

Module (2)

Module (2)

Module 1 Recap: Introduction to AI

Key Topics Covered:

1. Introduction to AI:

Definition: Development of computer systems that can perform tasks requiring human intelligence.

Goals: Mimic or simulate human-like intelligence.

2. Definitions of AI by Researchers:

Thinking Humanly: Mimicking human thought processes.

Acting Humanly: Behaving like a human.

Thinking Rationally: Emulating rational thought processes.

Acting Rationally: Performing actions in a rational manner.

3. Types of AI:

Strong AI: Aims to replicate human intelligence across diverse tasks.

Weak AI: Focuses on specific tasks within a narrow domain.

4. Challenges in AI Development:

Adaptability and generalization issues.

Strong AI aims to exceed human intelligence in various domains.

Weak AI specializes in limited, specific tasks.

5. Foundations of AI:

Contributions from philosophy, mathematics, economics, neuroscience, and computing.

6. Turing Test:

Proposed by Alan Turing in 1950.

Evaluates a machine's ability to exhibit human-like intelligence.

Involves a human evaluator engaging in conversation with both a machine and a human without knowing which is which. If the evaluator cannot distinguish between the two, the machine is said to have passed the test.

7. AI Winters:

Periods of reduced funding and interest due to over-promising, under-delivering, lack of progress, technical challenges, and criticism from the scientific community.

8. Applications of AI:

Healthcare: Medical imaging, drug discovery, personalized medicine.

Finance: Algorithmic trading, fraud detection, customer service.

9. Transportation: Autonomous vehicles, traffic management, ride-sharing, navigation.
10. Entertainment: Game theory, content recommendation, content creation, gaming.

Module 2: Agents and Environments

Main Objective:

- Understand the fundamental concepts of different agents, including their definitions and behaviors in various environments.

Module Breakdown:

1. **Agents and Environments Overview:**
Introduction to agents and their environments.
2. **Concept of Rationality:**
Understanding the concept of rationality in AI.
3. **Nature of Environments:**
Detailed analysis of different types of environments in which agents operate.
4. **Structure of Agents:**
Examining the components that make up an agent.
5. **Knowledge-Based Agents:**
Discussion on knowledge-based agents with practical examples.

Agents and Their Components

1. **Definition of an Agent:**
An agent is anything that perceives its environment through sensors and acts upon that environment through actuators.
2. **Key Components:**
Agent: Monitors and observes its environment.
Environment: The surroundings or conditions in which an agent operates.
Sensors: Tools through which an agent perceives its environment.
Actuators: Mechanisms through which an agent acts upon its environment.
3. **Agent's Percept and Percept Sequence:**
Percept: The input received by the agent from its environment.
Percept Sequence: The history of all percepts received by the agent over time.
4. **Agent Function:**
 1. Describes the agent's behavior by mapping any given percept sequence to an action.

2. The choice of action depends on the entire percept sequence observed up to that point.

By understanding these concepts, we can better analyze and design intelligent agents capable of performing tasks in various environments. This module will delve deeper into the specifics of how agents interact with their environments and make rational decisions based on their perceptual inputs.

Guided Notes on Agent and Environment in AI

Introduction to Agent and Environment

1. **Agent Definition:**

An agent perceives its environment through sensors and acts upon that environment through actuators.

Sensor: Collects data from the environment (percepts).

Actuator: Executes actions in the environment based on processed data.

2. **Agent-Environment Interaction:**

Input: Sensors gather data from the environment.

Processing: The agent processes the data to determine the appropriate action.

Output: Actuators perform the determined actions in the environment.

Diagram Explanation

Visualization of Agent and Environment:

The agent interacts with the environment.

Sensors receive inputs from the environment.

Actuators provide outputs or actions based on the agent's processing.

Standard Embedded System Structure vs. AI Agent

1. **Standard Embedded System:**

Sensors: Capture analog data from the environment.

ADC (Analog to Digital Converter): Converts analog signals to digital.

Processing Unit: Microcontroller, ASIC (Application-Specific Integrated Circuit), or FPGA (Field Programmable Gate Array) processes the digital data.

DAC (Digital to Analog Converter): Converts processed digital signals back to analog.

Actuators: Execute actions based on the processed data.

2. **AI Agent:**

Similar structure but focuses more on intelligent processing and decision-making.

Sensors and actuators are still key components.

The processing is more sophisticated, involving algorithms to decide the best action based on percepts.

Applications and Differences

1. AI Applications:

Common in AI applications like robotics, multi-agent systems, and intelligent software systems.

Focus on dynamic and uncertain environments where agents must adapt and make intelligent decisions.

2. Embedded Systems:

Designed for specific functions within larger systems.

Prioritize real-time operation, low power consumption, and reliability.

Often operate in constrained environments like autonomous control systems, industrial automation, consumer electronics, and medical devices.

Key Concepts Recap

Agent: Entity that perceives and acts.

Environment: External context of the agent's operation.

Sensor: Device that captures data from the environment.

Actuator: Device that executes actions in the environment.

Percept: Data received by the agent from the environment.

Percept Sequence: History of all percepts received by the agent.

Agent Function: Maps percept sequences to actions.

These notes encapsulate the fundamental concepts of agents and environments, focusing on their interactions and applications both in AI and embedded systems. Understanding these principles is crucial for developing intelligent systems capable of operating autonomously in diverse environments.

Module 2: Agents and Environments

Types of Agents

1. Human Agent:

Sensors: Eyes, ears, and other sensory organs.

Actuators: Hands, legs, vocal tract.

Function: Human agents use sensory organs to perceive their environment and actuators to interact and perform actions.

2. Robotic Agent:

Sensors: Infrared range finders, Natural Language Processing (NLP) systems, cameras.

Actuators: Various motors.

Function: Robotic agents gather information using sensors and execute physical actions using motors.

3. **Software Agent:**

Sensors: Keystrokes, file contents.

Actuators: Display outputs on the screen.

Function: Software agents autonomously perform tasks, perceive their environment, make decisions, and take actions based on predefined goals.

Communication: Interact with other agents or users through communication protocols.

Applications: Information retrieval, e-commerce, cybersecurity, automation.

Rationality and Simple Agent Examples

1. **Rationality in Agents:**

An agent is considered rational if it acts to achieve its goals based on the percept sequence it has received.

Rationality involves choosing actions that maximize performance, given the percepts and built-in knowledge.

2. **Vacuum Cleaner Agent Example:**

Environment: A simple world with two locations, A and B.

Task: Clean a square if it is dirty, move to the other square if not.

Perception: The agent perceives the current location and whether it is dirty.

3. **Agent Function:**

Simple Agent Function:

If the current square is dirty, clean it.

If the current square is clean, move to the other square.

Example Steps:

If at location A and A is dirty, clean A.

If at location A and A is clean, move to location B.

If at location B and B is dirty, clean B.

If at location B and B is clean, move to location A.

4. **Partial Tabulation Example:**

- **Scenario:**

At location A, if A is clean, move to B.

At location A, if A is dirty, clean A.

At location B, if B is clean, move to A.

At location B, if B is dirty, clean B.

5. **Good vs. Bad Agents:**

A good agent performs actions that consistently achieve its goals.

An intelligent agent optimizes its actions based on percepts and built-in

knowledge.

A bad or unintelligent agent may perform inefficient or incorrect actions.
a good sequence of inputs might train the agent better

Conclusion

This module provided an overview of different types of agents and their components, including human agents, robotic agents, and software agents. We discussed how agents interact with their environments using sensors and actuators. We also introduced the concept of rationality in agents, illustrated with a simple vacuum cleaner agent example. In the next module, we will delve deeper into the concept of rationality and good behavior in agents.

Concept of Rationality and Good Behavior in AI

Rationality in AI

Rational Agent: An agent that consistently makes decisions and takes actions to achieve its goals and objectives. It accurately perceives its environment, processes information effectively, and selects actions that maximize its expected utility or performance measure.

Agent Function: A mapping of percept sequences (inputs from the environment) to actions. Every entry in the agent function table must be correctly filled to ensure rational behavior.

Defining Good Behavior

Doing the Right Thing: Good behavior entails making optimal decisions based on available information and objectives, reflecting intelligent, goal-oriented actions.

Performance Measure: A metric to evaluate the agent's actions by assessing the sequence of environment states resulting from those actions. The performance measure is crucial to determine if the agent performed well.

Cause and Effect: The relationship between the agent's actions and the resulting changes in the environment. If the actions lead to desirable environment states, the agent is considered to have performed well.

Key Concepts in Rationality and Performance

Percept and Percept Sequence:

Percept: The input received by the agent from its environment.

Percept Sequence: The history of all percepts received over time.

Performance Measure: Evaluates the agent's performance based on the sequence

of environment states rather than the agent's internal states. It must align with the agent's goals and provide meaningful insights into its performance.

Example: Vacuum Cleaner Agent

Proposed Performance Measure: Amount of dirt cleaned in a given time (e.g., an eight-hour shift).

Improved Performance Measure: Rewarding the agent for maintaining a clean floor, with points awarded for each clean square at each time step, and possibly penalties for electricity consumption or noise generated.

Design Considerations for Performance Measures

Task-Specific Measures: The performance measure should be tailored to the specific task and environment, providing a quantitative metric for the agent's effectiveness.

Avoiding Misleading Measures: The measure should reflect the desired outcomes in the environment rather than how one thinks the agent should behave. For instance, instead of measuring the amount of dirt cleaned, measure the cleanliness of the floor over time.

Summary

Rationality: Making decisions that lead to goal achievement based on perceptual inputs.

Good Behavior: Achieving desired outcomes in the environment through intelligent actions.

Performance Measure: A critical component that should be task-specific and aligned with environmental goals to accurately assess the agent's effectiveness.

This detailed understanding of rationality and good behavior in agents provides a foundation for designing and evaluating AI systems that can effectively interact with their environments to achieve specified goals.

Module 2: Concept of Rationality

Rationality Overview:

Rationality in the context of intelligent agents refers to making decisions that maximize the performance measure, given the agent's perceptual input and prior knowledge. At any given time, rationality depends on four key factors.

Four Key Factors of Rationality:

1. Performance Measure:

Definition: Criteria for success, serving as a benchmark to assess outcomes.

Function: Quantifies or compares the effectiveness or efficiency of strategies or actions.

Dependence: Varies based on the goal, objective, and requirement of the system or agent. Context-specific and application-specific.

Purpose: Helps determine how well an agent is performing in achieving desired outcomes.

2. Agent's Prior Knowledge of the Environment:

Definition: Knowledge that includes facts, rules, assumptions, or patterns learned or provided through training or experience.

Function: Forms the foundation for making informed decisions and adapting behavior.

Impact: Directly influences the agent's performance measure. Accurate and extensive prior knowledge enhances the agent's effectiveness in achieving goals.

Importance: Updating and leveraging prior knowledge appropriately is crucial for improving the capabilities of the intelligent agent.

3. Actions that the Agent Can Perform:

Definition: Various ways the agent can interact with its environment to achieve its goals.

Factors Influencing Actions: Agent capabilities, constraints, and specific task requirements.

Types of Actions: May include physical movements, communication, decision-making processes, or any interaction that can influence the environment.

Purpose: Effective utilization of available actions is essential for navigating the environment, making decisions, and achieving objectives successfully.

4. Agent's Percept Sequence to Date:

Definition: The chronological sequence of perceptual information gathered by the agent from its environment.

Sources: Sensory inputs from sensors, cameras, microphones, etc.

Function: Helps build a model of the environment, track changes over time, and make informed decisions based on past observations.

Importance: Valuable for guiding actions and adapting to dynamic environments.

Rational Agent Behavior:

Objective: For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure.

Criteria: The action choice should be based on the evidence provided by the percept sequence and the agent's built knowledge.

Practical Example: Teacher in a Classroom

Components:

Agent: Teacher

Environment: Classroom

Actuators: Slides, speaker, discussion forum

Sensors: his PhD degrees and Knowledge of AI, trends in AI, instructions as per the degree patterns

Performance Measure: Feedback from students

Rationality in Practice:

The teacher (agent) uses his knowledge and trends to get what he has to teach(sensors)

Slides (actuators) are used to deliver information.

Feedback from students (performance measure) indicates the success of the teaching methods.

Summary of Rationality Concept

1. Agent
2. Environment
3. Actuators
4. Sensors
5. Performance Measure

To define rational behavior in an agent, five key components are essential:

Understanding and integrating these components ensure that an agent can make informed and rational decisions to achieve its objectives effectively.

With this, we conclude the module on the concept of rationality and good behavior in intelligent agents.

Module 2: Agents and Environments

Sub-module Recap: Types of Agents

Human Agent

Robotic Agent

Software Agent

Performance Measure and Task Environments

Key Points:

1. **Performance Measure:**

Dependent on the nature of the environment and the type of objective.
Critical for evaluating how well an agent performs in its given environment.

2. **Task Environment:**

Defines the problem space for which rational agents are solutions.
Specifies the task environment for a specific problem.
Includes various examples to illustrate different task environments.

3. **PEAS Framework:**

Performance Measure: Criteria to evaluate the success of the agent.

Environment: The surrounding conditions and entities the agent interacts with.

Actuators: Mechanisms through which the agent acts upon the environment.

Sensors: Tools through which the agent perceives the environment.

Designing Agents for Task Environments

Step-by-Step Approach:

1. **Define the Task Environment:**

As fully as possible, specify the task environment before designing the agent.

2. **Example: Vacuum Cleaner Agent**

Sensors: Detect dirt, wall bumpers.

Actuators: Vacuum motor, wheels.

3. **Complex Example: Automated Taxi Driver**

Performance Measure:

Safety

Fast travel times

Profit maximization

Environment:

Roads, other traffic, pedestrians, customers

Actuators:

Steering wheel, accelerator, brake, signal, horn, display

Sensors:

Cameras, sonar, speedometer, GPS, engine sensors, keyboard

Examples of Task Environments

Automated Taxi Driver:

Performance Measure:

Correct destination, minimized fuel consumption, minimized trip time and cost, minimized traffic law violations, passenger safety and comfort.

Environment:

Various roads, traffic, pedestrians, stray animals, roadworks, police cars, potholes.

Actuators:

Engine control, steering, braking, display screen, voice synthesizer.

Sensors:

Controllable video cameras, infrared or sonar sensors, speedometer, accelerometer.

Medical Diagnosis System:

Performance Measure:

Healthy patients, minimized cost and lawsuits.

Environment:

Patients, hospital staff.

Actuators:

Screen display, diagnostic questions, tests, treatment referrals.

Sensors:

Keyboards for symptom entry, patient responses.

Chess Agent:

Performance Measure:

Win, loss, or draw.

Environment:

Chessboard, opponent.

Actuators:

Moving pieces.

Sensors:

Board positions.

Refinery Controller:

Performance Measure:

Purity, yield, safety.

Environment:

Refinery operations.

Actuators:

Valves, pumps, heaters, displays.

Sensors:

Temperature, pressure, chemical sensors.

Playing Soccer:

Performance Measure:

Score, injuries, teamwork.

Environment:

Players, referees, field, crowd, goals, ball.

Actuators:

Player strength, stamina, coordination, limbs.

Sensors:

Eyes, ears, mouth, touch.

Exploring Titan:

Performance Measure:

Mobility, safety, data collection, navigation.

Environment:

Shuttle, rover, atmosphere, surface, ocean.

Actuators:

Communication devices, sustainability systems, reliability mechanisms.

Sensors:

Cameras, GPS, temperature and pressure sensors.

AI Bookshop:

Performance Measure:

Pricing, ease of use, shipping time.

Environment:

Website, internet users.

Actuators:

Information display, user interactions.

Sensors:

Image recognition, user inputs.

Page 33 39 45 for more examples including these

OneDrive (sharepoint.com)

Conclusion

For each type of agent and environment, define the Performance Measure , Environment , Actuators , and Sensors.

The PEAS framework helps in specifying the task environment comprehensively.

By understanding these elements, you can design agents that are well-suited to their specific tasks and environments.

By following these guidelines, you will be equipped to analyze and design intelligent agents capable of performing effectively in various environments.

Module 2: Agents and Environments

Understanding the Environment

Definition:

The environment is everything in the world surrounding the agent but is not part of the agent itself.

It provides the agent with situations to sense and act upon.

Properties of the Environment:

1. Fully Observable vs. Partially Observable:

Fully Observable: The agent's sensors can access the complete state of the environment at any given time.

Example: Chess game.

Partially Observable: The agent's sensors cannot access the complete state of the environment.

Example: Poker game.

Unobservable: The agent has no sensors to perceive the environment.

Example: A completely blindfolded agent in a room.

2. Static vs. Dynamic:

Static: The environment does not change while the agent is deliberating.

Example: Crossword puzzle.

Dynamic: The environment can change while the agent is deliberating.

Example: Taxi driving.

Static: Easier to deal with as the agent does not need to continuously observe the environment.

Dynamic: Requires constant observation and updates.

3. Discrete vs. Continuous:

Discrete: The environment has a finite number of percepts and actions.

Example: Chess game.

Continuous: The environment has an infinite number of percepts and actions.

Example: Self-driving car.

4. Deterministic vs. Stochastic:

Deterministic: The next state of the environment is completely determined by the current state and the agent's action.

Example: Chess game.

Stochastic: The next state of the environment is not completely determined by the current state and the agent's action.

Example: Poker game.

5. Single Agent vs. Multi-Agent:

Single Agent: Only one agent operates in the environment.

Example: Solitaire.

Multi-Agent: Multiple agents operate in the environment.

Example: Soccer game.

Multi-agent design problems differ significantly from single-agent problems.

6. **Episodic vs. Sequential:**

Episodic: The agent's actions are divided into discrete episodes. Each episode is independent.

Example: Image classification.

Sequential: The current decision could affect all future decisions.

Example: Chess game.

7. **Known vs. Unknown:**

Known: The agent knows the outcomes of its actions.

Example: A known rule-based game.

Unknown: The agent does not know the outcomes of its actions and must learn them.

Example: Real-world navigation.

8. **Accessible vs. Inaccessible:**

Accessible: The agent can obtain complete and accurate information about the state of the environment.

Example: An empty room with a thermostat.

Inaccessible: The agent cannot obtain complete and accurate information about the state of the environment.

Example: Predicting weather events.

Applying Environment Properties to Tasks

Examples:

1. **Crossword Puzzle:**

Fully Observable

Single Agent

Deterministic

Sequential

Static

Discrete

2. **Chess with Clock:**

Fully Observable

Multi-Agent

Deterministic

Sequential

Static

Discrete

3. **Poker:**

Partially Observable

Multi-Agent

Stochastic

Sequential

Static
Discrete

4. **Backgammon:**

Fully Observable
Multi-Agent
Stochastic
Sequential
Static
Discrete

5. **Taxi Driving:**

Partially Observable
Multi-Agent
Stochastic
Sequential
Dynamic
Continuous

6. **Medical Diagnosis:**

Partially Observable
Single Agent
Stochastic
Sequential
Static
Continuous

7. **Image Analysis:**

Fully Observable
Single Agent
Deterministic
Episodic
Static
Continuous

8. **Part Picking Robot:**

Partially Observable
Single Agent
Deterministic
Sequential
Dynamic
Discrete

9. **Refinery Controller:**

Fully Observable
Single Agent
Deterministic
Sequential

Dynamic
Continuous

Conclusion

In this module, we covered the concept of environments in AI, including various properties that define them and how these properties impact the tasks agents perform. Understanding the type of environment an agent operates in is crucial for designing effective AI systems.

Structure of Agents

1. Agent Program:

Implements the agent function, mapping percepts to actions.

Takes the current percept as input and returns an action to be performed by the actuators.

2. Agent Architecture:

The physical system (hardware) on which the agent program runs.

Includes sensors to perceive the environment and actuators to interact with it.

Must be compatible with the agent program to ensure it can perform the desired actions.

Overview

In this section, we'll dive into the internal workings of an agent, exploring how it functions and interacts with its environment. We'll focus on the design of the agent program that maps percepts to actions and the architecture that supports this program.

Key Concepts

Designing an Agent

To design an intelligent agent, we need to consider several components:

- Sensors: Gather information from the environment.
- Actuators: Execute actions based on the agent's decisions.
- Knowledge Representation: Store and manage the knowledge the agent uses.
- Decision-Making Algorithms: Determine the best actions based on percepts and knowledge.
- Memory: Keep track of past percepts and actions.

Types of Agent Programs

Agents can be classified based on their complexity and how they interact with the environment:

1. **Simple Reflex Agents:**

Behavior: React to current percepts without considering history.

Example: Thermostat adjusting temperature based on current reading.

Limitations: Cannot handle complex or dynamic environments effectively.

2. **Model-Based Reflex Agents:**

Behavior: Maintain an internal model of the world to handle partially observable environments.

Components: Internal state to keep track of parts of the environment that are not directly observable.

3. **Goal-Based Agents:**

Behavior: Act to achieve specific goals.

Components: Goals in addition to the internal state, allowing for planning and decision-making.

4. **Utility-Based Agents:**

Behavior: Maximize a utility function that represents performance.

Components: Utility function to evaluate different actions and choose the best one.

Simple Reflex Agents in Detail

Characteristics:

Operate on condition-action rules.

Ignore the history of percepts and only respond to the current state.

Effective for simple, predictable environments.

Example: Thermostat

Task: Maintain room temperature within a specific range (e.g., 16°C to 22°C).

Rules:

If temperature < 16°C, turn on the heater.

If temperature > 22°C, turn off the heater.

Implementation in Pseudocode:

```
def simple_reflex_agent(percept):
    rules = {
        'temp < 16': 'turn on heater',
        'temp > 22': 'turn off heater'
    }
    state = interpret_input(percept)
    rule = match_rule(state, rules)
    action = rule['action']
```

```
return action
def interpret_input(percept):
    # Interpret the percept to get the current state
    state = {"temperature": percept}
    return state
def match_rule(state, rules):
    # Find the rule that matches the current state
    for condition, action in rules.items():
        if eval(condition):
            return {'condition': condition, 'action': action}
    return None
```

In this example:

Interpret Input: Converts percept (temperature reading) into a state.

Match Rule: Finds the appropriate rule based on the current state.

Action: Executes the action specified by the matched rule.

Summary

The structure of agents involves designing the agent program and architecture to effectively map percepts to actions. Simple reflex agents are the most basic form, operating based on current percepts without considering history. As we progress, more complex agents incorporate internal models, goals, and utility functions to handle more sophisticated environments and tasks. This foundational understanding sets the stage for creating intelligent systems capable of adapting and functioning autonomously in diverse settings.

Model-Based Reflex Agent

Definition

Model-Based Reflex Agent : An artificial agent that incorporates an internal model of its environment to make decisions and take actions.

Partial Observability : When the agent cannot sense the complete state of the environment at each point in time.

Characteristics

Maintains a representation of the environment state.

Uses this internal model to anticipate the consequences of actions.

Example: An automatic dishwasher that senses the state of the dishes, water temperature, and detergent level to efficiently clean dishes.

Perception and Action

Perception : Sensing the state of the dishes and water temperature.

Action : Selecting appropriate actions based on the perceived state to achieve the goal of cleaning dishes.

E.g., If dishes are dirty and water temperature is low, fill the basin with hot water and activate the detergent dispenser.

If dishes are clean and water temperature is sufficient, drain the dirty water and initiate the drying cycle.

Decision Making Process

Incorporates current perception and internal model of the environment.

The internal model helps in making more informed decisions compared to a simple reflex agent.

Internal State

Maintains an internal state based on perceived history.

Updating the Internal State :

Requires knowledge about how the world evolves independently of the agent.

Knowledge of how the agent's actions affect the world.

Model of the World : Implemented in the agent program to reflect the world's workings.

Representation : Includes sensors, internal models, states, rules, and actions.

Components

Perception : Capturing the current state of the environment through sensors.

Internal Model : Maintains a representation of the environment, including its state and evolution over time.

Decision Making : Selects actions likely to lead to desirable outcomes based on perception and internal model.

Action Execution : Executes the chosen actions.

Goal-Based Agent

Definition

Goal-Based Agent : An agent that requires goal information to decide on actions.

Current state information alone is insufficient; goals are necessary for decision making.

Characteristics

Goal Information : Describes desirable situations (e.g., destination for a taxi).

Combines current state description with goal information to choose actions.

Example

Taxi at a Junction :

The taxi needs to decide whether to turn left, turn right, or go straight based on the goal (passenger's destination).

Components

Goal Setting : Agents have predefined goals or objectives to achieve.

Perception : Captures relevant information from the environment.

Goal Representation : Maintains the goal information.

Decision Making and Execution : Chooses and executes actions to achieve goals.

Utility-Based Agent

Definition

Utility-Based Agent : Uses a utility function to measure preference among different world states and actions.

Characteristics

Utility Function : Measures the desirability or value of different states or outcomes.

Allows comparison of actions based on utility (e.g., quicker, safer, more reliable, cheaper).

Provides a more general performance measure compared to goal-based agents.

Example

Autonomous Delivery System :

Goal: Deliver packages efficiently.

Perception : Uses sensors like cameras, lidar, and GPS to detect obstacles, traffic, and delivery locations.

Utility Function : Quantifies desirability considering factors like delivery time, energy consumption, and safety.

Decision Making : Evaluates routes and actions based on expected utility.

Components

Goal Representation : Describes the desired outcome.

Utility Function : Assigns numerical value (utility) to each possible outcome.

Decision Making : Chooses actions to maximize expected utility.

Execution : Carries out the chosen actions.

Summary

Model-Based Reflex Agent : Uses internal models for decision making in partially observable environments.

Goal-Based Agent : Requires goal information to decide actions in addition to current state information.

Utility-Based Agent : Uses a utility function to compare and select actions that maximize overall utility.

Knowledge-Based Agent

Definition

Knowledge-Based Agent : An agent that uses a base of knowledge to make decisions and solve problems.

Human Analogy

Human Knowledge : Humans know things and use this knowledge to perform tasks, such as driving a car.

Driving Example :

Knowledge of traffic rules, road signs, and driving techniques.

Acquired through education, training, and experience.

Allows navigation through various traffic conditions, informed decision-making, and appropriate reactions to unforeseen situations.

Key Points

Influence of Knowledge on Actions :

Knowledge of red lights, stopping for the car in front, pedestrian signals, and adjusting speed based on road conditions.

Continuous application of knowledge to assess and adapt to changing circumstances on the road.

Human Intelligence :

Achieved not just by reflexes but by reasoning processes operating on internal knowledge representations.

Knowledge-Based Agents in AI :

Embodies knowledge and reasoning processes.

Knowledge Base :

Central component of a knowledge-based agent.

A set of sentences expressed in a knowledge representation language.

Represents assertions about the world.

Components of Knowledge-Based Agent

1. Knowledge Base :

Stores information about the world.

Uses a knowledge representation language to express facts and rules.

2. Inference Engine :

Processes the information in the knowledge base.

Applies logical rules to draw conclusions and make decisions.

Learning Agent

Definition

Learning Agent : An agent that improves its performance over time through learning.

Key Components

1. Performance Element :

Responsible for selecting external actions.

The part of the agent that interacts with the environment.

2. Learning Element :

Responsible for making improvements.

Uses feedback from the environment to enhance performance.

3. Critic :

Provides feedback on the agent's actions.

Helps the learning element by evaluating performance.

4. Problem Generator :

Suggests actions to explore new possibilities.

Aims to improve the agent's performance by proposing alternative actions.

Process

The performance element perceives the environment and decides actions.

The learning element modifies the performance element based on feedback from the critic.

The problem generator explores new actions to improve learning and adaptation.

Summary of Module

Key Points to Recall

1. Agent and Agent Function :

An agent acts in an environment.

The agent function specifies actions based on percept sequences.

2. Performance Measure :

Evaluates the behavior of the agent in an environment.

Rational agents maximize the expected value of the performance measure.

3. Task Environment :

Specifies the performance measure, environment, actuators, and sensors.

Essential to fully specify the task environment when designing an agent.

4. Agent Program :

Implements the agent function.

Designs vary based on efficiency, compactness, and flexibility.

Appropriate design depends on the environment.

5. Types of Agents :

Simple Reflex Agent : Responds directly to percepts.

Model-Based Reflex Agent : Maintains an internal state to track unobserved aspects.

Goal-Based Agent : Acts to achieve specific goals.

Utility-Based Agent : Maximizes expected happiness or utility.

6. Knowledge-Based and Learning Agents :

Knowledge-based agents use a base of knowledge for reasoning and decision-making.

Learning agents improve their performance over time through feedback and adaptation.

Conclusion

In the next module, the focus will be on problem-solving techniques in AI.