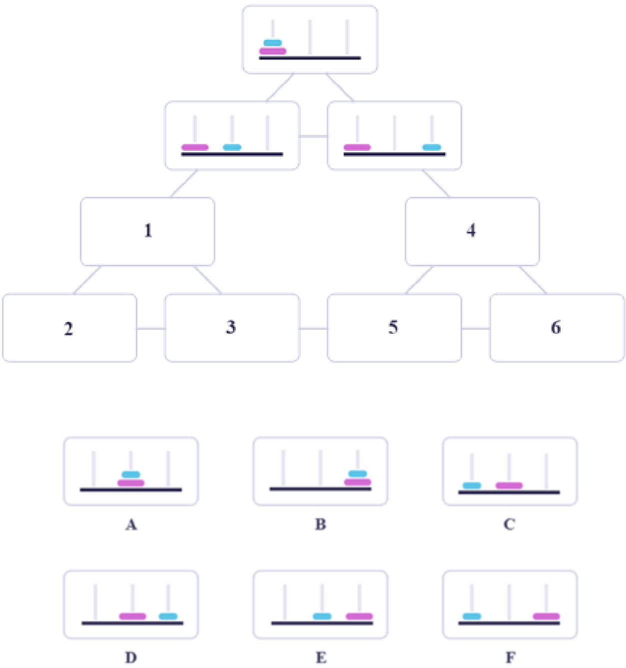
Transitions are possible moves between one state and another, such as NNNN to FNFN. It is important to note that we only count direct transitions that can be accomplished with a single action as transitions. A sequence of multiple transitions, for example, from A to C, from C to D, and from D to B (the goal), is a path rather than a transition.

Costs refer to the fact that, oftentimes the different transitions aren't all alike. They can differ in ways that make some transitions more preferable or cheaper (in a not necessarily monetary sense of the word) and others more costly. We can express this by associating with each transition a certain cost. If the goal is to minimize the total distance travelled, then a natural cost is the geographical distance between states. On the other hand, the goal could actually be to minimize the time instead of the distance, in which case the natural cost would obviously be the time. If all the transitions are equal, then we can ignore the costs.

The Towers of Hanoi Another is another similar puzzle is this category. The game involves three pegs, and two discs: one large, and one small. In the initial state, both discs are stacked in the first (leftmost) peg. The goal is to move the discs to the third peg. You can move one disc at a time, from any peg to another, as long as there is no other disc on top of it. It is not allowed to put a larger disc on top of a smaller disc.

The Towers of Hanoi

Attempting to solve the Tower of Hanoi exercises parts of the brain that help to manage time, present a business plan or make complex arguments. Not only is the Tower of Hanoi beneficial in physical and mental terms, but also in terms of certain jobs. The Tower of Hanoi is commonly used by psychologists to research and examine problem solving skills. Problem solving skills can be acquired by calculating moves and strategies while at the same time predicting possible outcomes. The recursive rule of the Tower of Hanoi is studied and applied in computer programming and algorithms which helps to reduce the amount of time it takes to create a program.



Solving problems with AI

As computers developed to the level where it was feasible to experiment with practical AI algorithms in the 1950s, the most distinctive AI problems were games and entertainment. Games provided a convenient restricted domain that could be formalized easily. Board games such as checkers, chess, and recently quite prominently Go — an extremely complex strategy board game originating from China at least 2500 years ago — have inspired countless researchers, and continue to do so.

Closely related to games, search and planning techniques were an area where AI led to great advances in the 1960s: algorithms with names such as the Minimax algorithm or Alpha-Beta Pruning, which were developed then, are still the basis for game playing AI, although of course more advanced variants have been proposed over the years. Here, we will study games and planning problems on a conceptual level.

Search and games

The simplest scenario, which we will focus on for the sake of clarity, are two-player, perfect-information games such as tic-tac-toe and chess.

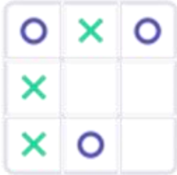
The Minimax algorithm

Maxine and Minnie are true game enthusiasts. They just love games. Especially two-person, perfect information games such as tic-tac-toe or chess. One day they were playing tic-tac-toe. Maxine, or Max as her friends call her, was playing with X. Minnie, or Min as her friends call her, had the Os. Min had just played her turn on the board. Max was looking at the board and contemplating her next move, as it was her turn, when she suddenly buried her face in her hands in despair, looking quite like Garry Kasparov playing Deep Blue in 1997. Yes, Min was close to getting three Os on the top row, but Max could easily put a stop to that plan. So why was Max so pessimistic? By analyzing the game step-by-step, using a game tree, you will find the solution.

Game theory

• Tic-Tac-Toe

• Chess



Game trees

To solve games using AI, we need to introduce the concept of a game tree. The different states of the game are represented by nodes in the game tree, very similar to the above planning problems. The idea is just slightly different. In the game tree, the nodes are arranged in levels that correspond to each player’s turns in the game so that the root node of the tree (usually depicted at the top of the diagram) is the beginning position in the game. In tic-tac-toe, this would be the empty grid with no Xs or Os played yet. Under root, on the second level, there are the possible states that can result from the first player’s moves, be it X or O. We call these nodes the children of the root node.

Each node on the second level, would further have as its children’s nodes the states that can be reached from it by the opposing player’s moves. This is continued, level by level, until reaching states where the game is over. In tic-tac-toe, this means that either one of the players gets a line of three and wins, or the board is full and the game ends in a tie. In order to be able to create game AI that attempts to win the game, we attach a numerical value to each possible end result. To the board positions where X has a line of three so that Max wins, we attach the value +1, and likewise, to the positions where Min wins with three Os in a row we attach the value -1. For the positions where the board is full and neither player wins, we use the neutral value 0 (it doesn’t really matter what the values are as long as they are in this order so that Max tries to maximize the value, and Min tries to minimize it). The only move Max can make is to put an X in the upper middle square to avoid immediate defeat. She, however, has all the reason to be serious since by playing in the bottom-right corner, Min can guarantee a win. The inevitable victory of Min, who is — of course — a strategic player, can also be seen from the value of the game, which is –1 (value of the root node level 1). The values on the second level are 0, 0, and –1. The values on the third level are –1, 0, –1, 0, –1, –1, which are the same as the values on the bottom level.

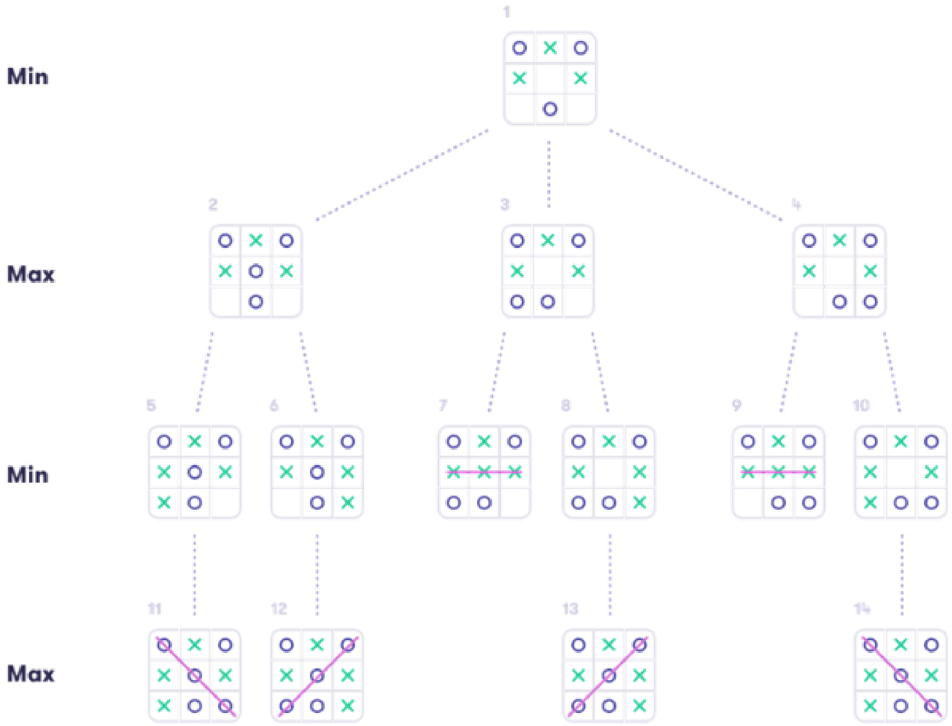
The most important lesson in this section is to apply the above kind of reasoning repeatedly to determine the result of the game in advance from any board position. The value of the root node, which is said to be the value of the game, tells us who wins (and how much, if the outcome is not just plain win or lose): Max wins if the value of the game is +1, Min if the value is –1, and if the value is 0, then the game will end in a draw. In other games, the value may also take other values (such as the monetary value of the chips in front of you in poker for example). This all is based on the assumption that both players choose what is best for them and that what is best for one is the worst for the other (so called zero-sum game).

Having determined the values of all the nodes in the game tree, the optimal moves can be deduced: at any Min node (where it is Min's turn), the optimal choice is given by the child node whose value is minimal, and conversely, at any Max node (where it is Max's turn), the optimal choice is given by the child node whose value is maximal. Sometimes there are many equally good choices that are, well, equally good, and the outcome will be the same no matter which one of them is picked.

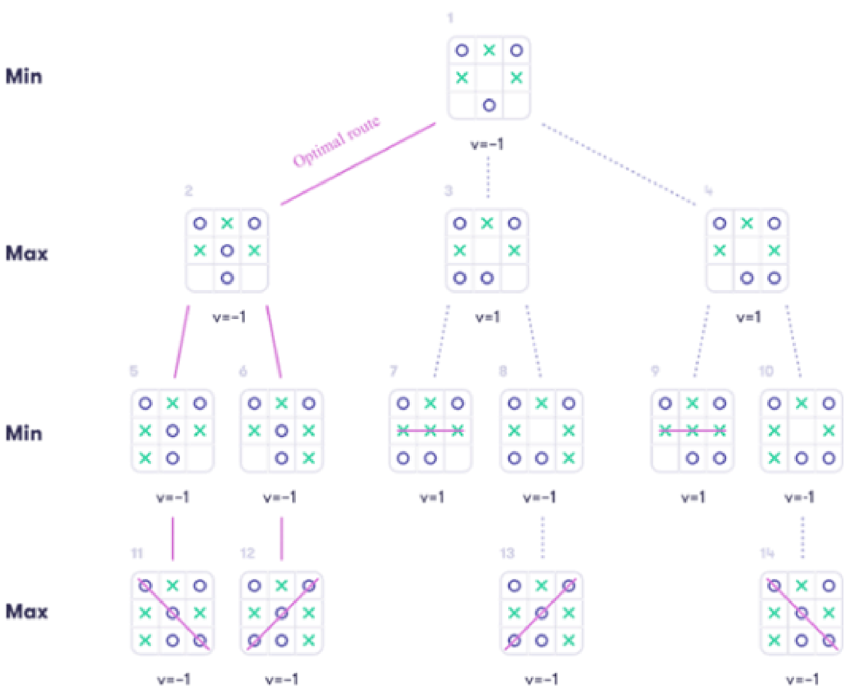
We can exploit the above concept of the value of the game to obtain an algorithm called the Minimax algorithm. It guarantees optimal game play in, theoretically speaking, any deterministic, two-person, perfect-information zero-sum game. Given a state of the game, the algorithm simply computes the values of the children of the given state and chooses the one that has the maximum value if it is Max's turn, and the one that has the minimum value if it is Min's turn. Such games include tic-tac-toe, connect four, chess, Go, etc. Rock-paper-scissors is not in this class of games since it involves information hidden from the other player; nor are Monopoly or backgammon which are not deterministic.

Note — Minimax (sometimes called MinMax, MM or saddle point) is a decision rule used in artificial intelligence, decision theory, game theory, statistics, and philosophy for minimizing the possible loss for a worst case (maximum loss) scenario. When dealing with gains, it is referred to as "maximin" — to maximize the minimum gain. Originally formulated for n-player zero-sum game theory, covering both the cases where players take alternate moves and those where they make simultaneous moves, it has also been extended to more complex games and to general decision-making in the presence of uncertainty.

Game Tree



The Minimax algorithm -- Determining who wins



In many games, however, the game tree is simply way too big to traverse in full. For example, in chess the average branching factor, i.e., the average number of children (available moves) per node is about 35. That means that to explore all the possible scenarios up to only two moves ahead, we need to visit approximately 35 x 35 = 1 225 nodes – probably not your favorite pencil-and-paper homework exercise. A look-ahead of three moves requires visiting 42 875 nodes; four moves 1 500 625; and ten moves 2758547353515625 (that's about 2.7 quadrillion) nodes.

A few more tricks are needed to manage massive game trees. If we can afford to explore only a small part of the game tree, we need a way to stop the Minimax algorithm before reaching an end-node, i.e., a node where the game is over and the winner is known. This is achieved by using a so-called heuristic evaluation function that takes as input a board position, including the information about which player's turn is next, and returns a score that should be an estimate of the likely outcome of the game continuing from the given board position. Good heuristics for chess, for example, typically count the amount of material (pieces) weighted by their type: the queen is usually considered worth about two times as much as a rook, three times a knight or a bishop, and nine times as much as a pawn. The king is of course worth more than all other things combined since losing it amounts to losing the game. Further, occupying the strategically important positions near the middle of the board is considered an advantage and the heuristics assign higher value to such positions.

The Minimax algorithm presented above requires minimal changes to obtain a depth-limited version where the heuristic is returned at all nodes at a given depth limit: the depth simply refers to the number of steps that the game tree is expanded before applying a heuristic evaluation function. It may look like we have a method to solve any problem by specifying the states and transitions between them, and finding a path from the current state to our goal. Alas, things get more complicated when we want to apply AI in real world problems. Basically, the number of states in even a moderately complex real-world scenario grows out of hand, and we can't find a solution by exhaustive search (brute force) or even by using clever heuristics. Moreover, the transitions which take us from one state to the next when we choose an action are not deterministic. This means that whatever we choose to do will not always completely determine the outcome because there are factors that are beyond our control, and that are often unknown to us.

The algorithms we have discussed above can be adapted to handle some randomness, for example randomness in choosing cards from a shuffled deck or throwing dice. This means that we will need to introduce the concept of uncertainty and probability (statistics). Only thus we can begin to approach real-world AI instead of simple puzzles and games. Game theory and statistics combined thus set the scene for fundamental AI methods, i.e. machine learning and deep learning.