UNIT-1 Artificial Intelligence

Definition of AI: Artificial Intelligence (AI) is the simulation of human intelligence in machines that are designed to think and learn like humans. These systems can perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation**.**

Importance of AI in modern technology

AI is now integrated into various industries, enhancing capabilities in areas like healthcare, finance, transportation, and customer service. Its ability to analyze vast amounts of data and make decisions is transforming how businesses operate.

First work in AI

The first work that is now generally recognized as AI was done by Warren McCulloh and Walter Pits in 1943. They drew on three sources:

o **Knowledge of basic Physiology and Functions of Neurons in brain.**
o **A formal analysis of propositional logic**
o **Turing's Theory of Computation**

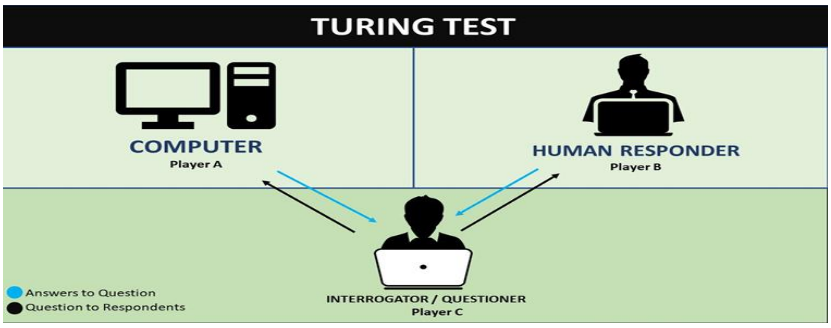
Key Findings

- They proposed a model of artificial neurons in which each neuron can be characterized asbeing **on** and **off**.
- They showed that any computable function could be computed by some network ofconnected neurons.
- They also suggested that suitably defined networks can also learn.

History of AI

AI, or Artificial Intelligence, has a rich history spanning over several decades. Below are the key milestones in the development of AI:

- **Early Foundations (1940s-1950s)**:
  - *Alan Turing (1950)*: Turing proposed the idea of a "universal machine" and introduced the famous Turing Test to determine a machine's ability to exhibit intelligent behaviour indistinguishable from a human.

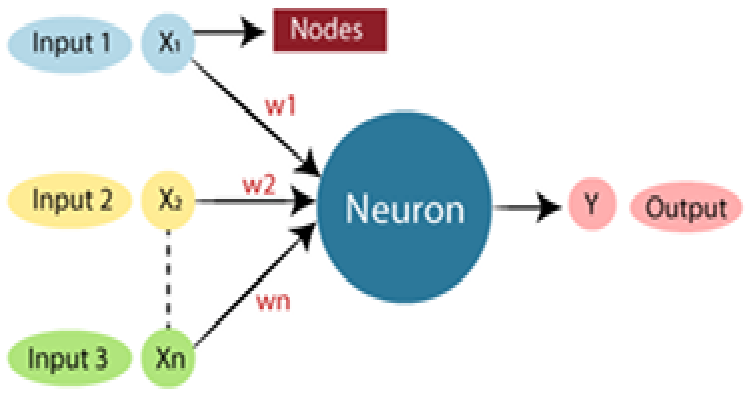


- *John von Neumann*: His contributions to the architecture of modern computers laid the groundwork for the development of AI.

- **The First AI Programs (1950s)**:
  - *Logic Theorist (1956)*: The first AI program, developed by Allen Newell and Herbert Simon, was able to prove mathematical theorems.
  - *Dartmouth Conference (1956)*: This conference is considered the official birth of AI. Researchers such as John McCarthy(***Father of AI***), Marvin Minsky, and others participated and discussed how to make machines think like humans.
- **The Rise and Fall of AI (1960s-1970s)**:
  - *Symbolic AI and Expert Systems*: AI research initially focused on symbolic reasoning and expert systems, where machines were programmed with vast knowledge in specific domains (e.g., DENDRAL, MYCIN).
  - *AI Winter*: Several obstacles, such as limited computational power, caused funding to dry up and led to a period of reduced interest known as the AI Winter.
- **AI Resurgence (1980s-1990s)**:
  - *Neural Networks*: In the 1980s, AI research saw a resurgence with the development of backpropagation and the improvement of neural networks.
  - *Machine Learning*: By the 1990s, AI shifted from rule-based systems to data-driven approaches like machine learning.
- **Modern AI (2000s-Present)**:
  - *Deep Learning*: The rise of deep learning and advanced neural networks in the 2010s (e.g., Google's AlphaGo) brought AI back into the limelight.
  - *AI Applications*: Today, AI is applied across diverse fields like healthcare, autonomous vehicles, natural language processing (e.g., GPT models), and more.

Milestones for AI after 1980s

- **Neural Networks and Backpropagation (1986)**:
  - Geoffrey Hinton and others revived interest in AI with the re-discovery of neural networks and the development of the backpropagation algorithm, allowing multi-layered networks (deep learning) to be trained efficiently.



- **Lisp Machines (1980s)**:
  - Specialized computers designed to run AI programs using the Lisp programming language were developed. However, they were eventually overtaken by general-purpose computers.
- **Resurgence of AI Research**:
  - In the late 1980s, AI began gaining traction again, especially with the application of probabilistic reasoning, neural networks, and data-driven learning methods, shifting away from rule-based systems.

---

6. Breakthrough in Games and Algorithms (1990s)

- **Deep Blue (1997)**:
  - Developed by IBM, Deep Blue became the first computer system to defeat a reigning world chess champion, Garry Kasparov, in a six-game match. This milestone demonstrated AI's ability to tackle complex strategic games.
- **Support Vector Machines (1990s)**:
  - Support vector machines (SVM) and other machine learning algorithms were developed and applied to various classification problems, marking a shift toward data-driven AI approaches.

---

7. The Advent of Modern AI (2000s)

- **AI in Robotics**:
  - AI became integrated with robotics, leading to advances in autonomous systems. Examples include the Mars Rover missions, where AI helped in navigation and decision-making in uncharted terrains.
- **Natural Language Processing (NLP)**:
  - Significant improvements were made in NLP, enabling more effective machine translation, speech recognition (e.g., Google Translate, Siri), and text understanding.
- **AI in Real-World Applications**:
  - AI began being used for commercial applications such as recommendation systems (Amazon, Netflix), search engines (Google), and autonomous systems like self-driving cars.

---

8. The Rise of Deep Learning (2010s-Present)

- **DeepMind's AlphaGo (2016)**:
  - AlphaGo, a deep learning-based AI, defeated the world champion Go player, Lee Sedol. Go was considered a much more complex game than chess due to its vast number of possible moves, highlighting the power of deep neural networks and reinforcement learning.
- **AI in Image Recognition and Speech**:
  - The ImageNet Challenge (ILSVRC) became a key platform where AI systems, particularly deep convolutional neural networks (CNNs), dramatically improved image recognition tasks.
- **Reinforcement Learning**:
  - Reinforcement learning algorithms like those used in AlphaGo showcased the ability of AI to learn from trial and error in complex environments, further enhancing AI's learning capabilities.

Artificial Intelligence (AI) Classification

Based on capabilities and functionalities:

- **Reactive Machines:** These are the most basic type of AI. They can react to specific situations or inputs but don't have memory or past experience to influence their decisions. Example: IBM's Deep Blue, the chess-playing AI.
- **Limited Memory:** These AI systems can use past experiences or historical data to make current decisions. They have a temporary or limited memory to store past information and predictions. Example: Self-driving cars.
- **Theory of Mind:** This is a more advanced type of AI that researchers are still working to develop. These systems would understand emotions, people, beliefs, and be able to interact socially.Example: More advanced personal assistants that can understand human emotions and react accordingly.

Self-aware AI: This is the most advanced form of AI, which is still theoretical and not yet realized. These systems would have their own consciousness, self-awareness, and emotions. They would be smarter and more capable than the human mind.

Agents and Environments in Artificial Intelligence (AI)

1. What is an Agent in AI?

- **Definition**: An *agent* in AI is any entity (software or hardware) that can perceive its environment through sensors and act upon that environment through actuators to achieve a set of goals.
- **Characteristics of an AI Agent**:
  - **Perception**: The agent receives input from its environment via sensors.
  - **Action**: The agent responds to the environment using actuators.

    - **Autonomy**: The agent operates without human intervention, making its own decisions based on its perception and goals.
    - **Goal-Driven**: The agent is designed to achieve certain objectives.

---

2. Structure of an Agent

AI agents are structured into four key components:

- **Sensors**: Used by the agent to gather information about the environment. In a software system, sensors could be data inputs like user commands or environmental data.
- **Actuators**: The components through which an agent performs actions. For a robot, these could be motors controlling movement, while in software, actuators might represent system commands or responses.
- **Perception**: The agent's ability to interpret sensor data and create an internal representation of the environment.
- **Decision-Making Process**: The core intelligence of the agent, where it decides what actions to take to achieve its goals. This decision process may involve:
    - Rule-based logic.
    - Search algorithms.
    - Machine learning models.

---

3. Types of Agents

- **Simple Reflex Agents**:
    - These agents act based solely on the current perception, without considering history or past experiences.
    - They follow a condition-action rule (if condition, then action).
    - Example: A thermostat that turns the heater on if the temperature drops below a set point.
- **Model-Based Reflex Agents**:
    - These agents maintain an internal state that represents aspects of the world that are not immediately visible.
    - They use a model of the environment to make decisions, considering both the current perception and their internal state.
    - Example: A vacuum robot that keeps track of which areas have already been cleaned.
- **Goal-Based Agents**:
    - These agents take actions to achieve specific goals. They have a clear objective and make decisions based on what actions will bring them closer to achieving that goal.
    - Example: A navigation system that plans the shortest route to a destination.
- **Utility-Based Agents**:
    - Utility-based agents aim not just to achieve goals but to optimize the achievement of these goals based on a utility function (a measure of success or happiness).

    - Example: A self-driving car that not only reaches the destination but does so while minimizing fuel consumption or travel time.
- **Learning Agents**:
    - These agents can improve their performance over time based on past experiences. They learn from data and adjust their behavior accordingly.
    - Example: A recommendation system that improves its suggestions by learning from user interactions.

---

4. What is an Environment in AI?

- **Definition**: The *environment* refers to everything outside the agent that it interacts with. This includes any external system, surroundings, or input that can affect the agent's decisions and actions.
- **Perception of the Environment**:
    - Agents perceive the environment through their sensors. This perception may be complete (full knowledge of the state of the environment) or incomplete (partial knowledge).
- **Environment's Influence**:
    - The environment provides feedback to the agent based on its actions, and the agent must adapt its behavior based on changes or reactions from the environment.

---

5. Types of Environments

- **Fully Observable vs. Partially Observable**:
    - *Fully Observable*: The agent has complete access to the relevant information about the environment at all times. Example: Chess, where the board's state is always fully visible.
    - *Partially Observable*: The agent has limited access to the environment, and some parts may be hidden. Example: Poker, where the opponent's cards are not visible.
- **Deterministic vs. Stochastic**:
    - *Deterministic*: The environment's outcome is predictable and depends only on the current state and the agent's actions. Example: A board game like checkers.
    - *Stochastic*: The environment's outcome includes randomness and uncertainty. Example: Autonomous driving where external factors (e.g., weather, human drivers) may introduce unpredictability.
- **Static vs. Dynamic**:
    - *Static*: The environment does not change while the agent is making decisions. Example: Solving a puzzle where the configuration doesn't change over time.
    - *Dynamic*: The environment evolves over time, even while the agent is deliberating. Example: A robot navigating a changing environment, such as a factory floor with moving obstacles.
- **Discrete vs. Continuous**:

- *Discrete*: The environment consists of a limited number of distinct states or actions. Example: A board game where players move between specific squares.
    - *Continuous*: The environment allows for an infinite number of possible states or actions. Example: A self-driving car that must navigate in a world where speed and direction are continuous.
- **Episodic vs. Sequential**:
    - *Episodic*: The agent's experience is divided into discrete episodes, and the outcome of one episode does not affect the next. Example: Image classification, where each image is processed independently.
    - *Sequential*: The current decision affects future decisions, creating a sequence of actions that must be planned together. Example: A chess game, where each move affects future gameplay.
- **Known vs. Unknown**:
    - *Known*: The agent understands the rules and structure of the environment, which allows for better decision-making. Example: Chess, where all rules are known in advance.
    - *Unknown*: The agent must learn the rules and dynamics of the environment through exploration and interaction. Example: A reinforcement learning agent in an unfamiliar video game.

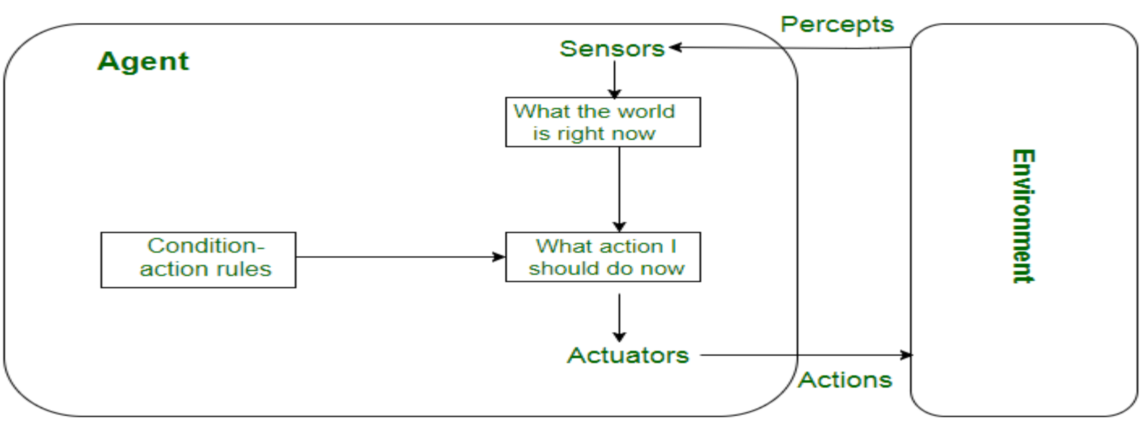
---

6. Agent-Environment Interaction

The interaction between agents and environments is often modeled as a feedback loop:

1. **Perception**: The agent perceives the state of the environment through sensors.
2. **Decision**: Based on its perception and internal model, the agent decides on an action.
3. **Action**: The agent performs the chosen action in the environment using its actuators.
4. **Feedback**: The environment responds to the agent's actions, potentially changing the state of the environment.

This loop continues as long as the agent is operational, allowing it to continuously adjust its behavior to achieve its goals.



---

7. Performance Measures for Agents

Agents are typically evaluated based on how well they perform in the environment. Performance measures may include:

- **Goal Achievement**: Did the agent successfully accomplish its objective?
- **Efficiency**: How quickly or resource-efficiently did the agent achieve its goal?
- **Adaptability**: Can the agent adapt to changes in the environment?
- **Learning**: Can the agent improve its performance over time?

---

8. Examples of Agents and Environments

- **Robotic Vacuum Cleaner (e.g., Roomba)**:
  - *Agent*: The vacuum cleaner itself.
  - *Environment*: The room it cleans, including obstacles like furniture and walls.
  - *Actions*: Moving, cleaning, avoiding obstacles.
  - *Sensors*: Bump sensors, cameras, or infrared sensors to perceive obstacles.
- **Autonomous Car**:
  - *Agent*: The self-driving system.
  - *Environment*: The road, traffic, pedestrians, and weather conditions.
  - *Actions*: Steering, braking, accelerating.
  - *Sensors*: Cameras, LIDAR, GPS, radar.
- **Chess AI (e.g., Deep Blue)**:
  - *Agent*: The chess-playing program.
  - *Environment*: The chessboard and the opponent’s moves.
  - *Actions*: Moving pieces on the board.
  - *Sensors*: Input of the current board configuration.

---

2. Problems of AI

AI faces a variety of challenges and problems, including:

- **Knowledge Representation**: How to represent knowledge about the world in a form that a machine can use to solve complex problems.
- **Learning and Adaptation**: Ensuring AI systems can learn from data and adapt to new environments or situations.
- **Common Sense Reasoning**: Building AI that can understand and reason like humans with common sense.
- **Uncertainty Handling**: Dealing with incomplete or noisy information while making decisions.
- **Ethical Concerns**: Issues around AI safety, job displacement, bias in algorithms, privacy concerns, and the implications of AI decision-making.

- **Natural Language Understanding**: Machines still struggle to understand and generate human language as naturally as humans do.
- **Scalability**: Many AI algorithms do not scale well with large datasets or complex tasks.

---

3. AI Techniques

AI employs several techniques to address various tasks:

- **Search Algorithms**: AI uses techniques like breadth-first search (BFS), depth-first search (DFS), A* algorithm to navigate problem spaces.
- **Knowledge Representation**: Methods such as semantic networks, frames, logic (propositional and first-order), and ontologies are used to represent information.
- **Machine Learning**: Includes supervised, unsupervised, and reinforcement learning. Some common models are decision trees, neural networks, and support vector machines.
- **Natural Language Processing (NLP)**: Techniques for processing and understanding human language, such as parsing, tokenization, and sentiment analysis.
- **Heuristic Methods**: Heuristics are rules of thumb that help narrow down the search for solutions (e.g., using Manhattan distance in puzzles like the 8-puzzle).
- **Metaheuristic Approaches**: Techniques like genetic algorithms, particle swarm optimization, and ant colony optimization are used to solve optimization problems.

---

4. Problem Solving in AI

Problem-solving in AI involves:

1. **Identifying the Problem**: First, a problem must be defined clearly.
2. **Formulating a Plan**: Developing strategies to tackle the problem (e.g., using algorithms, heuristics, or machine learning models).
3. **Executing the Plan**: Implementing the solution and continuously adjusting based on feedback.

---

5. Problem Space and Search

- **Problem Space**: This refers to the complete set of possible states and actions that can be taken to solve a given problem.
- **Search**: AI techniques explore the problem space to find solutions. Types of search methods include:
    - *Uninformed Search*: Algorithms like BFS, DFS, where no additional information is used.
    - *Informed Search*: Heuristic-based algorithms like A* that use extra knowledge to guide the search more efficiently.

---

6. Defining the Problem as State Space Search

- **State Space**: The collection of all possible configurations of a problem.
- **Initial State**: The starting configuration.
- **Goal State**: The desired configuration or solution.
- **Operators**: Actions that move from one state to another.
- **Search Algorithms**: The mechanism AI uses to explore the state space, such as BFS, DFS, or A*.

---

7. Problem Characteristics

Characteristics of AI problems include:

- **Solvability**: Whether the problem has a defined solution.
- **Complexity**: How challenging it is in terms of computational effort.
- **Static vs. Dynamic**: Static problems have fixed parameters, whereas dynamic ones evolve over time.
- **Deterministic vs. Non-deterministic**: Deterministic problems have predictable outcomes; non-deterministic ones involve uncertainty.
- **Decomposability**: Can the problem be broken into smaller sub-problems?

---

8. Tic-Tac-Toe Problem

- **Description**: Tic-Tac-Toe is a simple game involving a 3x3 grid where two players (X and O) take turns marking the cells. The objective is to get three of their marks in a row (vertically, horizontally, or diagonally).
- **State Space**: Each unique configuration of X's and O's on the grid represents a state. There are a finite number of possible game configurations (3^9 = 19,683 states).
- **Solution Strategy**: AI can solve this using minimax search, which evaluates the optimal move by considering all possible future states and backtracking from the goal.

  Problem formulation for Tic-Tac-Toe

- **Initial State:** An empty 3x3 grid, with the first player ready to move.
- **Actions:** The possible placements of 'X' or 'O' in any empty cell.
- **Result:** The new board state updated with cell selection after a player makes a move.
- **Goal Test:** A player wins by placing three marks in a row (horizontally, vertically, or diagonally), or the game results in a draw if the board is full with no winner.
- **Path Cost:** Uniform, typically 1 per move, with the total cost reflecting the number of moves to reach a terminal state (win, loss, or draw).

---

9. 8-Puzzle Problem

- **Description**: The 8-puzzle is a 3x3 grid puzzle with 8 numbered tiles and one empty space. The goal is to rearrange the tiles to achieve a specific goal configuration.
- **State Space**: Every arrangement of the tiles is a unique state. The total number of states for an 8-puzzle is 9! (362,880 possible configurations).
- **Solution Strategies**:
    - *Breadth-First Search (BFS)*: Explores all possible moves layer by layer but can be computationally expensive.
    - *A* Algorithm*: Uses heuristics like Manhattan distance to guide the search more efficiently.
    - *Heuristics*: The number of misplaced tiles or total distance of tiles from their goal position is often used to guide the search.

8-Puzzle Problem Formulation:

- **Initial State** :A specific scrambled configuration of the 8 tiles and a blank space
- **State Space:** All possible configurations of the 8 tiles and blank space.
- **Actions:** Moving a tile adjacent to the blank space into the blank space.
- **Transition Model:** The result of moving a tile into the blank space, leading to a new state.
- **Goal Test:** Checking whether the current configuration matches the goal configuration.
- **Path Cost:** Typically, each move has a uniform cost of 1; the goal is to minimize the total number of moves.
- **Terminal States:** The state where the puzzle is solved and matches the goal configuration.
- **Optimality Criteria:** Achieving the goal state in the fewest number of moves.

AI programming languages

Lisp is a programming language that's known for its use of lists as a fundamental data type and its role in artificial intelligence (AI) and symbolic computing. Here are some things to know about Lisp:

History: John McCarthy designed Lisp in 1958, making it one of the oldest programming languages still in use today.

Name: Lisp stands for "List Processing".

Notation: Lisp has a distinctive prefix notation that uses parentheses.

Influential: Lisp has influenced the development of many other programming languages.

Versions: Some well-known versions of Lisp include Common Lisp, Scheme, and Clojure. Logo: Logo is a version of Lisp that was created for children.

Functional programming: Lisp is a functional programming language that's known for its ability to manipulate data strings.

AI: Lisp is widely used in AI applications.

Ex-

(defun square (x)

(* x x))

(square 4) ; This will return 16.

Prolog (PROgramming in LOGic) is a logic programming language commonly used for tasks involving symbolic reasoning and knowledge representation. The syntax of Prolog is very different from traditional procedural programming languages.

Basic Prolog Syntax:

- **Facts** define unconditional truths.
- **Rules** define relationships based on facts or other rules.
- **Queries** are used to ask questions, and Prolog uses logic to deduce answers.
- **Variables** (like `X` and `Y`) are placeholders that Prolog will try to fill in when answering queries.

Ex- % Rule to calculate the area of a rectangle

area_of_rectangle(Length, Width, Area) :-

Area is Length * Width.

% **Query for** area_of_rectangle with values{5,10}

?- area_of_rectangle(5, 10, Area).

Area = 50.

Explanation: `Length` and `Width` are inputs.

The rule `Area is Length * Width` calculates the area by multiplying the two values.

`Area` is the output.

Q1.-Solve for initial state

```
1 2 3        1 6 3          1 3 2
4 6 7        2 4 5          3 4 5
  5 8        7   8          7 8
```

Find the path cost (No. moves)

Q-2 **Write PEAS for agents for:**

Self- Driving Car, Automated Vacuum Cleaner, Chess-Playing Agent, Spam Email Filter