1. Introduction to Artificial Intelligence

1.1 What is AI?

People have wondered if machines could be intelligent even before the advent of modern computers. Is it really possible for a computer behave in an intelligent way? Or to think the way that humans do? Some would argue that while computers may be capable of impressive pure computation, this is somehow different from true thought (whatever that is). Others see computers as simply another kind of medium through which the mysterious patterns of thought may pass. In this text, we will keep these philosophical questions in the background and instead focus on the practiced disciplines of **computational artificial intelligence and machine learning**. Careful study of these computational techniques can lead to insights about the philosophical questions and, indeed, understanding the computational side of things makes the bigger picture clearer.

The study of artificial intelligence encompasses a wide variety of subfields that are concerned with making computers behave in intelligent ways. These subfields include **logical reasoning, robotics, machine learning, planning, natural language processing, computer vision, probabilistic reasoning**, and more. This variety of topics makes AI something of an eclectic collection of ideas and techniques, though all share the greater goal of making progress toward artificially intelligent computers and machines.

1.2 Defining Intelligence

Before we begin looking closely at particular AI methods, it is helpful to think about how we actually define *intelligent behavior*. Eugene Fink of Carnegie Mellon University has a nice, partially humorous collection of quotes, several of which are listed here:

- **Herbert Simon**: We call programs intelligent if they exhibit behaviors that would be regarded intelligent if they were exhibited by human beings.
- **Elaine Rich and Kevin Knight**: AI is the study of how to make computers do things at which, at the moment, people are better.
- **Douglas Baker**: AI is the attempt to make computers do what people think computers cannot do.

- **Astro Teller**: AI is the attempt to make computers do what they do in the movies.

1.2.1 Human-Like External Behavior

As you can tell from several of these quotes, some people think that human behavior is a good benchmark for artificial intelligence. If an AI system behaves like a human would in a similar situation, then we could say that it is intelligent. While we humans do exhibit behavior that is sometimes perhaps intelligent, we are also quite adept at behaving irrationally and making poor decisions. Because of this, we may want to have a measure of intelligence that is separate from human abilities.

1.2.2 Human-Like Internal Algorithms

What if an AI system were to behave outwardly as a human, but have a vastly different internal way of producing that behavior? That is, should we be worried about the actual algorithm used by the AI system and require it to be human-like to be considered intelligent?

This discussion comes up often and one famous example is from the aftermath of IBM's Deep Blue computer defeating Gary Kasparov, the world chess champion in 1997. The chess-playing techniques used by Deep Blue were very different from the methods that human grandmasters describe as their own. Deep Blue was well-suited to perform many brute force calculations about possible future board positions and then choose a move based on the best predicted outcome. Our understanding is that humans typically do not consciously calculate as many future board positions, but are able to recognize larger board patterns and then use those patterns to look ahead at a more limited number of future game positions. After the competition many critics argued that Deep Blue was not a good example of AI because of these differences.

Most AI researchers do not worry about making their systems operate exactly like humans do internally. Human cognitive systems are often used as a starting blueprint for designing and implementing AI systems (for example, computer vision systems are modeled after biological vision systems), but following human methods is not a requirement for progress in AI research. On the other hand, **cognitive scientists** often do work that is related to AI, but they are interested in modeling human cognition through computational means. These two fields are related, but have slightly different end goals.

AI researchers do often compare their systems to human performance on the same tasks, as this is a relatively easy way to let others know how well the system is doing. This is a rather informal metric, though, since different humans may have quite different abilities on the given task. Nevertheless, it is a good starting point for showing the promise of a newly-proposed AI method.

1.2.3 The Turing Test

The most famous measure of AI is the Turing Test. The test is named after Alan Turing, a computer science pioneer and one of the founders of the modern study of artificial intelligence, who proposed the idea in 1950. Turing suggested that a computer could be considered intelligent if it were able to imitate a human in conversation so that a human judge were not able to tell it was a computer. The most common modern version of the Turing Test is a setup where a human judge sits in one room and has two electronic conversations: one with a computer imitating a human and one with an actual human. The interface is such that the judge has no other information about each conversation partner other than the text responses received. The judge's task is to determine which conversation partner is the human and which is the computer. If the judge is unable to differentiate the partners in a statistically reliable way, then the computer is said to have passed the Turing Test.

Of course, the choice of human judge and the time of the conversation are important factors in whether the test is meaningful or not. Non-expert judges might be more easily fooled by various conversational tricks that can be performed by very simple programs. A very early conversational program called **ELIZA** attempted to mimic a psychoanalyst. Many of its responses are very simple manipulations of past human input. For example, you might ask "what day is it?" and ELIZA might respond with, "Is it because of the people you hang around with that you say what day is it?" This kind of nonsensical text manipulation might fool a particularly uncritical judge, but it is clearly not an example that most experts would agree shows meaningful artificial intelligence.

1.2.4 Objective Intelligence

Comparing against human performance and using human judges are reasonable starting benchmarks for measuring intelligence, but we may also want to consider how an AI system compares to a more objective intelligence function. We can imagine some kind of function that evaluates the performance of the AI versus optimal performance, for some definition of "optimal". For example, for a game-playing AI, the evaluation function could score the system based on how often it chooses the best game move.

Determining a measure of objective intelligence may be very difficult for some problem domains. For example, an AI used to fly airplanes could be measured by the percentage of safe trips it performs, but we would really rather know if the system were able to provide good plane-flying performance even under all the conditions that it has not yet been tested in. The set of possible environmental conditions encountered by a plane on even short trips would be so large that testing the plane on all of them would be infeasible.

1.3 An Example AI Problem

Many car companies are optimistic that automated cars will be road-ready within the next five to ten years. Driving a multi-thousand pound chunk of metal down the freeway at 70 mph will certainly require passengers to feel a good deal of trust in AI! As a way to introduce some of the common problems encountered in AI, let's consider some of the sub-problems of creating an automated car.

Figure 1.1: A self-driving Audi that made it up Pike's Peak in 27 minutes in 2010. (engaget.com)

Perhaps the first thing to consider for an automated car is the fact that it, of course, is a physical vehicle operating in the real world and not just a software simulation. This introduces many problems from **robotics**. Our automated car must have an array of sensors to gather data about its surroundings. The sensor data must then be processed in a timely manner and sent to the various car sub-systems.

The car may be equipped with one or more cameras that are constantly recording video streams. Part of our car AI may need to use techniques from **computer vision** to split single video images in meaningful ways to separate out the different distinct objects in view. The objects present in the image must then be classified according to type (stop sign, bicyclist, dog, tree, etc.). Once the car can recognize these different objects, then it can decide a reasonable course of action (continue ahead, brake immediately, etc.).

Our car will need to plan its route from point A to point B. It may be equipped with mapping software that has destinations stored as nodes in a graph with connecting edges representing roadways. The car could then **search** through the abstract graph to find a path to the desired destination. There may be an extremely large number of possible paths, though, so it would be helpful for the car to search through the possibilities in a smart way.

The real-world environment that our automated car operates in is likely to have nearly constant unexpected events occurring. A dog could run across the path of the car. Hail could block the forward-facing camera. Heavy wind could push the car out of its lane. We need to make sure our car can handle these unusual events in a robust way. That is, it should be able to handle uncertainty about what to expect next. If an unlikely event occurs, we want our car to be able to handle it using **probabilistic reasoning**.

How can we make our car drive well in the first place? We could hire a huge team of human experts to enter in all the required **logic and reasoning** for driving a car. A logical rule might be: *If there is a bicyclist within 10 meters of the front of the car, then slow down*. To enter in all the possible logical driving rules would require the largest group of driver's ed instructors in the world. Instead it would be much easier, more efficient, and better to have the car use **machine learning** to learn how to drive. It could learn by directly observing humans driving or perhaps by using a system of rewards and punishments. Each meter safely driven is worth one point. Each collision with a pothole is worth -100 points. Each collision with another car is worth -1,000,000,000 points. After a substantial learning period, hopefully our car will be able to drive safely.

1.4 Problems

Problem 1.1 Come up with your own definition of intelligence. Do computers currently fit the definition in any way? What about animals like gorillas, dogs, mice, or fish?

Problem 1.2 What is one problem that almost everyone would agree requires true intelligence to solve? When do you suppose computers will be able to solve it?

Problem 1.3 Are there well-defined problems that you think computers will never solve? Are these problems solvable by humans?

Problem 1.4 Computers are currently (2015) superior to humans in many games including chess and the quiz show Jeopardy. Humans still dominate the game of Go, though. If you were designing a game to specifically give an advantage to human players, how would you design it?

2. Agents, Environments, and Problem States

Before we begin to think about and work on the various algorithms employed in AI systems, it is helpful to define the types of problems we are interested in solving. Using the general framework described in this chapter, we can setup problems and reason about their solutions using a common language.

What do we want our AI systems to do? The answer is *a wide range of things!* We want AI systems to automatically drive cars as in the example from Chapter 1, but we also want systems to analyze images and video, to understand and use human languages, to recognize what we say out loud, to diagnose diseases with pinpoint accuracy, and more. These problems cover a broad scope of system capabilities, so it is helpful to extract out their common features in order to develop their solutions in a slightly more abstract way. In particular, we will think of our AI systems as **agents** making decisions about how to interact with various types of **environments** that consist of well-defined **states**.

2.1 Agents

To think about how to setup problems in AI, it can be helpful to imagine we are in the position of the computer or robot attempting to solve a problem that requires intelligence of some kind. What information do we use to make a decision about what to do next? What are our possible options for our next actions?

It is traditional in AI to think of robots or software AI systems as **agents** that make decisions and take actions based on their current environmental inputs. In the next section we will think about how environments for AI problems might be represented.

> **Definition 2.1.1 — Agent.** An AI system that is designed to respond to an environment and make decisions or choose actions in order to solve a problem or reach a goal.

The agent then operates much like mathematical function: it receives inputs from the environment and then produces output as an action or as a decision. Here are several examples of AI agents with brief descriptions of their environmental inputs and their outputs.

■ **Example 2.1 — Chess-Playing Agent.** A chess-playing agent might receive environmental input about the current configuration of the chessboard and then produce as output the next move it would like to make. ■

Figure 2.1: The environment of a chess-playing agent (pixabay.com)

■ **Example 2.2 — Automated Car.** Our automated car from Chapter 1 receives a great deal of environmental input from its collection of sensors (cameras, LIDAR, GPS, etc.). Its output is a choice of what to do next (slow down, turn to the left, etc.) given that input data. ■

■ **Example 2.3 — Computer Diagnostic Physician.** Suppose we have an AI agent that diagnoses various types of cancer when given biopsy images. These images would form the agent's environmental input and its output would then be a decision about the classification of the image: malignant or benign tumor. ■

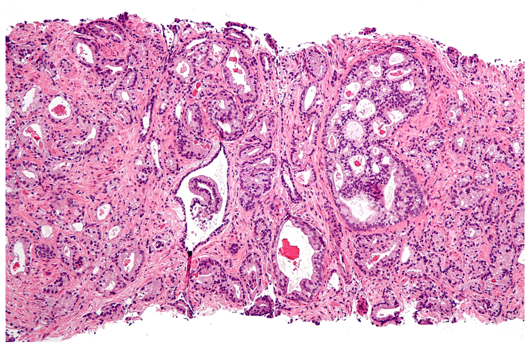

Figure 2.2: A needle biopsy of prostate adenocarcinoma, the most common type of prostate cancer. (wikipedia.org)

Note that we use the term *agent* to talk about both hardware and software AI systems. We can think of an autonomous bipedal robot making its way through a physical environment as an agent, but we should also think of software systems operating in virtual environments as agents.

2.2 Environments

We call the set of external factors around an agent the agent's **environment**. For some problems, the environment is very well-defined and the elements of the environment are clear. For example, for our chess-playing agent the environment is clearly the confines of a chess game with a chess board and the various chess pieces. In this case, the environment is simplified from the kind of environment that we humans face. A chess-playing agent only really needs to be concerned with the details of the chess board and the current chess game and the agent can completely ignore the glare of the late afternoon light off the chess board, what the current humidity is outside, and whether there is enough milk in the fridge.

On the other hand, a robotic agent operating in the real world has a much larger environment to worry about. If the agent is trying to navigate around a building and avoid all the doors, walls, desks, and people in it, then it cannot simply be concerned with the smaller problem of finding a clear path from start to finish. Its environment would force it to consider many unforeseen circumstances as people move about within the building, its own tires slip on different floor surfaces, and doors that should be open are locked.

In general, we can classify environments in several different ways according to their properties.

Environment Size and Complexity: First, environments come in different sizes and complexities. The Tic-Tac-Toe game environment is much smaller and simpler than the environment of an image labeling agent that may be given, say, any picture online and then be required to produce an English sentence about the contents of that image. The size and complexity of the environment influences how an agent might attempt to make decisions within that environment.

Environment Visibility: Second, environments like chess are entirely known to the agent. There are no surprises in chess as there is no hidden game information and there are no random events. For more complex real-world environments or detailed virtual simulations, there is often environmental information that is not known to the agent. In these cases the agent must reason about the uncertainties in the environment to continue to make good decisions even without perfect information. A poker-playing agent would not know which cards an opponent currently holds. A mars rover may have information about what sort of physical enviroment will be encountered on mars, but it would not have a complete map showing the details of the surface of mars down to each and every rock and bit of sand.

Environment Dynamics: Third, there may be other agents acting within the same environment and causing the state of the environment to change over time. Any multiple-player game agent would need to take into account how the other players in the game may affect the environment. The robot navigating an office building would need to avoid other mobile robots and walking humans to successfully reach its goal. It is important to keep in mind if the environment is only changed by the agent itself or whether outside influences such as other agents are dynamically changing the environment.

2.3 Environment States

How do we measure the size or complexity of an environment? We can do this using the concept of **environment states**.

Definition 2.3.1 — Environment State. A snapshot of an environment's configuration of variable values.

For example, for our chess-playing agent, the environment is the chess board and all the pieces within a game and an environmental state is the current set of remaining pieces in the game along with their exact positions and information about whose turn it is. *Note: if you're a chess aficionado, then the true state would be a bit more complicated since it would also contain information about whether or not each side had castled, whether kings and rooks had already moved thereby disallowing a future castle, and information about pawns using double moves to keep track of possible en passant captures.*

The size of an agent's environment can then be described by the **number of possible states** within that environment. This collection of all possible states is called the **state space**. For chess, the state space is quite large even though the game board looks to be a managable size. To get a very rough estimate you could consider placing between 2 and 32 of the pieces on any of the 64 board squares. Claude Shannon, the founder of information theory, made an early estimate of the number

Figure 2.3: The current configuration of the chess board with all the pieces and their locations is a single environmental state for the chess-playing agent.

of possible chess states as 10^{43}. This value is gigantic even for the fastest computers. For example, the fastest supercomputer cluster in early 2015 is Tianhe-2 owned by China's National University of Defense Technology and it is able to compute over 33 quadrillion floating point operations per second. Even at this speed, Tianhe-2 would take over 10^{18} years to compute all the possible chess states at the overly optimistic rate of one board position per operation.

Following are a couple more examples of environmental state spaces:

■ **Example 2.4 — State Space of a Robotic Arm.** A robotic arm may move in 3-D space by activating its servos to adjust angles on its various joints. A simple state of the robotic arm's environment is just its gripper's current (x, y, z) location in physical space. This definition of environmental state may be suitable for some problems, but we also may want to consider the state to be the list of exact angles of all the arm's joints along with the (x, y, z) locations of any nearby objects that the arm is meant to manipulate. ■

■ **Example 2.5 — State Space for an Image Recognition System.** An image recognition system may not be the same as the previous examples that interact with their environments to change the environment state, but we can still think of them as operating within their own kind of state space. If we have a system that can take an image as input and then classify that image as either containing a human face or not, then we could think of the state space as the space of all possible images that may be used as input. If the input images were grayscale and relatively low resolution at 320x240, and if each pixel were one byte, then we would have 320x240x256 possible input images to the system and each one would correspond with a different state. ■

2.3.1 Practical Environmental States

One of the biggest challenges of artificial intelligence is dealing with large environmental state spaces. We saw that even in the highly constrained chess environment there were way too many states to process in a practical way. Real-world problems are far less constrained and have so many states that it is infeasible to enumerate even tiny percentages of them. Basically, any non-trivial problem will have too many states to investigate each one individually. Much of the work in AI is centered around trying to reduce the size of the environmental state space down to a reasonable size by ignoring states that are not worth considering or by grouping similar states together so that they can be treated in a similar fashion.

2.4 Problems

Problem 2.1 Describe the environment for a Tic-Tac-Toe agent. How big (exactly) is the environmental state space?

Problem 2.2 Briefly describe states in an automated car's environment. Can you roughly calculate how large the overall state space might be given your state description?

Problem 2.3 How do you think humans deal with the giant number of possible states in the real world?