Agents

1 Rational agents

An agent is an entity that perceives and acts, **everything that recognizes the environment through sensors and acts upon the environment through actuators**.

A rational agent (or intelligent agent), on the other hand, is a person or entity that always aims to perform optimal actions based on given premises and information. A rational agent can perform specific, predictable and repetitive tasks for applications with some level of individualism.

How do an intelligent agent work? Its main components are:

- **Sensors**: a device that detects environmental changes and sends the information to other devices;

- **Actuators**: machine component that convert energy into motion;

- **Effectors**: device that applies the motion to perform a specific task.

What is the difference between actuators and effectors? An actuator is a device that converts energy into motion, while an effector is a device that applies that motion to perform a specific task. For example, a motor can be considered an actuator, while a robotic arm controlled by the motor would be considered an effector.

Rationality defines the level of being reasonable.

2 Agent and environment

An AI system can be defined as the study of a rational agent and its environment. A rational agent can be anything that makes decisions, typically a person, robot or software. An agent can be seen as a black box interacting with the environment in two ways:

- takes input from the environment (perceives)

- produces an output (acts)

The agents sense the environment through sensors and act on their environment through actuators.

How can we implement this box? How do the system will convert perceptions into actions?

3 Agent architecture

Abstractly, an agent can be seen as a function that maps from percept histories to actions:

$$f : P^* \to A \tag{1}$$

A *percept* is a sequence of *perceptions*: maybe previous perceptions could help us in the future.

For a given task we seek the agent (or class of agents) with the best performance: it aims to choose whichever action maximizes the expected value of the performance measure given the percept sequence to date. Computational limitations make perfect rationality unachievable.

4 Problem solving

Firstly we have the **problem formulation**: we define a set of states (called **state space**) and one among them as **initial state** (starting point). Then we need to specify a **goal state** (or a set of goal states), that is the condition we want the agent to achieve. The agent will use artificial intelligence techniques such as search algorithms, decision making algorithms and machine learning to make decisions and take action to reach the goal state from the initial state; this step is called **search process**.

The agent can implement the search process **online**, acting without complete knowledge (e.g. in real-time scenarios), or **offline**.

Online problem solving means that searching takes place as data becomes available, while in offline problem solving data is accumulated over a period of time and the model can search using all that information.

5 Agent types

Five basic types in order of increasing generality:

- simple reflex agents
- model-based reflex agents
- goal-based agents
- utility-based agents
- learning agents

6   Simple reflex agents
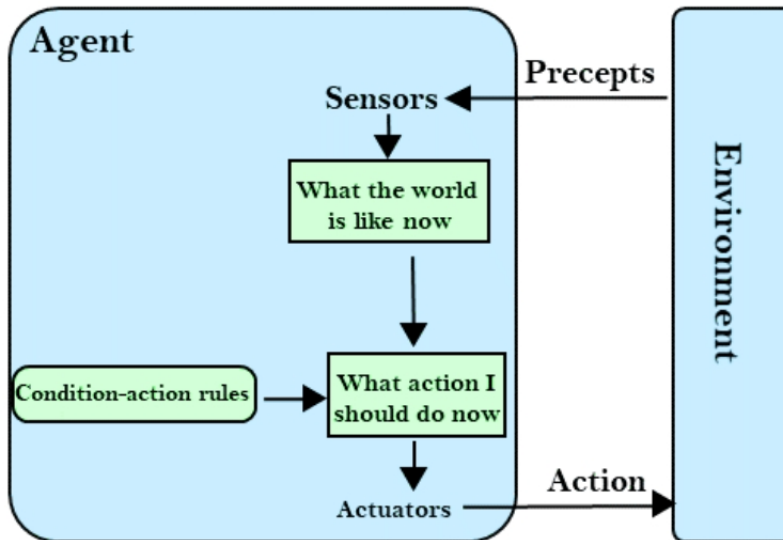
TLDR: A simple reflex agent operates based on a condition-action rule. It selects actions solely based on the current percept (input) without considering the agent's history or future consequences.

The current percept is used rather than the percept history to act by these agents. The core of these agents is the **condition-action rule**. The condition-action rule is a rule that maps a condition to an action: they perceive something, they do something else in response to that.

Simple reflex agents are a direct connection between perception and action; they can't make prediction since the model does not know how the world will change in response to the action. They have very limited intelligence and require a fully observable environment. Environmental changes are not adaptable, meaning that if we want to act differently to a condition we need to change them (cause of the condition-action rule).

Example: Suppose the agent is a taxi that goes from point A to point B. This agent will have a set of condition-action rules like this:

- **if car at crossroad 1: go left**

- **if car at crossroad 2: go right**

- **if traffic light is green: go forward**

- **...**

7 Model-based reflex agents
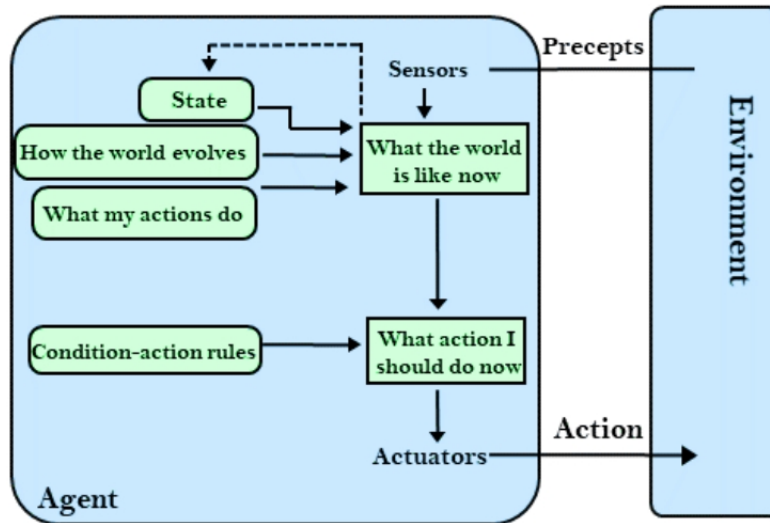
TLDR: A model-based agent maintains an internal model of the world, allowing it to keep track of the current state and predict the consequences of its actions.

Model-based agents still have that sort of condition-action rule set and they don't have a goal. They have a model of the world; each time they take an action they know how that will effect the world; the action is still taken on the basis of the condition-action rule set.

Basically we add a model of the world to the agent to predict how the world will change based on its actions.

Simple-reflex agents	Model-based agents
• look at the environment and act accordingly	• look at the environment;
	• ask themselves which effects will each action have;
	• update the model;
	• consult the same condition-action rule set;
	• take the action;

In the case of a model-based agent, the taxi agent maintains an internal model of the world, which includes relevant information about the environment, such as roads, traffic conditions, and any other pertinent details. This internal model allows the agent to have a representation of the current state of the world and predict the consequences of its actions. If we want the agent to go from A to B, it can identify multiple ways to do it but it does not know which is the best way: it knows that, if in point (a) it nees to turn right, if in point (b) it needs to go left and so on, but everything needs to be explicitly coded into the same condition-action rule set.

8 Goal-based agents
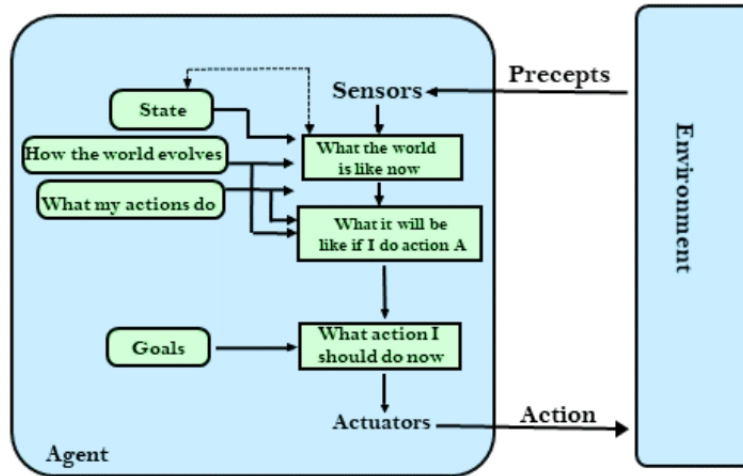
TLDR: A goal-based agent selects actions based on its current state and a predefined goal. It uses its internal knowledge to identify the sequence of actions required to achieve the goal.

Goal based agents use **goal information**. The main advantages are:

- the knowledge supporting a decision is explicitly modeled and thereby modifications are allowed;

- the agent, knowing how the world evolve, can also project himself towards the future and say if a sequence of actions can eventually let him reach the goal;

These agents are more flexible because the knowledge that supports its decisions is represented explicitly and can therefore be easily modified.

Referring to the previous example, a goal-based taxi agent easily adapt to reach a different point just by changing the goal - $\cancel{B} \to C$ - (changing the destination of the taxi). The simple reflex/model-based reflex agent's rule set is difficult to adapt in comparison.

9 Utility-based agents
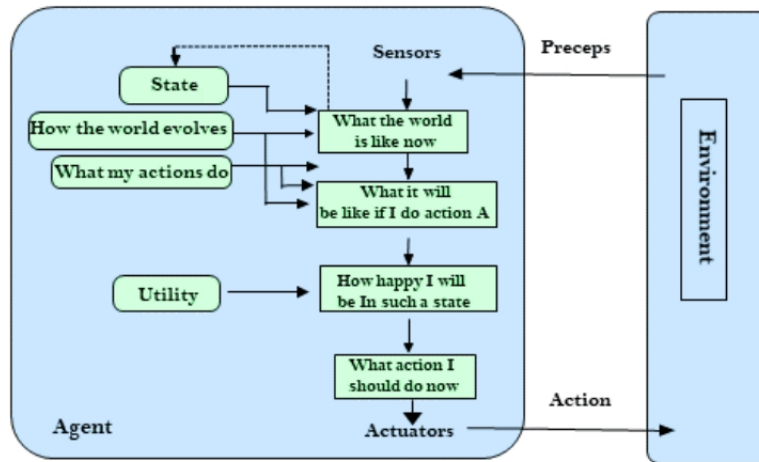
TLDR: A utility-based agent makes decisions by assigning a utility value to each possible action. It considers both the immediate and long-term consequences of its actions to maximize overall utility.

Choices made by an utility based agent are based on utility. The utility measures the goodness of the outcome of an action. The agent can therefore take **the best** action among the available options.

Using the aforementioned taxi examples, if the taxi is a goal-based agent, it can take one of many different paths to reach the goal and all of them are the same from its perspective. If the taxi is a utility-based agent, it will choose the best path according to a criterion (e.g. traffic conditions or shortness). **The goal alone is not enough to generate high quality behaviour in most environments**; many action sequences will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable or cheaper than others.

In the taxi domain, an utility-based taxi agent assigns a utility value to each potential path to the goal based on factors such as distance, traffic conditions etc. It can choose the path that maximizes this utility function.

Goal just provides a crude binary distinction between "happy" or "unhappy" states. A performance measure assigns a score to any given sequence of environment states.

Goals alone are not enough to generate high-quality behaviour in most environments.

For example, many action sequences will get the taxi to its destination (thereby achieving the goal), but some are quicker, safer, more reliable or cheaper than others.

We have already seen that a performance measure assigns a score to any given sequence of environment states, so it can easily distinguish between more and less desirable ways of leading the agent to its goal. An agent's utility function is essentially an internalization of the performance measure. Provided that the internal utility function and the external performance measures are in agreement, an agent that chooses actions to maximize its utility will be rational according to the external performance measure.

- When there are conflicting goals, only some of which can be achieved (for example, speed and safety), the utility function specifies the appropriate trade-off.

- When there are several goals that the agent can aim for, none of which can be achieved with certainty, utility provides a way in which the likelihood of success can be weighed against the importance of the goals.

An agent that possesses an explicit utility function can make rational decisions with a general-purpose algorithm that does not depend on the specific utility function being maximized.

10   Learning agents
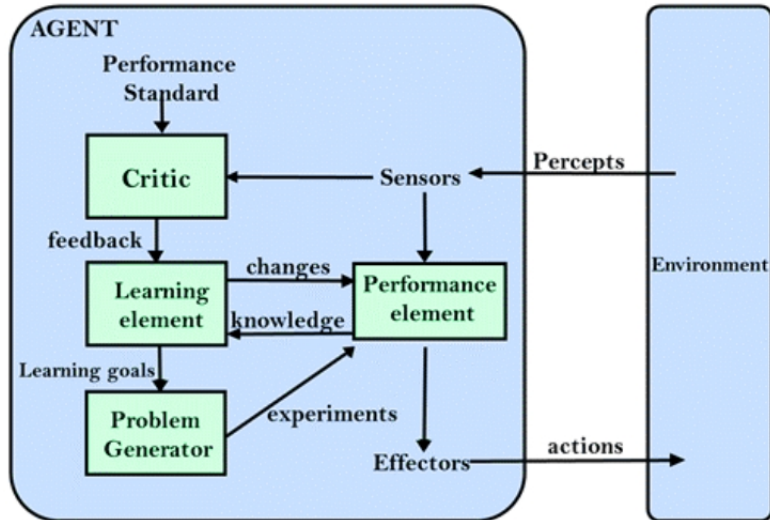
TLDR: A learning agent improves its performance over time by learning from its interactions with the environment. It uses various techniques like supervised learning, reinforcement learning, or unsupervised learning to update its behavior.

All these can be turned into learning agents. A learning agent is an agent with the capability of learning from its previous experience. What is it needed to talk about learning agents?

- *Learning element*: element that enables learning from previous experience;

- *Critic*: provides feedback on how well the agent is doing concerning a fixed performance standard;

- *Performance element*: the actions to be performed are selected;

- *Problem generator*: acts as a feedback agent that performs certain tasks such as making suggestions that will lead to new and informative experiences;