# Chapter 1. Introduction to Artificial Intelligence

In this chapter, we are going to discuss the concept of **Artificial Intelligence** (**AI**) and how it's applied in the real world. We spend a significant portion of our everyday life interacting with smart systems. It can be in the form of searching for something on the internet, Biometric face recognition, or converting spoken words to text. Artificial Intelligence is at the heart of all this and it's becoming an important part of our modern lifestyle. All these system are complex real-world applications and Artificial Intelligence solves these problems with mathematics and algorithms. During the course of this book, we will learn the fundamental principles that are used to build such applications and then implement them as well. Our overarching goal is to enable you to take up new and challenging Artificial Intelligence problems that you might encounter in your everyday life.

By the end of this chapter, you will know:

- What is AI and why do we need to study it?
- Applications of AI
- Branches of AI
- Turing test
- Rational agents
- General Problem Solvers
- Building an intelligent agent
- Installing Python 3 on various operating systems
- Installing the necessary Python packages

## What is Artificial Intelligence?

**Artificial Intelligence** (**AI**) is a way to make machines think and behave intelligently. These machines are controlled by software inside them, so AI has a lot to do with intelligent software programs that control these machines. It is a science of finding theories and methodologies that can help machines understand the world and accordingly react to situations in the same way that humans do.

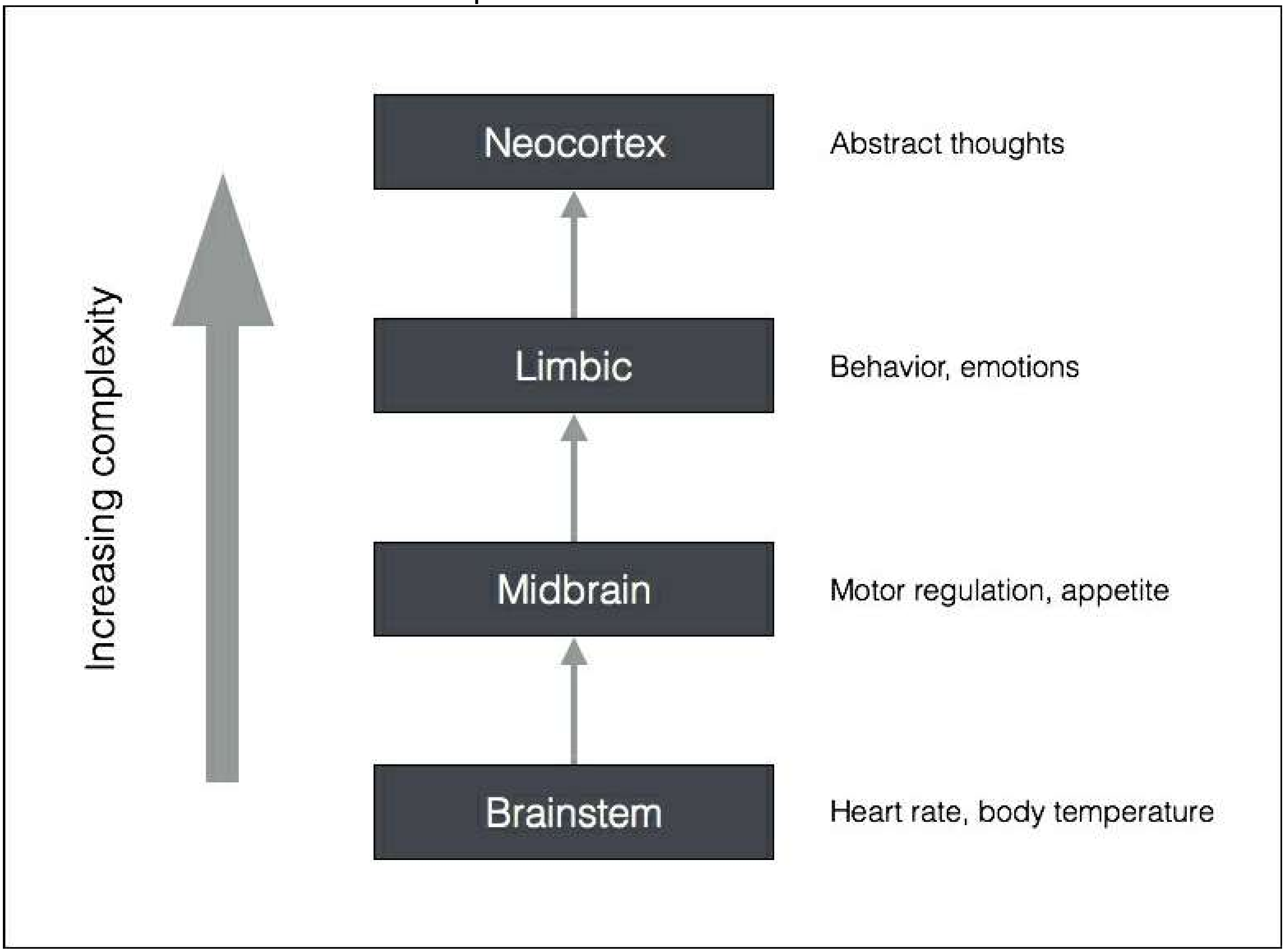
If we look closely at how the field of AI has emerged over the last couple of decades, you will see that different researchers tend to focus on different concepts to define AI. In the modern world, AI is used across many verticals in many different forms. We want the machines to sense, reason, think, and act. We want our machines to be rational too.

AI is closely related to the study of human brain. Researchers believe that AI can be accomplished by understanding how the human brain works. By mimicking the way the human brain learns, thinks, and takes action, we can build a machine that can do the same. This can be used as a platform to develop intelligent systems that are capable of learning.

# Why do we need to study AI?

AI has the ability to impact every aspect of our lives. The field of AI tries to understand patterns and behaviors of entities. With AI, we want to build smart systems and understand the concept of intelligence as well. The intelligent systems that we construct are very useful in understanding how an intelligent system like our brain goes about constructing another intelligent system.

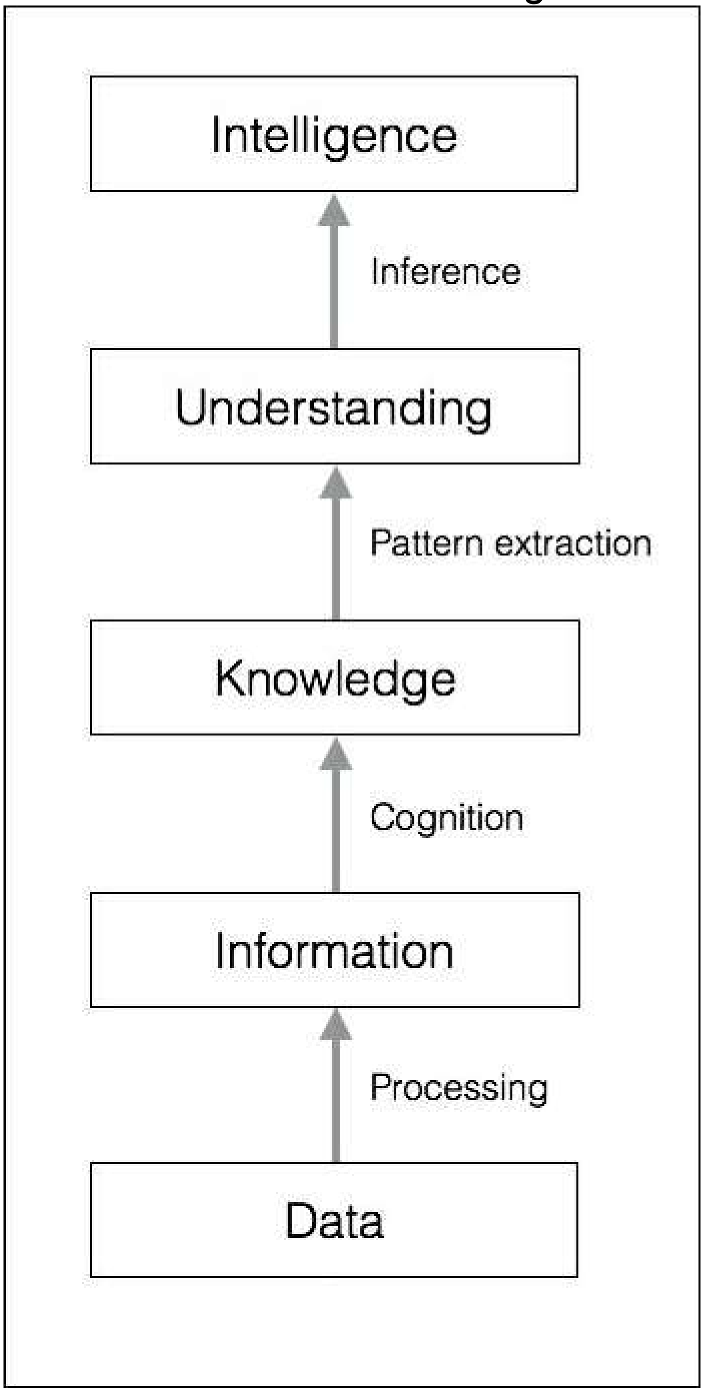
Let's take a look at how our brain processes information:




Compared to some other fields such as Mathematics or Physics that have been around for centuries, AI is relatively in its infancy. Over the last couple of decades, AI has produced some spectacular products such as self-driving cars and intelligent robots that can walk. Based on the direction in which we are heading, it's pretty obvious that achieving intelligence will have a great impact on our lives in the coming years.

We can't help but wonder how the human brain manages to do so much with such effortless ease. We can recognize objects, understand languages, learn new things, and perform many more sophisticated tasks with our brain. How does the human brain do this? When you try to do this with a machine, you will see that it falls way behind! For example, when we try to look for things such as extraterrestrial life or time travel, we don't know if those things exist. The good thing about the holy grail of AI is that we know it exists. Our brain is

the holy grail! It is a spectacular example of an intelligent system. All we have to do is to mimic its functionality to create an intelligent system that can do something similar, possibly even more.
Let's see how raw data gets converted to wisdom through various levels of processing:




One of the main reasons we want to study AI is to automate many things. We live in a world where:

- We deal with huge and insurmountable amounts of data. The human brain can't keep track of so much data.
- Data originates from multiple sources simultaneously.
- The data is unorganized and chaotic.
- Knowledge derived from this data has to be updated constantly because the data itself keeps changing.
- The sensing and actuation has to happen in real time with high precision.

Even though the human brain is great at analyzing things around us, it cannot keep up with the preceding conditions. Hence, we need to design and develop intelligent machines that can do this. We need AI systems that can:

- Handle large amounts of data in an efficient way. With the advent of Cloud Computing, we are now able to store huge amounts of data.
- Ingest data simultaneously from multiple sources without any lag.
- Index and organize data in a way that allows us to derive insights.
- Learn from new data and update constantly using the right learning algorithms.
- Think and respond to situations based on the conditions in real time.

AI techniques are actively being used to make existing machines smarter, so that they can execute faster and more efficiently.

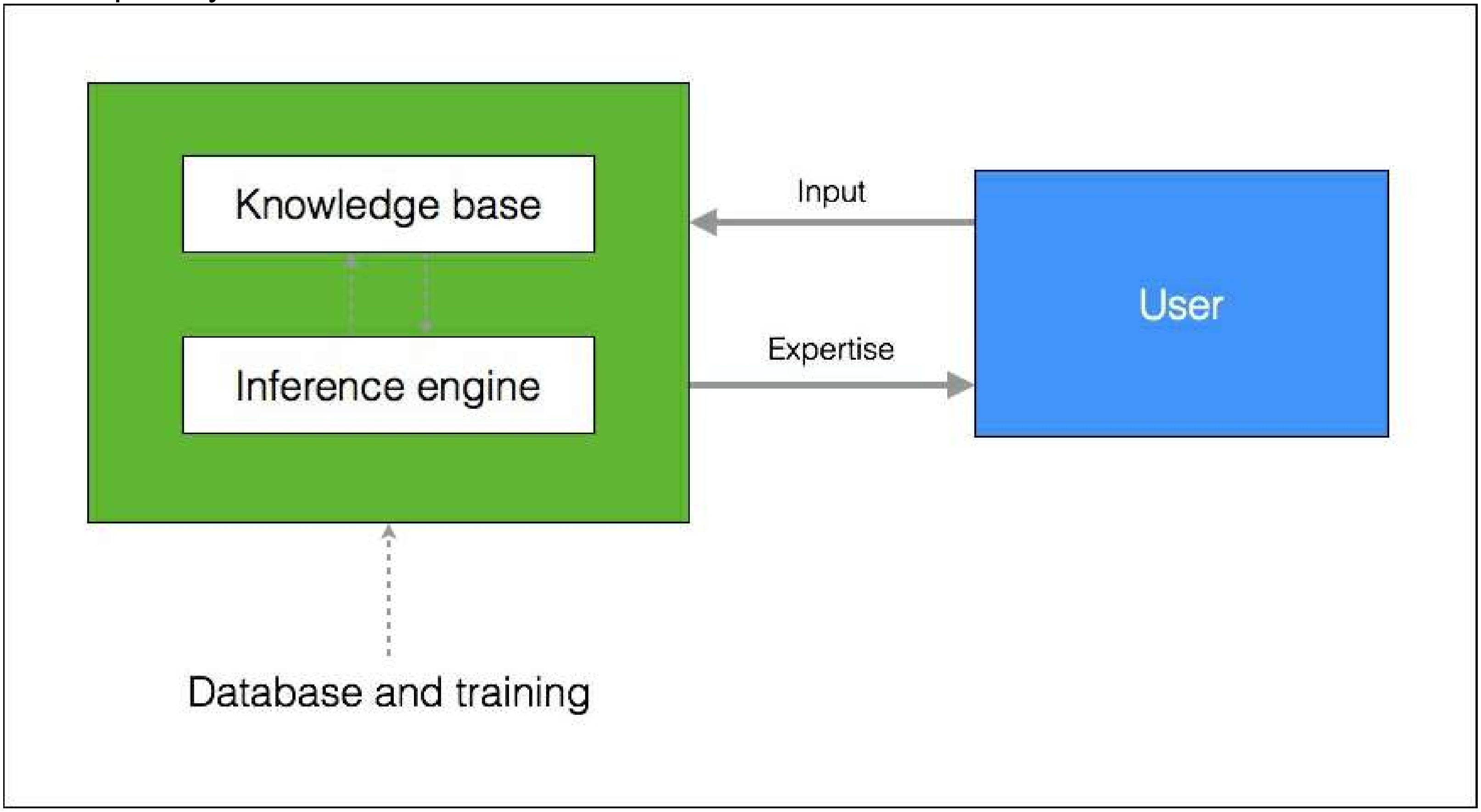
# Applications of AI

Now that we know how information gets processed, let's see where AI appears in the real world. AI manifests itself in various different forms across multiple fields, so it's important to understand how it's useful in various domains. AI has been used across many industries and it continues to expand rapidly. Some of the most popular areas include:

- **Computer Vision**: These are the systems that deal with visual data such as images and videos. These systems understand the content and extract insights based on the use case. For example, Google uses reverse image search to search for visually similar images across the Web.

- **Natural Language Processing**: This field deals with understanding text. We can interact with a machine by typing natural language sentences. Search engines use this extensively to deliver the right search results.

- **Speech Recognition**: These systems are capable of hearing and understanding spoken words. For example, there are intelligent personal assistants on our smartphones that can understand what we are saying and give relevant information or perform an action based on that.
- **Expert Systems**: These systems use AI techniques to provide advice or make decisions. They usually use databases of expert knowledge areas such as finance, medicine, marketing, and so on to give advice about what to do next. Let's see what an expert system looks like and how it interacts with the user:

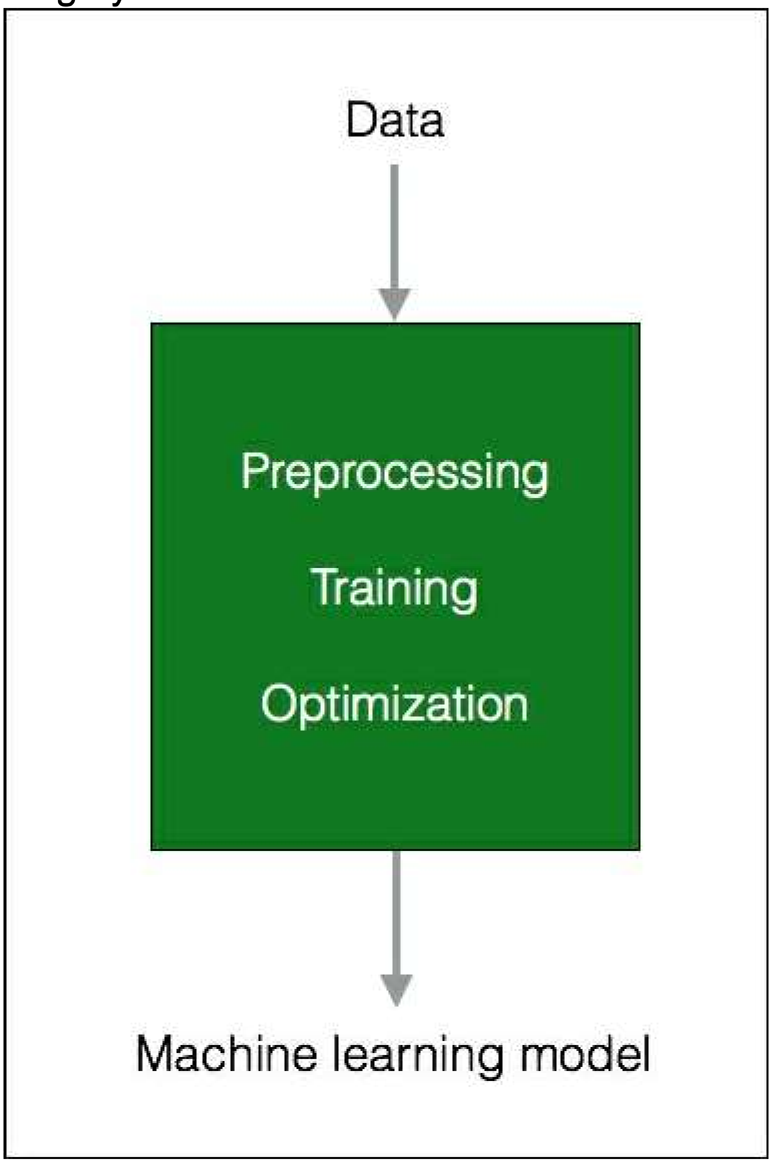



- **Games**: AI is used extensively in the gaming industry. It is used to design intelligent agents that can compete with humans. For example, **AlphaGo** is a computer program that can play the strategy game Go. It is also used in designing many other types of games where we expect the computer to behave intelligently.
- **Robotics**: Robotic systems actually combine many concepts in AI. These systems are able to perform many different tasks. Depending on the situation, robots have sensors and actuators that can do different things. These sensors can see things in front of them and measure the temperature, heat, movements, and so on. They have processors on board that compute various things in real time. They are also capable of adapting to the new environments.

# Branches of AI

It is important to understand the various fields of study within AI so that we can choose the right framework to solve a given real-world problem. Here's a list of topics that are dominant:

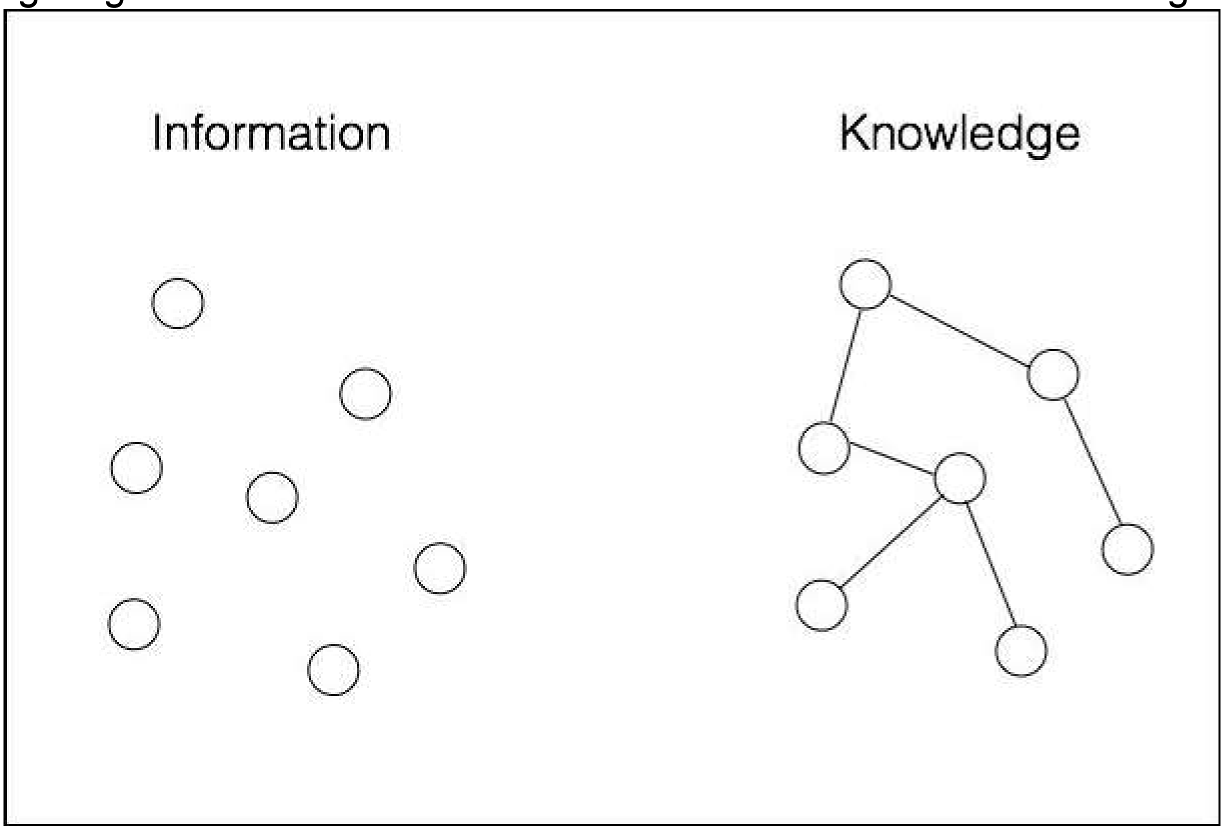
- **Machine learning and pattern recognition**: This is perhaps the most popular form of AI out there. We design and develop software that can learn from data. Based on these learning models, we perform predictions on unknown data. One of the main constraints here is that these programs are limited to the power of the data. If the dataset is small, then the learning models would be limited as well. Let's see what a typical machine learning system looks like:




When a system makes an observation, it is trained to compare it with what it has already seen in the form of a pattern. For example, in a face recognition system, the software will try to match the pattern of eyes, nose, lips, eyebrows, and so on in order to find a face in the existing database of users.

- **Logic-based AI**: Mathematical logic is used to execute computer programs in logic-based AI. A program written in logic-based AI is basically a set of statements in logical form that express facts and rules about a particular problem domain. This is used extensively in pattern matching, language parsing, semantic analysis, and so on.

- **Search**: The Search techniques are used extensively in AI programs. These programs examine a large number of possibilities and then pick the most optimal path. For example, this is used a lot in strategy games such as Chess, networking, resource allocation, scheduling, and so on.
- **Knowledge representation**: The facts about the world around us need to be represented in some way for a system to make sense of them. The languages of mathematical logic are frequently used here. If knowledge is represented efficiently, systems can be smarter and more intelligent. Ontology is a closely related field of study that deals with the kinds of objects that exist. It is a formal definition of the properties and relationships of the entities that exist in a particular domain. This is usually done with a particular taxonomy or a hierarchical structure of some kind. The following diagram shows the difference between information and knowledge:




- **Planning**: This field deals with optimal planning that gives us maximum returns with minimal costs. These software programs start with facts about the particular situation and a statement of a goal. These programs are also aware of the facts of the world, so that they know what the rules are. From this information, they generate the most optimal plan to achieve the goal.
- **Heuristics**: A heuristic is a technique used to solve a given problem that's practical and useful in solving the problem in the short term, but not guaranteed to be optimal. This is more like an educated guess on what approach we should take to solve a problem. In AI, we frequently encounter situations where we cannot check every single possibility to pick the best option. So we need to use heuristics to achieve the goal. They are used extensively in AI in fields such as robotics, search engines, and so on.
- **Genetic programming**: Genetic programming is a way to get programs to solve a task, by mating programs and selecting the fittest. The programs are encoded as a set of genes, using an algorithm to get a program that is able to perform the given task

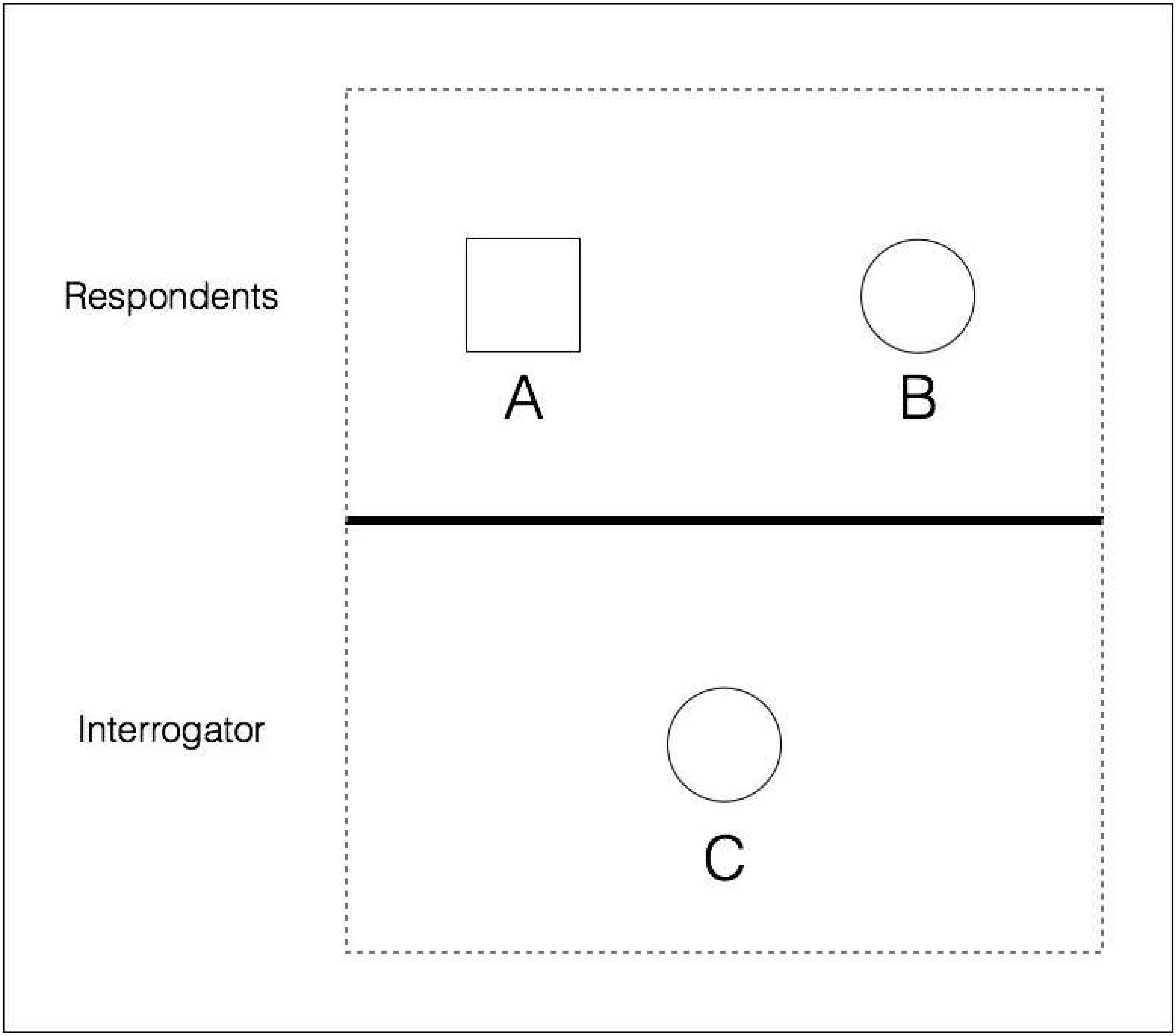
really well.

# Defining intelligence using Turing Test

The legendary computer scientist and mathematician, *Alan Turing*, proposed the Turing Test to provide a definition of intelligence. It is a test to see if a computer can learn to mimic human behavior. He defined intelligent behavior as the ability to achieve human-level intelligence during a conversation. This performance should be sufficient to trick an interrogator into thinking that the answers are coming from a human.

To see if a machine can do this, he proposed a test setup: he proposed that a human should interrogate the machine through a text interface. Another constraint is that the human cannot know who's on the other side of the interrogation, which means it can either be a machine or a human. To enable this setup, a human will be interacting with two entities through a text interface. These two entities are called respondents. One of them will be a human and the other one will be the machine.

The respondent machine passes the test if the interrogator is unable to tell whether the answers are coming from a machine or a human. The following diagram shows the setup of a Turing Test:




As you can imagine, this is quite a difficult task for the respondent machine. There are a lot of things going on during a conversation. At the very minimum, the machine needs to be well versed with the following things:

- **Natural Language Processing**: The machine needs this to communicate with the interrogator. The machine needs to parse the sentence, extract the context, and give an appropriate answer.
- **Knowledge Representation**: The machine needs to store the information provided before the interrogation. It also needs to keep track of the information being provided during the conversation so that it can respond appropriately if it comes up again.
- **Reasoning**: It's important for the machine to understand how to interpret the information that gets stored. Humans tend to do this automatically to draw conclusions in real time.
- **Machine Learning**: This is needed so that the machine can adapt to new conditions in real time. The machine needs to analyze and detect patterns so that it can draw

inferences.

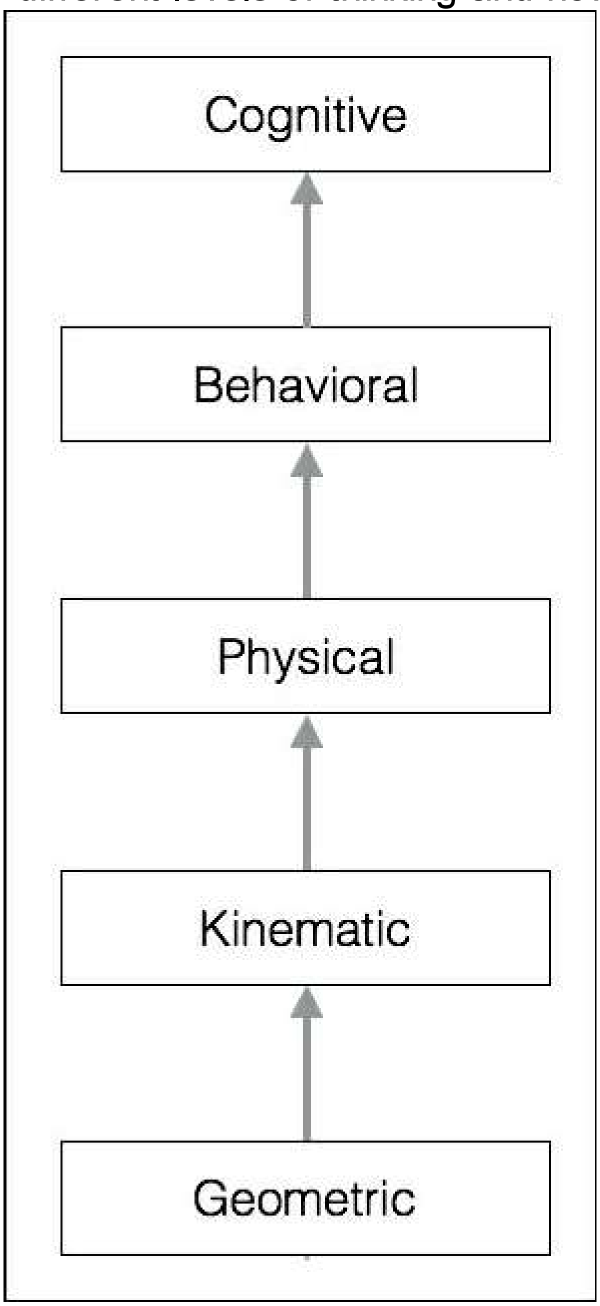
You must be wondering why the human is communicating with a text interface. According to Turing, physical simulation of a person is unnecessary for intelligence. That's the reason the Turing Test avoids direct physical interaction between the human and the machine. There is another thing called the Total Turing Test that deals with vision and movement. To pass this test, the machine needs to see objects using computer vision and move around using Robotics.

# Making machines think like humans

For decades, we have been trying to get the machine to think like a human. In order to make this happen, we need to understand how humans think in the first place. How do we understand the nature of human thinking? One way to do this would be to note down how we respond to things. But this quickly becomes intractable, because there are too many things to note down. Another way to do this is to conduct an experiment based on a predefined format. We develop a certain number of questions to encompass a wide variety of human topics, and then see how people respond to it.

Once we gather enough data, we can create a model to simulate the human process. This model can be used to create software that can think like humans. Of course this is easier said than done! All we care about is the output of the program given a particular input. If the program behaves in a way that matches human behavior, then we can say that humans have a similar thinking mechanism.

The following diagram shows different levels of thinking and how our brain prioritizes things:

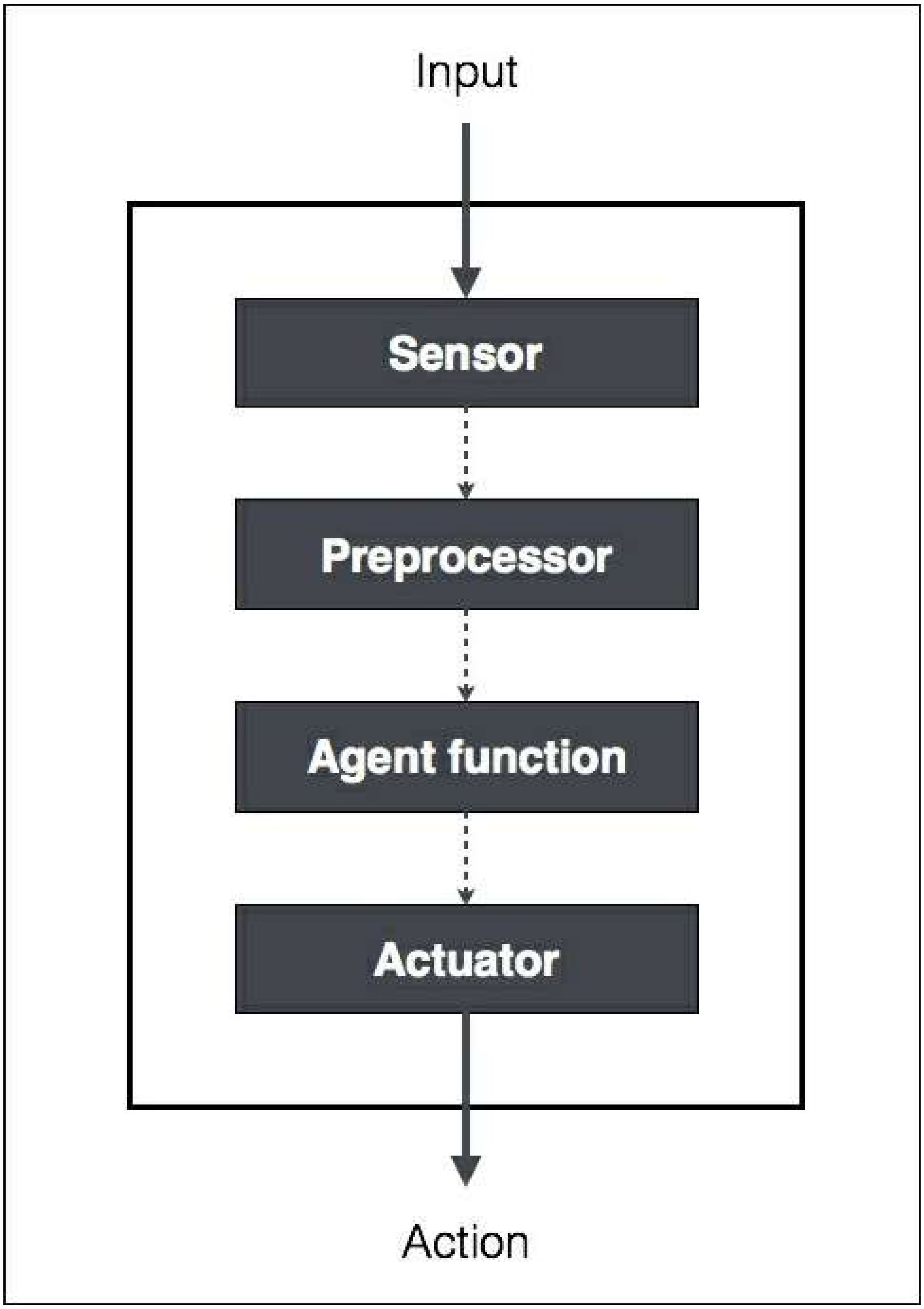



Within computer science, there is a field of study called **Cognitive Modeling** that deals with simulating the human thinking process. It tries to understand how humans solve problems. It takes the mental processes that go into this problem solving process and turns it into a software model. This model can then be used to simulate human behavior. Cognitive modeling is used in a variety of AI applications such as deep learning, expert systems, Natural Language Processing, robotics, and so on.

# Building rational agents

A lot of research in AI is focused on building rational agents. What exactly is a rational agent? Before that, let us define the word rationality. Rationality refers to doing the right thing in a given circumstance. This needs to be performed in such a way that there is maximum benefit to the entity performing the action. An agent is said to act rationally if, given a set of rules, it takes actions to achieve its goals. It just perceives and acts according to the information that's available. This system is used a lot in AI to design robots when they are sent to navigate unknown terrains.

How do we define the *right* thing? The answer is that it depends on the objectives of the agent. The agent is supposed to be intelligent and independent. We want to impart the ability to adapt to new situations. It should understand its environment and then act accordingly to achieve an outcome that is in its best interests. The best interests are dictated by the overall goal it wants to achieve. Let's see how an input gets converted to action:




How do we define the performance measure for a rational agent? One might say that it is directly proportional to the degree of success. The agent is set up to achieve a particular task, so the performance measure depends on what percentage of that task is complete. But we must think as to what constitutes rationality in its entirety. If it's just about results, can the agent take any action to get there?

Making the right inferences is definitely a part of being rational, because the agent has to act rationally to achieve its goals. This will help it draw conclusions that can be used successively. What about situations where there are no provably right things to do? There are situations where the agent doesn't know what to do, but it still has to do something. In this situation, we cannot include the concept of inference to define rational behavior.

# General Problem Solver

The **General Problem Solver** (**GPS**) was an AI program proposed by *Herbert Simon*, *J.C. Shaw*, and *Allen Newell*. It was the first useful computer program that came into existence in the AI world. The goal was to make it work as a universal problem-solving machine. Of course there were many software programs that existed before, but these programs performed specific tasks. GPS was the first program that was intended to solve any general problem. GPS was supposed to solve all the problems using the same base algorithm for every problem.

As you must have realized, this is quite an uphill battle! To program the GPS, the authors created a new language called **Information Processing Language** (**IPL**). The basic premise is to express any problem with a set of well-formed formulas. These formulas would be a part of a directed graph with multiple sources and sinks. In a graph, the source refers to the starting node and the sink refers to the ending node. In the case of GPS, the source refers to axioms and the sink refers to the conclusions.

Even though GPS was intended to be a general purpose, it could only solve well-defined problems, such as proving mathematical theorems in geometry and logic. It could also solve word puzzles and play chess. The reason was that these problems could be formalized to a reasonable extent. But in the real world, this quickly becomes intractable because of the number of possible paths you can take. If it tries to brute force a problem by counting the number of walks in a graph, it becomes computationally infeasible.

## Solving a problem with GPS

Let's see how to structure a given problem to solve it using GPS:

1. The first step is to define the goals. Let's say our goal is to get some milk from the grocery store.
2. The next step is to define the preconditions. These preconditions are in reference to the goals. To get milk from the grocery store, we need to have a mode of transportation and the grocery store should have milk available.
3. After this, we need to define the operators. If my mode of transportation is a car and if the car is low on fuel, then we need to ensure that we can pay the fueling station. We need to ensure that you can pay for the milk at the store.

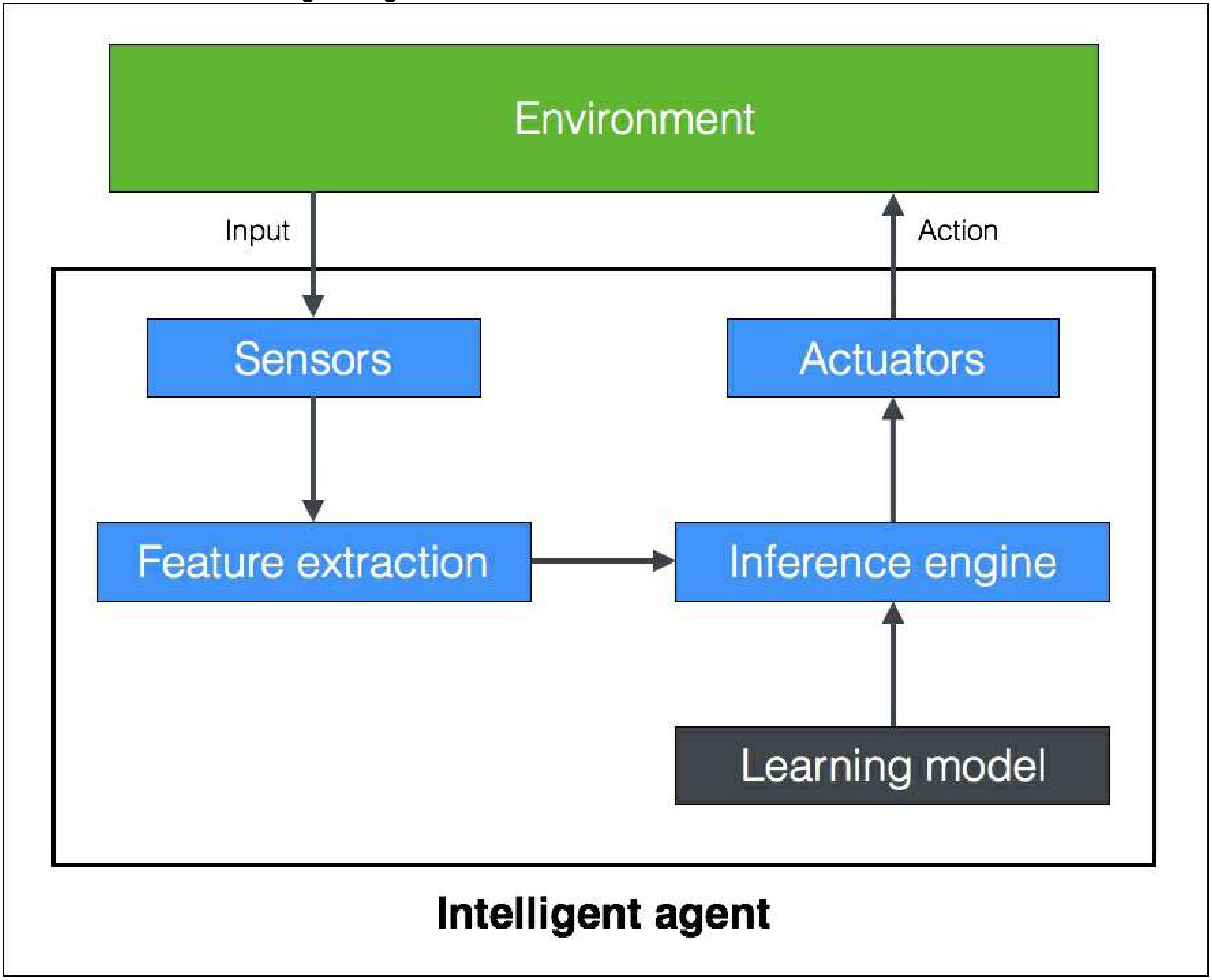
An operator takes care of the conditions and everything that affects them. It consists of actions, preconditions, and the changes resulting from taking actions. In this case, the action is giving money to the grocery store. Of course, this is contingent upon you having the money in the first place, which is the precondition. By giving them the money, you are changing your money condition, which will result in you getting the milk.

GPS will work as long as you can frame the problem like we did just now. The constraint is that it uses the search process to perform its job, which is way too computationally complex and time consuming for any meaningful real-world application.

# Building an intelligent agent

There are many ways to impart intelligence to an agent. The most commonly used techniques include machine learning, stored knowledge, rules, and so on. In this section, we will focus on machine learning. In this method, the way we impart intelligence to an agent is through data and training.

Let's see how an intelligent agent interacts with the environment:




With machine learning, we want to program our machines to use labeled data to solve a given problem. By going through the data and the associated labels, the machine learns how to extract patterns and relationships.

In the preceding example, the intelligent agent depends on the learning model to run the inference engine. Once the sensor perceives the input, it sends it to the feature extraction block. Once the relevant features are extracted, the trained inference engine performs a prediction based on the learning model. This learning model is built using machine learning. The inference engine then takes a decision and sends it to the actuator, which then takes the required action in the real world.

There are many applications of machine learning that exist today. It is used in image recognition, robotics, speech recognition, predicting stock market behavior, and so on. In order to understand machine learning and build a complete solution, you will have to be familiar with many techniques from different fields such as pattern recognition, artificial neural networks, data mining, statistics, and so on.

# Types of models

There are two types of models in the AI world: Analytical models and Learned models. Before we had machines that could compute, people used to rely on analytical models. These models were derived using a mathematical formulation, which is basically a sequence of steps followed to arrive at a final equation. The problem with this approach is that it was based on human judgment. Hence these models were simplistic and inaccurate with just a few parameters.

We then entered the world of computers. These computers were good at analyzing data. So, people increasingly started using learned models. These models are obtained through the process of training. During training, the machines look at many examples of inputs and outputs to arrive at the equation. These learned models are usually complex and accurate, with thousands of parameters. This gives rise to a very complex mathematical equation that governs the data.

Machine Learning allows us to obtain these learned models that can be used in an inference engine. One of the best things about this is the fact that we don't need to derive the underlying mathematical formula. You don't need to know complex mathematics, because the machine derives the formula based on data. All we need to do is create the list of inputs and the corresponding outputs. The learned model that we get is just the relationship between labeled inputs and the desired outputs.