Agent, Search and Game Playing

Course Outline

1. Black- Box Model of Agent
2. Intentionality and Goals
3. Games, Search, Heuristic and Pruning
4. Strategies Rules
5. Making Simple Game- Playing Agent for TTT
6. Evaluation Functions, Utilitarian, Decision Making, Planning, Internal Representation

Agent

- In computer science, an agent is a **software or hardware system** that is designed to act **autonomously or semi-autonomously** in an environment, with the ability to perceive that environment and act upon it to achieve specific goals or objectives.
- Agents can be classified based on various characteristics such as their ***level of autonomy, their degree of intelligence, and their interaction with other agents or users.***
- Some examples of agents include ***chatbots, recommendation engines, intelligent personal assistants, autonomous vehicles, drones, and robots.***

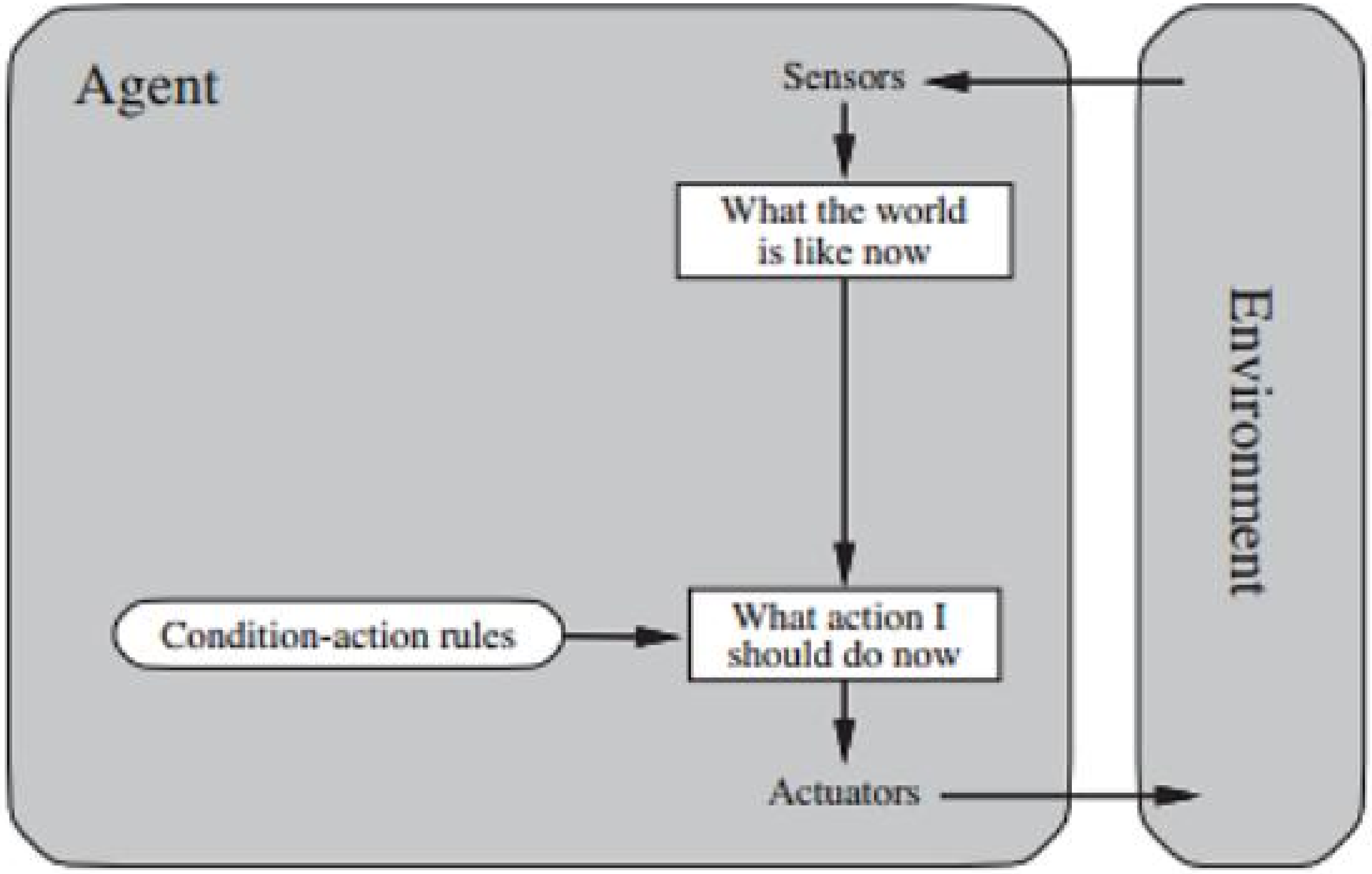
Agent

- The behavior of an agent is driven by a set of rules or algorithms that allow it to ***perceive the environment, reason about it, and take appropriate actions*** to achieve its goals.
- These ***rules and algorithms can be based on various techniques***, such as machine learning, decision trees, or expert systems.
- The performance of an agent is typically evaluated against a specific ***performance measure or metric***, which defines the objectives or goals that the agent is trying to achieve.
- For example, the performance of a chatbot can be evaluated based on its ability to answer user queries accurately and efficiently.

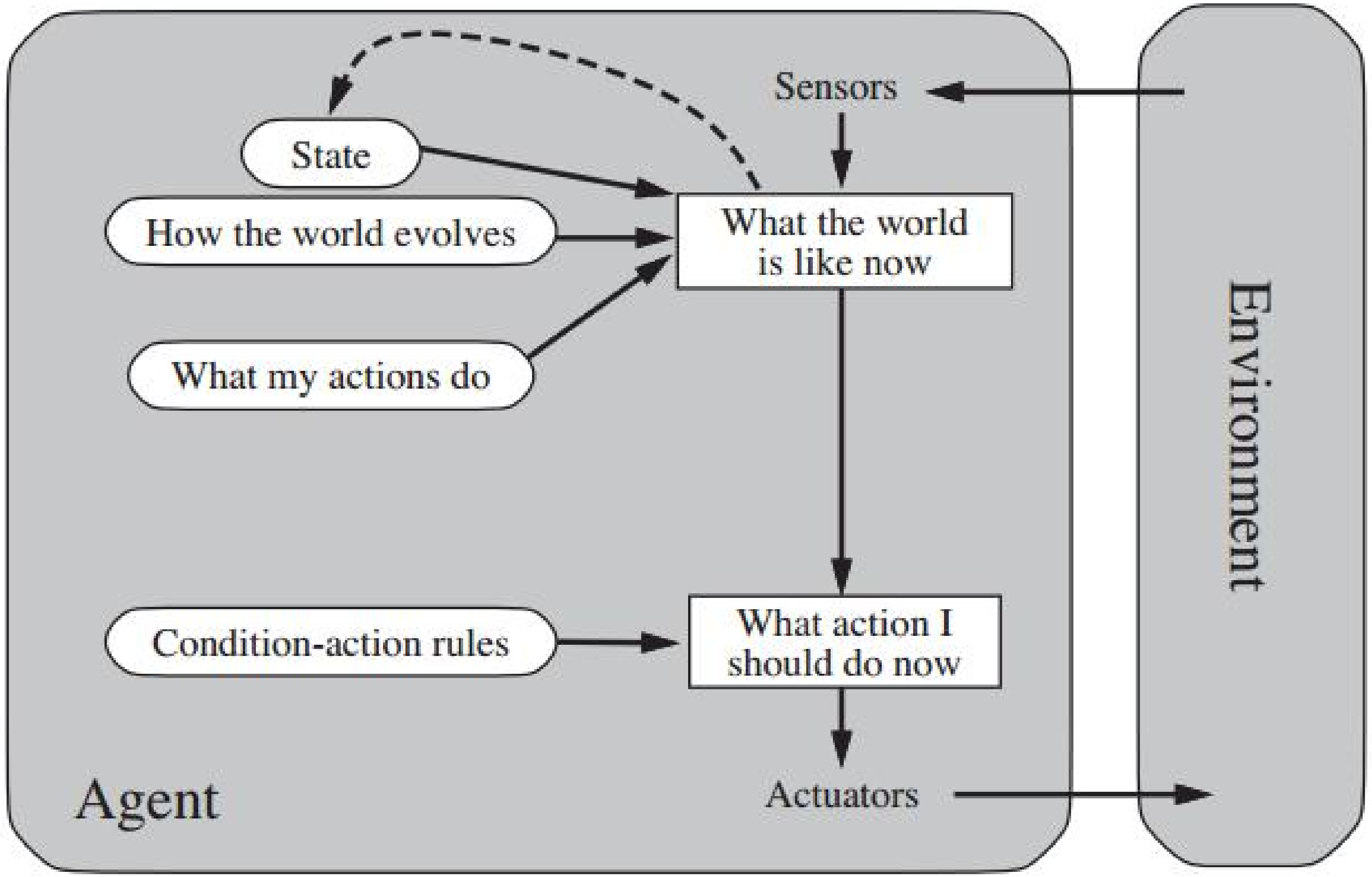
Types of Agent

- ***Simple reflex agents:***
  - These agents select actions based solely on the ***current percept***, without considering past percepts or future consequences.
  - For example, a thermostat that turns on the air conditioning when the temperature rises above a certain threshold is a simple reflex agent.
- ***Model-based reflex agents:***
  - These agents maintain an internal model of the world and use it to make decisions based on ***past percepts and predicted future outcomes***.
  - For example, a robot vacuum cleaner that maps out a room and avoids obstacles while cleaning is a model-based reflex agent.

Simple Reflex Agent



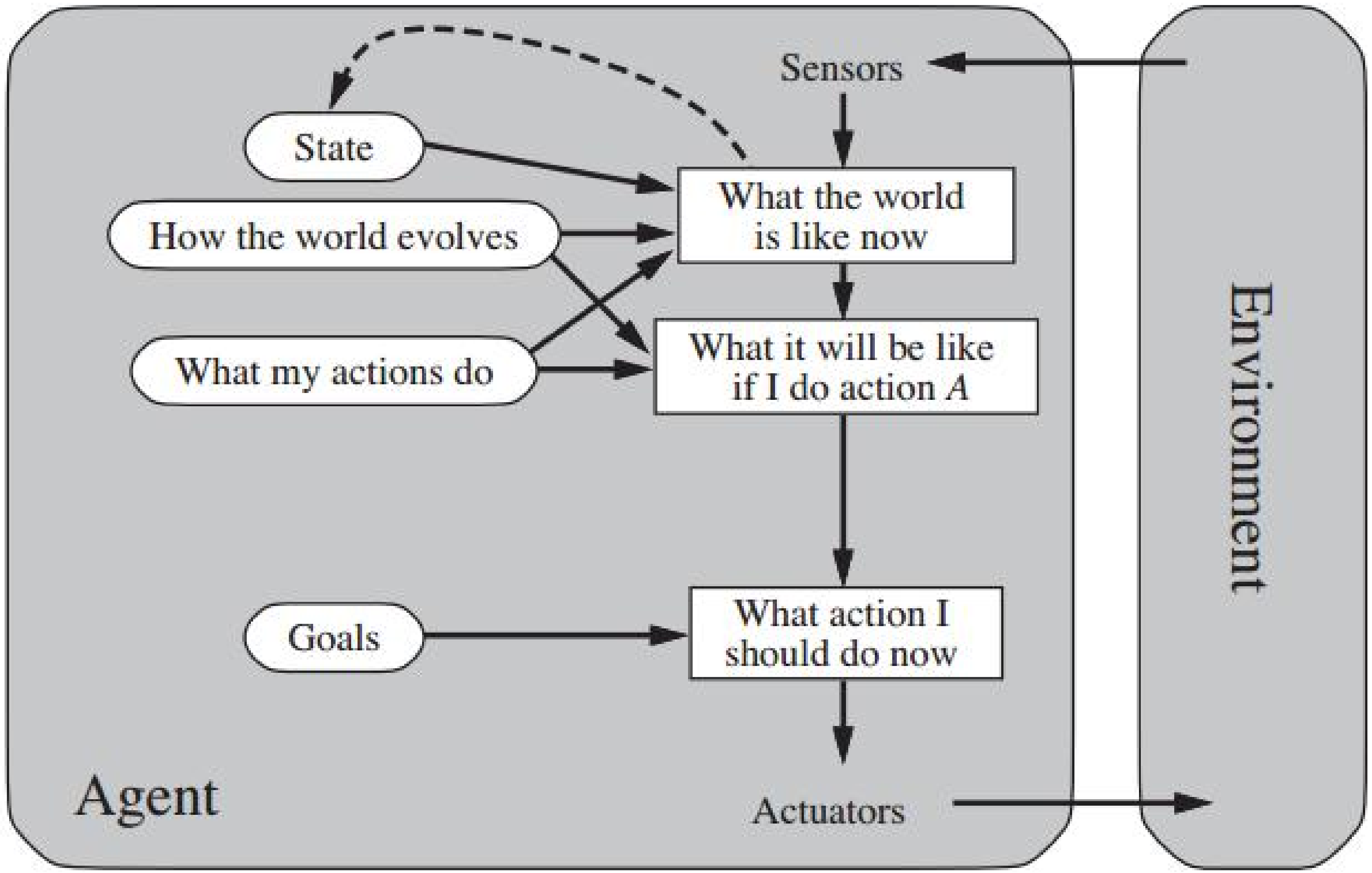
Model Based Agent



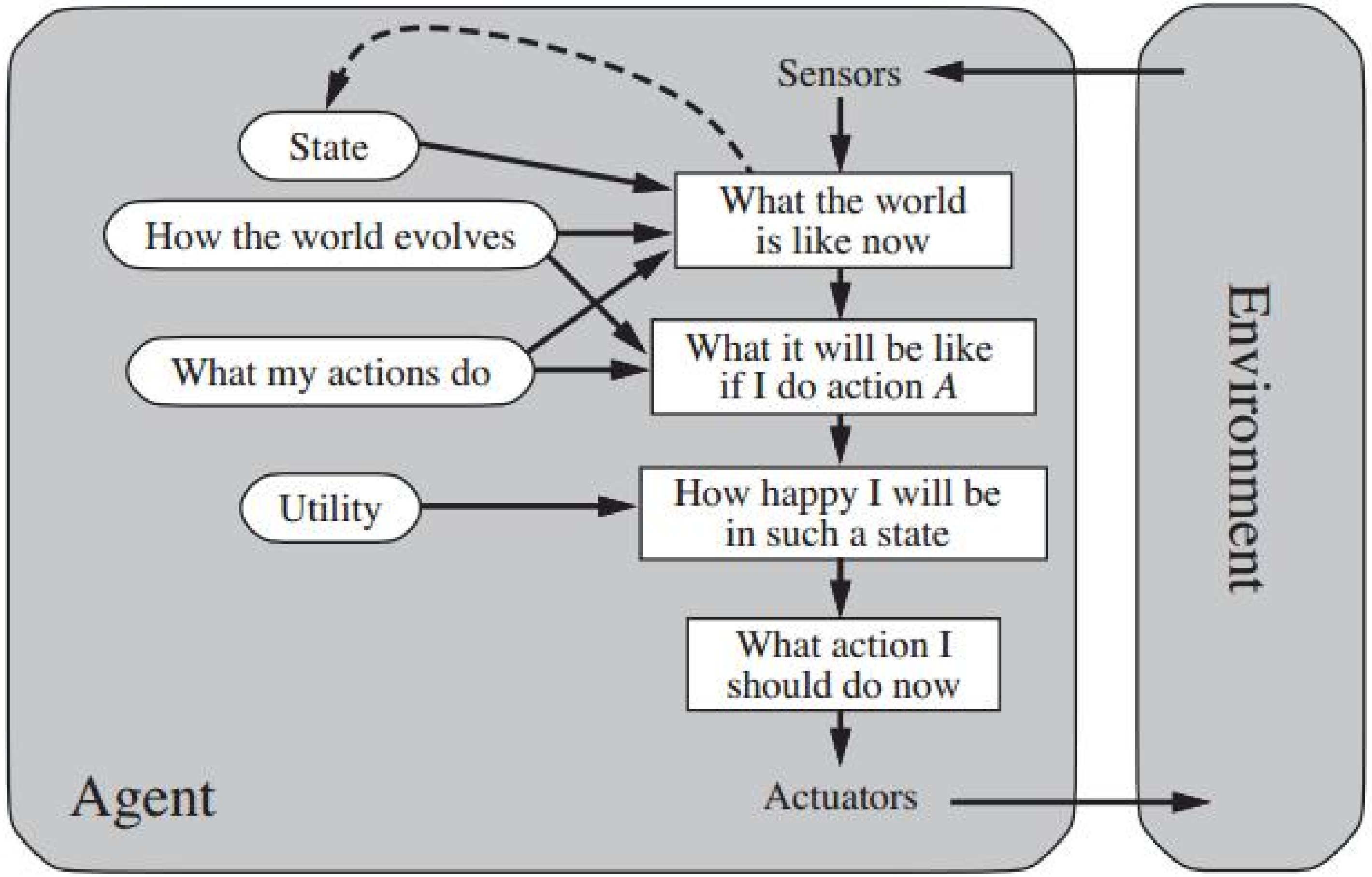
Types of Agent

- ***Goal-based agents:***
  - These agents have a set of goals they want to achieve and plan their actions based on these goals.
  - ***For example,*** a delivery drone that plans its route to minimize delivery time and avoid obstacles is a goal-based agent.
- ***Utility-based agents:***
  - These agents select actions based on a measure of the desirability of outcomes, called utility.
  - ***For example,*** a financial trading agent that buys and sells stocks based on maximizing profit is a utility-based agent.
- ***Learning agents:***
  - These agents improve their performance over time by learning from experience.
  - ***For example,*** a chatbot that learns from user interactions and improves its responses is a learning agent.

Goal Based Agent



Utility Based Agents



Types of Agent

- ***Reactive agents:***
  - These agents ***react to changes in their environment in real-time,*** without maintaining any internal state.
  - The car's sensors (e.g., cameras, radar, lidar) provide percepts of the current state of the road, such as the presence of other vehicles, pedestrians, traffic lights, and road signs.
- ***Deliberative agents:***
  - These agents ***reason about the world and generate plans*** to achieve their goals.
  - For example, a personal assistant agent that schedules meetings and optimizes travel plans is a deliberative agent.
- ***Multi-agent systems:***
  - These are ***collections of agents that work together to achieve a common goal.***
  - For example, a team of robots that collaboratively explore and map an unknown environment is a multi-agent system.

Rational Agent

- A ***rational agent*** is an intelligent agent that acts to achieve the best possible outcome or expected utility, based on its knowledge and reasoning about the world.
- In other words, a rational agent selects the best possible action based on its current knowledge and understanding of the world, given a specific goal or objective.
- To be ***rational***, an agent must have the ability to perceive its environment, reason about it, and act in a way that maximizes its expected utility.
- In practice, designing a ***rational agent*** can be challenging because it requires knowledge of the environment, the performance measure, and the decision-making rules.

Rational Agent

- Consider a ***stock trading system*** that uses machine learning algorithms to analyze market trends and identify profitable investments.
- The system is equipped with sensors that collect data on various market indicators, such as ***stock prices, trading volume, news headlines, and social media sentiment.***
- Based on this data, the system's rational agent calculates the expected utility of each potential investment, taking into account factors such as ***risk, return, and liquidity.***
- The agent then selects the investment with the ***highest expected utility*** and executes the corresponding trade.
- In this example, the rational agent is capable of making decisions that ***maximize the expected return on investment based*** on the available market data and the system's predefined investment goals.
- It uses a combination of sensing, modeling, and decision-making techniques to perform its task, and it can adapt its strategies over time based on the feedback it receives from the market.

Intelligent Agent

- An **intelligent agent** is an autonomous software or hardware system that can perform a specific task or set of tasks, based on its perception of the environment and its ability to reason and act upon that environment to achieve its goals or objectives.
- Intelligent agents can range from simple systems, such as chatbots or recommendation engines, to complex systems, such as autonomous vehicles, drones, or robots.
- These agents can be designed to operate in **different environments**, such as virtual or physical environments, and can interact with users or other agents in various ways.

PEAS(Performance, Environment, Actuator, Sensors)

- ***Performance measure:***
  - This is a metric that defines the success of an agent's behavior.
  - It specifies the criteria by which the agent's performance is evaluated.
  - For example, for a chess-playing agent, the performance measure could be the number of games won or the time taken to win a game.
- ***Environment:***
  - This is the external context in which the agent operates.
  - It includes all the entities and objects that the agent interacts with.
  - The environment can be physical, virtual, or a combination of both.
  - For example, the environment for a robot cleaning agent could be a physical room with furniture and objects.

PEAS(Performance, Environment, Actuator, Sensors)

- ***Actuators:***
  - These are the physical or virtual devices that the agent uses to act on the environment.
  - They are responsible for executing the actions that the agent decides to take.
  - For example, the actuators for a robot cleaning agent could be its motors and brushes.
- ***Sensors:***
  - These are the physical or virtual devices that the agent uses to perceive the environment.
  - They are responsible for collecting data about the state of the environment that the agent can use to make decisions.
  - For example, the sensors for a robot cleaning agent could be its cameras and sensors that detect dirt and obstacles.

Types of Environment

Fully observable vs. partially observable: An environment is fully observable if the agent's sensors provide complete information about the environment at each point in time. If the agent's sensors provide incomplete or partial information, then the environment is partially observable.

Deterministic vs. stochastic: An environment is deterministic if the next state of the environment is completely determined by the current state and the action taken by the agent. If the environment is not completely determined, then it is stochastic.

Episodic vs. sequential: An episodic environment consists of a series of independent episodes, where the agent's action in each episode is determined only by the current episode's state. In contrast, a sequential environment is a continuous process where the current decision depends on past decisions and future consequences.

Types of Environment

Fully observable vs. partially observable: Chess vs Poker

Deterministic vs. stochastic: Tic-tac-toe vs Ludo

Episodic vs. sequential: Sudoku vs Chess

Types of Environment

Static vs. dynamic: A static environment is an environment that remains unchanged while the agent is deliberating. In contrast, a dynamic environment is one that changes over time.

Discrete vs. continuous: A discrete environment consists of a finite number of distinct and well-defined states and actions. In contrast, a continuous environment has an infinite number of possible states and actions.

Single-agent vs. multi-agent: In a single-agent environment, there is only one agent operating in the environment. In contrast, a multi-agent environment involves multiple agents that interact with each other to achieve a common goal.

Types of Environment

Static vs. dynamic: temperature monitoring system vs Navigation through a traffic-filled city

Discrete vs. continuous: Playing chess vs Flying a drone in an open air space.

Single-agent vs. multi-agent: Solving a maze vs Playing a game of soccer with two teams.

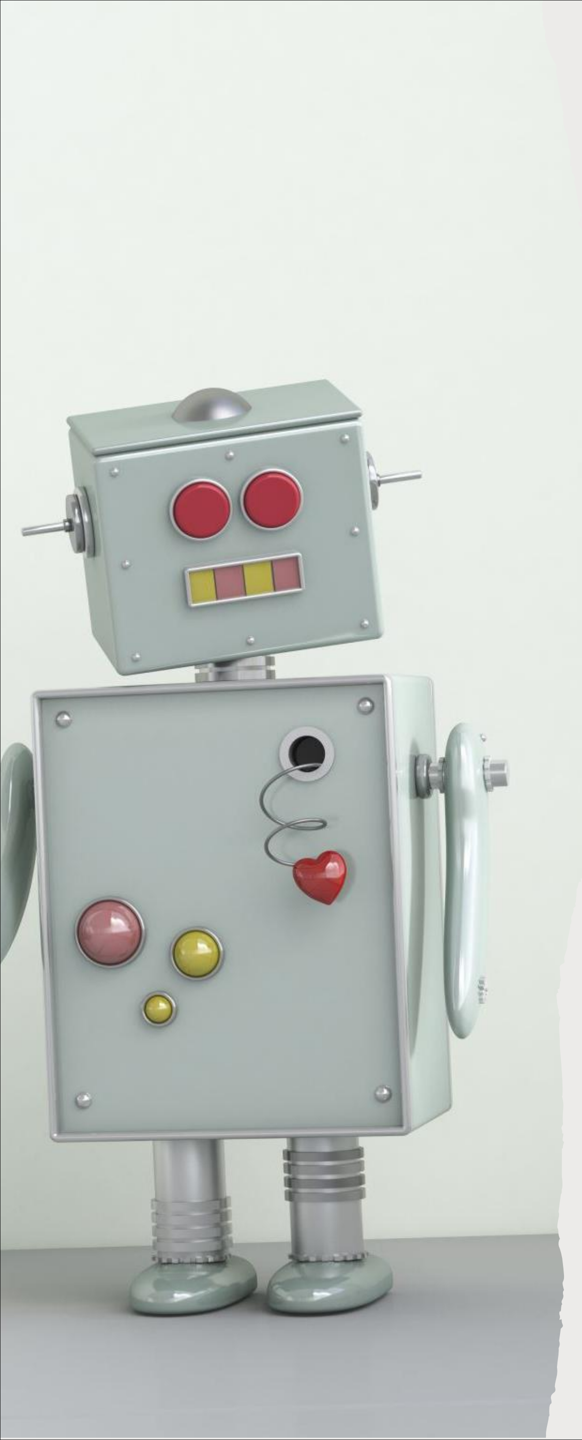
Agent Program

An agent program is a computer program that defines the behavior of an intelligent agent.

It is essentially a set of rules or instructions that govern how an agent perceives its environment, makes decisions, and takes actions.

The agent program receives inputs from sensors, processes the inputs, and then generates outputs that are sent to the actuators.

Overall, the design of an effective agent program requires a deep understanding of both the environment and the capabilities and limitations of the agent.



Agent Program

- Consider a simple vacuum cleaning robot that moves around a room, detects dirt on the floor, and cleans it up. The agent program for this robot might look like this:
  - ***Start the robot and initialize its sensors and actuators.***
  - ***Move forward until an obstacle is detected.***
  - ***When an obstacle is detected, turn left or right and continue moving.***
  - ***When dirt is detected, stop and activate the cleaning mechanism.***
  - ***After cleaning, continue moving and searching for more dirt.***
  - ***When the battery is low, return to the charging station and recharge.***
  - ***Repeat the process until the entire room is clean.***
- This is a simple rule-based program that defines the behavior of the ***vacuum cleaning robot***. It relies on the ***robot's sensors*** (such as infrared sensors or cameras) to detect obstacles and dirt on the floor, and its ***actuators*** (such as wheels or a cleaning mechanism) to move and clean.

Black- Box Model of Agent

A **black-box** model of an agent refers to a type of artificial intelligence model where the internal workings of the agent are ***not transparent or interpretable to an external observer.***

In other words, the model is treated as a ***"black box"*** where inputs are fed into the system, and outputs are generated ***without any understanding*** of how the agent arrives at those outputs.

In the context of intelligent systems, black-box models are often used in situations where the complexity of the problem or the size of the data set is too large for humans to understand or process.

Examples of such applications include natural language processing, image and speech recognition, and autonomous vehicles.

In these cases, the ***agent*** is designed to learn from data ***through trial-and-error*** and generate outputs based on its learned patterns.

Black- Box Model of Agent

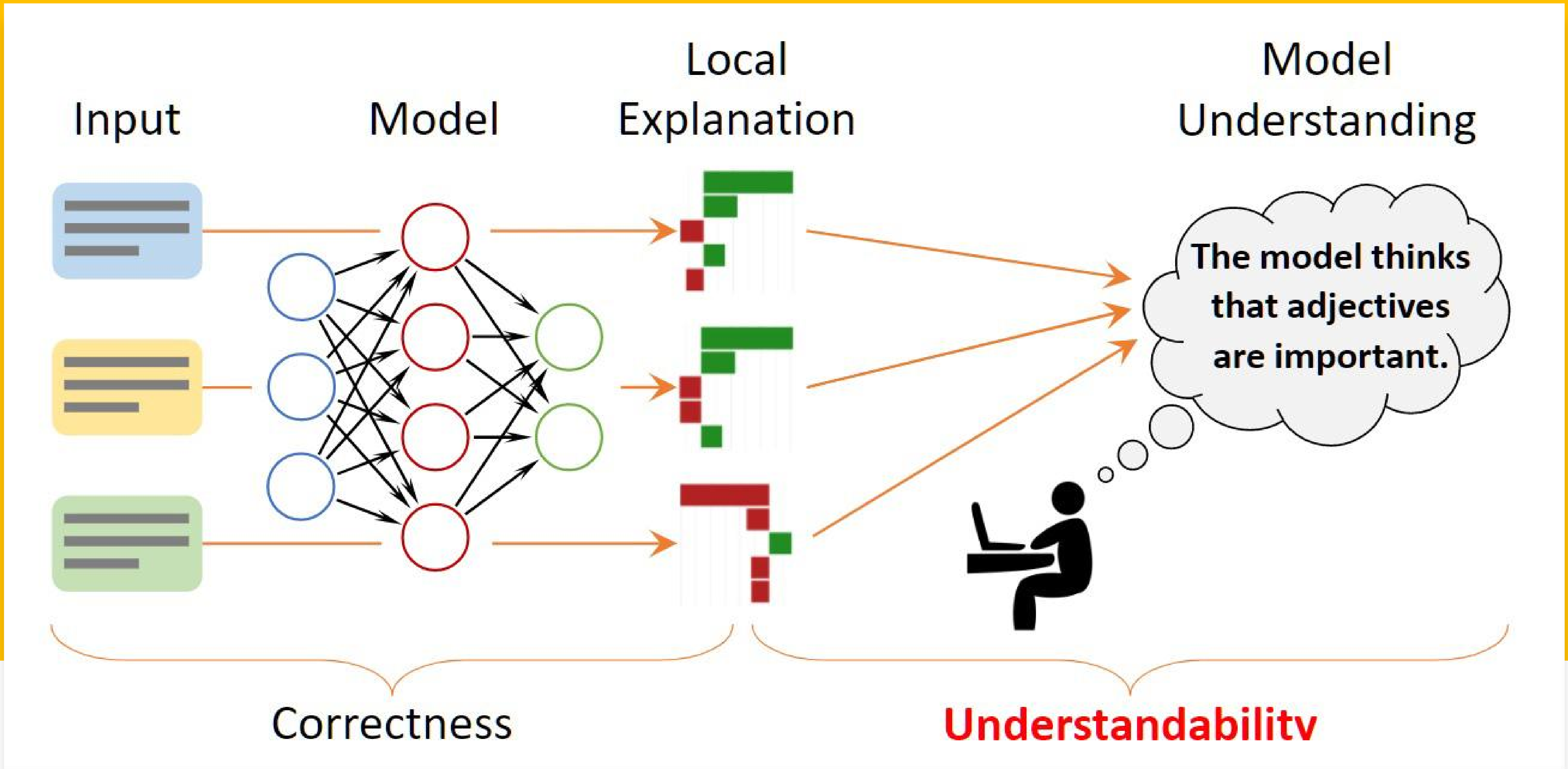
While black-box models can achieve high accuracy, they have several limitations.

One limitation is that the ***model's*** decisions cannot be easily explained or interpreted, which can be problematic in situations where transparency and accountability are crucial.

Another limitation is that black-box models can be ***susceptible to biases***, leading to unfair or discriminatory outcomes.

To address these limitations, researchers are exploring new techniques for designing more transparent and interpretable black-box models, as well as alternative models that are inherently more explainable.

These efforts aim to improve the reliability and trustworthiness of intelligent systems.


Input
Model
Local Explanation
Model Understanding
The model thinks that adjectives are important.
Correctness
Understandability

[redacted]

Agent	Human	Machine
Sensors	Eyes, ears, other organs	Cameras, IR finder
Actuators	Hands, legs mouth, other body parts	Various motors for actuators

Alpha Beta Pruning

- Alpha-beta pruning is an optimization technique used in game-playing algorithms, particularly in the minimax algorithm, to reduce the number of nodes that are evaluated during a search.
- In the minimax algorithm, a search tree is constructed to represent all possible moves and counter-moves for both players in a game.
- This search tree can be very large and it may be impossible to search through every possible move.
- Alpha-beta pruning allows the algorithm to search more efficiently by cutting off branches of the search tree that are unlikely to produce a better move.

Alpha Beta Pruning

- The basic idea behind alpha-beta pruning is to keep track of two values, alpha and beta, that represent the minimum and maximum values that the current player can achieve, respectively.
- As the algorithm searches through the tree, it updates these values and prunes branches of the tree that are no longer relevant based on the current values of alpha and beta.
- For example, if the algorithm is searching a branch of the tree and encounters a node that is worse for the current player than a previously evaluated node, it can stop searching that branch since the current player will never choose that move.
- This is because the previously evaluated node had a higher value for the current player, so they will choose that move instead.

Making Simple Game- Playing Agent for TTT

Define the game state: Tic Tac Toe is a two-player game played on a 3x3 board. You can represent the game state as a 3x3 matrix, where each element can take on three values: "X" for player 1's move, "O" for player 2's move, or "-" for an empty square.

Define the game rules: The goal of the game is to get three in a row, either horizontally, vertically, or diagonally. The first player to achieve this wins the game. If all squares are filled without any player achieving three in a row, the game is a draw.

Define the agent's actions: The agent's actions are the moves it can make on the board. The agent needs to choose an empty square to place its symbol ("X" or "O") on the board.

Making Simple Game- Playing Agent for TTT

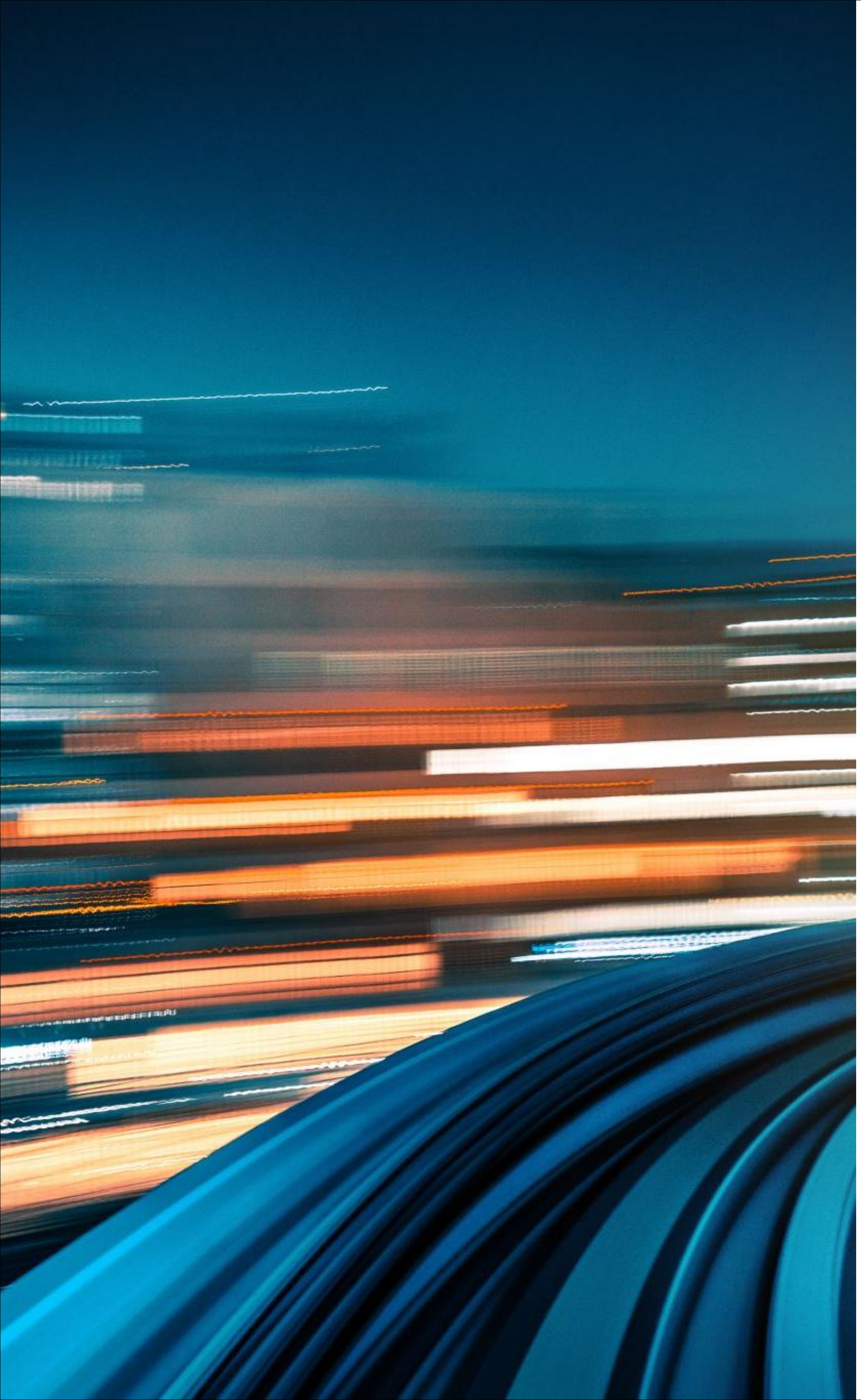
Define the agent's strategy: The agent's strategy should be based on the current state of the game and the desired outcome (win or draw). A simple strategy for the agent could be to choose the first available empty square.

Implement the game-playing agent: Write a program that takes the current state of the game and the player to move as inputs and returns the chosen move. You can use a simple algorithm like minimax or alpha-beta pruning to determine the best move for the agent.

Test the agent: Test the agent by playing it against a human player or against another game-playing agent. Evaluate its performance in terms of the number of wins, losses, and draws.

Intentionality and Goals

- Intentionality refers to the ability of an agent to have a mental state that represents something in the external world.
- In other words, it is the capacity to be about something, or to refer to something.
- This is a key feature of human cognition, and it is also an important aspect of artificial intelligence and agent-based systems.
- In intelligent agents, intentionality is closely related to the notion of goals.
- Goals are the desired outcomes or states that an agent aims to achieve in a given situation.
- An agent with intentionality is able to represent these goals in its mental states and use them to guide its behavior and decision-making.



Intentionality and Goals

- For example, a self-driving car can be designed with the goal of safely transporting passengers from one location to another.
- This goal can be represented in the car's programming and used to guide its behavior, such as selecting the safest route, obeying traffic laws, and avoiding collisions with other vehicles.
- Intentionality and goals are important concepts in the design of intelligent agents because they allow agents to act purposefully and adaptively in complex environments.
- By representing their goals and using them to guide their actions, agents can make intelligent decisions that take into account both their current state and the desired outcomes they are striving to achieve.

Evaluation Functions

- Evaluation functions are used to evaluate the desirability of different game states.
- It is a ***heuristic function that assigns a score to a given state of the game based on the current player's advantage.***
- For example, in chess, the evaluation function could assign higher scores to states where the player has more pieces, controls more space, or has more powerful pieces on the board.

Utilitarian

- Utilitarianism is a ***philosophical theory*** that states the best action is the one that maximizes overall happiness or pleasure.
- In artificial intelligence, utilitarianism is used to evaluate the utility of different actions or outcomes.
- For example, a self-driving car may be programmed to take the route that minimizes the total travel time, thus maximizing the overall utility of its passengers.

Planning

- Planning refers to the ***process of deciding on a course of action to achieve a specific goal.***
- In artificial intelligence, planning algorithms are used to generate a sequence of actions that will lead to a desired outcome.
- For example, a robot tasked with assembling a car may use a planning algorithm to determine the optimal sequence of actions to complete the task.

Internal Representation

- Internal representation refers to the ***mental models that an agent uses to reason about the world.***
- In artificial intelligence, internal representation is used to model the state of the world and the relationships between different objects or entities.
- For example, in a natural language processing system, words are represented as vectors in a high-dimensional space, allowing the system to reason about the meaning of different words and their relationships to each other.

Thank you