UNIT 1: Introduction

Concept of AI, history, current status, scope, agents, environments, Problem Formulations,Review of tree and graph structures, State space representation, Search graph and Search tree.

What is Artificial Intelligence (AI) ?

Artificial Intelligence (AI) is when **machines think and act smart** like humans. It helps computers **learn from experience, solve problems, understand language, and make decisions** without being directly told what to do.

Examples of AI in Daily Life:

- **Google Search** → AI helps find answers quickly.
- **Voice Assistants (Alexa, Siri)** → AI listens and responds to questions.
- **Self-Driving Cars** → AI helps cars drive without human control.
- **Chatbots** → AI answers customer queries automatically.
- **Recommendation Systems** → AI suggests movies (Netflix), songs (Spotify), and products (Amazon).

How Does AI Work?

AI **learns from data** using techniques like **Machine Learning** and **Deep Learning** to improve over time. It tries to **think, learn, and make decisions** like a human brain.

AI works by **collecting data, analyzing patterns, and learning from examples.** Algorithms adapt over time, improving their accuracy with more information. They mimic human thought processes, solving tasks like recognizing images, understanding language, or making decisions. This self-improvement cycle defines how AI grows smarter and achieves truly remarkable capabilities continually.

History of AI

1950s: Birth of AI

- **Alan Turing's Turing Test**: A test to check if a machine can think like a human.
- **AI as a Field**: The term **"Artificial Intelligence"** was introduced.

1960s: Early AI Programs

- AI programs focused on solving **math problems** and **playing chess**.
- AI research expanded, and universities started **AI labs**.

1980s: Machine Learning & Expert Systems

- **Machine Learning**: AI started learning from data using **neural networks**.
- **Expert Systems**: AI was used to **make decisions like human experts**.

2000s: Big Data & Industry Use

- **Big Data**: More data and powerful computers helped AI improve.
- **AI in Industry**: Used in **healthcare, banking, and security**.

2010s-2020s: Deep Learning & AI Everywhere

- **Deep Learning**: AI became better at **understanding images, speech, and text**.
- **AI in Daily Life**: Self-driving cars, smart assistants, and robots became common.

Conclusion

AI has **grown from simple programs to powerful tools** that help in daily life, science, and industry! 🚀

Technologies of Artificial Intelligence (AI)

AI is built on several key areas that help it work in different ways. These are called the **pillars of AI**.

---

1. Machine Learning (ML)

- AI learns from **data** and improves over time without needing extra coding.
- Used in **recommendations, fraud detection, and self-learning systems**.

2. Natural Language Processing (NLP)

- AI understands and **processes human language** like text and speech.
- Used in **chatbots, translation apps, and voice assistants**.

3. Robotics

- AI helps robots **think and act** like humans in different environments.
- Used in **factories, self-driving cars, and healthcare robots**.

4. Computer Vision

- AI sees and understands **images and videos** like humans do.
- Used in **face recognition, medical scans, and security cameras**.

5. Expert Systems

- AI acts like a **human expert** in specific areas.
- Used in **medical diagnosis, troubleshooting, and legal advice**.

6. Speech Recognition

- AI **listens** to human speech and converts it into text.
- Used in **voice assistants (Alexa, Siri) and voice-to-text apps**.

7. Planning and Scheduling

- AI **plans tasks and schedules** events automatically.
- Used in **route planning, delivery systems, and time management**.

8. Knowledge Representation

- AI stores and uses **knowledge** to make smart decisions.
- Used in **search engines, AI chatbots, and smart assistants**.

Conclusion

These **pillars** work together to make AI **smarter and more useful** in everyday life! 🚀

Agent

An **agent** in AI is something that **observes** what's happening around it and then **takes actions** to reach a goal. It uses **sensors** to collect information and **actuators** to perform actions. AI agents are used in many real-life applications, like self-driving cars, chatbots, and robots.

PEAS Framework

The **PEAS framework** helps us understand how an **AI agent** works by breaking it into four parts:

---

1. Performance Measure – How Do We Know the Agent is Doing Well?

- This tells us **what success means** for the AI agent.
- It could be **winning a game, driving safely, cleaning a room properly**, etc.
- Example: A self-driving car is successful if it **follows traffic rules, avoids accidents, and reaches the destination quickly**.

---

2. Environment – Where Does the Agent Work?

- This is the **place or situation** where the AI agent operates.
- The environment can be **a road, a chessboard, a chat application, or even a hospital**.
- Example: In chess, the environment is the **chessboard**; in a vacuum cleaner, it's the **floor of a house**.

---

3. Actuators – What Actions Can the Agent Take?

- Actuators are **the parts of the agent that perform actions** in the environment.
- These could be **robotic arms, wheels, a speaker, or even text responses**.
- Example: A robot vacuum **moves, sucks in dirt, and empties waste**; a chatbot **types messages** in response to a user.

---

4. Sensors – How Does the Agent Collect Information?

- Sensors help the agent **see, hear, or sense** what is happening around it.
- They help the AI make **better decisions** based on real-world data.
- Example: A self-driving car uses **cameras and radar** to detect traffic; a chatbot **reads user messages** to understand their needs.

Conclusion

The **PEAS framework** helps in designing AI agents by making sure they have **clear goals, a working environment, the ability to act, and ways to gather information**. 🚀

Examples of AI Agents Using PEAS Framework

1. Self-Driving Car Agent

- **Performance Measure**
  - Drive safely without accidents.
  - Follow traffic rules.
  - Take the shortest and fastest route.
- **Environment**
  - Roads, traffic signals, pedestrians, and other vehicles.
- **Actuators**
  - Steering, brakes, accelerator, indicator lights.
- **Sensors**
  - Cameras, GPS, speed sensors, radar.

---

2. Chatbot Agent

- **Performance Measure**
  - Understands user queries.
  - Gives correct and helpful responses.
  - Responds quickly.
- **Environment**
  - User chats/messages, internet for searching answers.
- **Actuators**
  - Displays text responses, suggests links or solutions.
- **Sensors**
  - Keyboard input, voice input (if voice-based).

---

3. Vacuum Cleaner Agent

- **Performance Measure**
  - Cleans dust efficiently.
  - Covers all areas of the room.
  - Uses battery wisely.
- **Environment**
  - Floors, furniture, walls, obstacles like toys or shoes.
- **Actuators**
  - Wheels for movement, suction motor to collect dust.
- **Sensors**
  - Dirt sensors, obstacle detectors, battery level sensors.

---

4. Water-Jug Problem Agent

- **Performance Measure**
  - Measure the exact amount of water required.
  - Use the least number of steps (fill, empty, pour).
  - Minimize water wastage.
- **Environment**
  - Two jugs of fixed capacities.
  - A water source for filling jugs.
  - A sink or ground for discarding extra water.
- **Actuators**
  - Fill a jug with water.
  - Empty a jug.
  - Pour water from one jug to another.
- **Sensors**
  - Level indicator to check how much water is in the jugs.
  - Goal checker to verify if the required water amount is reached.

Conclusion

AI agents work by sensing their environment and taking actions to achieve their goals. The **PEAS** framework helps define how they function in different situations, such as **self-driving cars, chatbots, vacuum cleaners, and even solving the water-jug problem**. This makes it easier to design smart systems that can perform tasks efficiently.

Search Space

Search Space refers to the **set of all possible states** that an AI system can explore to find a solution to a given problem. It is a fundamental concept in problem-solving and search algorithms.

In simple terms, Search Space refers to all the possible solutions or answers to a given problem. It includes every potential option that can be considered before finding the best or correct solution.

Key Concepts of Search Space:

1. **State Space Representation**:
   - The search space consists of all possible configurations (states) that an AI agent can be in while solving a problem.
   - Each state represents a possible intermediate or final solution.
2. **Search Graph & Search Tree**:
   - A **Search Graph** is a graph where nodes represent states, and edges represent possible transitions between them.
   - A **Search Tree** is a tree structure where each node is expanded based on possible actions.
3. **Problem Formulation in Search Space**:
   - **Initial State**: The starting point of the problem.
   - **Goal State**: The desired solution or objective.
   - **Operators (Actions)**: Allowed moves to transition between states.

- **Path Cost**: The cost associated with moving from one state to another.

State Space Progression in Games:

In any game, the state space represents the total number of possible positions or situations that can occur. At the beginning, the state space is vast because multiple moves and choices are available. However, as the game progresses, decisions are made, and certain moves become unavailable, gradually reducing the number of possible states. This natural reduction continues until the game reaches its final stages, where only a few or no moves remain. Therefore, the state space consistently decreases over time, narrowing down to a definitive outcome as the game concludes.

Search Space in Games

The **search space** in a game refers to all the possible moves and game states that can be reached from the starting position. The **larger the search space**, the more complex the game is for an AI to solve. Let's explore search spaces in different games.

---

1. Search Space in Chess

- **What it means**: Chess has a **huge** search space because there are **millions of possible moves** after just a few turns.
- **Search Tree Example**:
  - The initial board is the **root** (starting state).
  - Every legal move creates a **new state** (branches).
  - The AI must check many moves ahead to decide the best one.
- **Search Space Size**: **Extremely Large** (~10^120 possible game states).
- **Example**:
  - If the AI looks **two moves ahead**, there could be **hundreds of possibilities**.
  - Looking **four moves ahead**, the number increases **exponentially**.

---

2. Search Space in Ludo

- **What it means**: Ludo's search space is **randomized** because of the **dice rolls**.
- **Search Tree Example**:
  - The board starts with four pieces at home.
  - Each dice roll decides how far a piece moves.
  - There are **many possible ways** to move pieces based on dice outcomes.
- **Search Space Size**: **Very Large** (due to randomness).
- **Example**:
  - If you roll a **6**, you can move a piece out or move an existing piece forward.
  - The AI must plan for **future dice rolls and opponent moves**.

---

3. Search Space in Tic-Tac-Toe

- **What it means**: Tic-Tac-Toe has a **small and simple** search space compared to Chess and Ludo.
- **Search Tree Example**:
  - The board starts empty.
  - Each turn, a player places "X" or "O" in an empty spot.
  - There are only **9 positions** to fill, so the search space is limited.
- **Search Space Size**: **Small** (~26,830 possible games).
- **Example**:
  - The first player places "X" in one of **9 spaces**.
  - The second player places "O" in one of the **remaining 8 spaces**.
  - The game tree grows, but **ends quickly** compared to Chess.
