ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEM

UNIT-1

Intelligence:

Intelligence is defined as mental capability that involves the ability to reason, to plan, to solve problems, to think abstractly, to comprehend complex ideas, to learn quickly and to learn from experience. It is not merely booking learning, a narrow academic skill, or test-taking smartness.

In simple words, intelligence is nothing but thinking skills and the ability to adapt to and to learn from life's everyday experiences.

Concepts:

The concept of intelligence is complex and has been defined in various ways by different scholars. Intelligence can be understood as the ability to learn, understand, and apply knowledge and skills to solve problems and adapt to new situations. It involves cognitive abilities such as reasoning, problem-solving, perception, and memory.

Howard Gardner introduced the theory of multiple intelligences, which suggested that intelligence is not a single entity but rather a combination of different abilities. Gardner identified eight different types of intelligence, including linguistic, logical-mathematical, spatial, bodily-kinesthetic, musical, interpersonal, intrapersonal, and naturalist.

Sternberg's triarchic theory of intelligence focuses on three main aspects of intelligence: analytical, creative, and practical. This theory emphasizes the importance of adapting to the environment, using problem-solving strategies, and applying knowledge to practical situations.

Emotional intelligence is another aspect of intelligence that has gained importance in recent years. It involves the ability to recognize and manage one's own emotions and those of others. This ability is essential for interpersonal relationships and has been linked to success in various areas of life.

Intelligence testing has been widely used to measure intelligence, with IQ tests being the most popular. However, these tests have been criticized for their cultural bias and limited scope in measuring intelligence.

Artificial Intelligence:

According to the father of Artificial Intelligence John McCarthy, it is "The science and engineering of making intelligent machines, especially intelligent computer programs".

Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think.

Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans. Some of the activities computers with artificial intelligence are designed for include:

- Speech recognition
- Learning
- Planning
- Problem solving

Turing Test:

Alan Turing's 1950 article Computing Machinery and Intelligence discussed conditions for considering a machine to be intelligent. He argued that if the machine could successfully pretend to be human to a knowledgeable observer, then you certainly should consider it intelligent. This test would satisfy most people but not all philosophers.

The observer could interact with the machine and a human by teletype (to avoid requiring that the machine imitate the appearance or voice of the person), and the human would try to persuade the observer that it was human, and the machine would try to fool the observer.

The Turing test is a one-sided test. A machine that passes the test should certainly be considered intelligent, but a machine could still be considered intelligent without knowing enough about humans to imitate a human.

Applications of AI:

1. Gaming – AI plays crucial role in strategic games such as chess, poker, tic-tac-toe, etc., where machine can think of large number of possible positions based on heuristic knowledge.

2. Natural Language Processing – It is possible to interact with the computer that understands natural language spoken by humans.

3. Expert Systems – There are some applications which integrate machine, software, and special information to impart reasoning and advising. They provide explanation and advice to the users.

4. Vision Systems – These systems understand, interpret, and comprehend visual input on the computer. For example,

- Doctors use clinical expert system to diagnose the patient.
- Police use computer software that can recognize the face of criminal with the stored portrait made by forensic artist.

5. Speech Recognition – Some intelligent systems are capable of hearing and comprehending the language in terms of sentences and their meanings while a human talks to it. It can handle different accents, slang words, noise in the background, changes in human noise due to cold, etc.

6. Handwriting Recognition – The handwriting recognition software reads text written on paper by a pen or on screen by a stylus. It can recognize the shapes of the letters and convert it into editable text.

7. Intelligent Robots – Robots are able to perform the tasks given by a human. They have sensors to detect physical data from the real world such as light, heat, temperature, movement, sound, bump, and pressure. They have efficient processors, multiple sensors and huge memory, to exhibit intelligence. In addition, they are capable of learning from their mistakes, and they can adapt to the new environment.

Search Techniques:

Search techniques in AI refer to the methods and algorithms that are used to explore the possible solutions to a problem or goal. These techniques help to efficiently navigate through the space of possible solutions to find an optimal or near-optimal solution. Here are some common search techniques used in AI:

1. **Breadth-First Search (BFS):** BFS is a simple search algorithm that explores all the nodes at a given depth level before moving on to the next depth level. It uses a queue data structure to keep track of the nodes to be visited next.
2. **Depth-First Search (DFS)**: DFS is a search algorithm that explores as far as possible along each branch before backtracking. It uses a stack data structure to keep track of the nodes to be visited next.
3. **A* Search:** A* is a heuristic search algorithm that combines the best of both BFS and DFS. It uses a heuristic function to estimate the distance to the goal, and it explores the most promising path first. It uses a priority queue data structure to keep track of the nodes to be visited next.
4. **Greedy Best-First Search:** This algorithm is like A* but does not consider the cost of the path from the start node to the current node. It only considers the heuristic function to select the most promising path.
5. **Hill Climbing Search:** Hill Climbing is a local search algorithm that starts from a random solution and repeatedly selects the neighboring solution with the highest value until no better neighbor can be found. It is used for optimization problems.
6. **Genetic Algorithms:** Genetic Algorithms are a metaheuristic that mimics the process of natural selection to find a good solution to a problem. It uses a population of potential solutions that evolve over time through crossover and mutation operations.

State Spaces:

State space is a key concept in artificial intelligence that refers to the set of all possible states that a system can be in. In AI, state space is used to represent and solve complex problems through a process called state space search. This process involves exploring the state space to find a solution that meets certain criteria.

The concept of state space is used in a variety of AI applications, including game playing, problem-solving, planning, and robotics. For example, in game playing, state space is used to represent the current position of the game and all possible moves that can be made. In problem-solving, state space is used to represent the current state of the problem and the possible actions that can be taken to solve it.

State space is typically represented as a graph or tree, where nodes represent states and edges represent transitions between states. The initial state is represented by the root node, and the goal state is represented by one or more leaf nodes. The search process starts from the root node and explores the graph or tree until a solution is found.

State space search algorithms are used to explore the state space and find a solution. These algorithms can be categorized into two main types: uninformed search and informed search. Uninformed search algorithms, such as breadth-first search and depth-first search, explore the state space without any knowledge of the problem structure. Informed search algorithms, such as A* search and iterative deepening A*, use heuristic functions to guide the search towards the goal state.

In addition to search algorithms, state space can also be used to represent the knowledge of an AI system. In this case, the state space represents the set of all possible facts and their relationships. The AI system can then reason about the state space to make decisions or solve problems.

Problem Characteristics:

1. **Complexity:** AI problems are usually complex and involve many variables and data. This complexity is due to the vast amount of information that AI systems need to process and analyze to make accurate predictions or decisions.
2. **Uncertainty:** AI problems are often characterized by uncertainty, as the input data may be incomplete, noisy, or even contradictory. AI systems must be able to handle uncertainty by considering multiple possibilities and probabilities.
3. **Dynamic nature:** AI problems are often dynamic and constantly changing. The input data and the environment in which the AI system operates may change over time, requiring the AI system to adapt and learn continuously.
4. **Non-linearity:** AI problems may involve non-linear relationships between the input data and the output or decisions made by the AI system. These non-linearities can make it difficult to predict the behavior of the AI system accurately.

5. **Scalability:** AI problems can be scalable, meaning that they may need to handle large amounts of data and perform complex computations efficiently. Scalability is crucial for AI systems to perform well in real-world applications.
6. **Interpretability:** AI systems may produce outputs that are difficult for humans to interpret, making it challenging to understand why a particular decision was made. Interpretability is an essential consideration for AI systems in applications where transparency is required.
7. **Ethics:** AI systems may have ethical implications, such as bias, discrimination, or unintended consequences. These ethical considerations must be carefully evaluated to ensure that the AI system is beneficial and does not harm individuals or society.

Production Rules:

Production rules, also known as production systems or production rule systems, are a type of formalism used in artificial intelligence for knowledge representation and reasoning. They consist of a set of if-then rules that define the conditions under which certain actions or behaviors should be performed.

Each rule in a production system has two parts: a condition (if) and an action (then). The condition specifies the set of circumstances or facts that must be present for the rule to be applied, while the action specifies the task or behavior that should be executed if the condition is met.

Production rules are often used in expert systems, which are AI systems designed to mimic the decision-making abilities of human experts in a specific domain. By encoding the rules and heuristics that human experts use to solve problems, expert systems can provide intelligent advice and guidance to users in a wide range of fields, from medicine and law to finance and engineering.

Some key benefits of production rules in AI include their modularity, which allows for easy modification and updates, and their ability to handle uncertain and incomplete information. However, production systems can also suffer from scalability and efficiency issues, particularly when dealing with large and complex rule sets.

Production system characteristic:

A production system is a type of computer program that is used in artificial intelligence (AI) for automated reasoning and decision making. It is designed to process a set of rules and data, analyze them, and then generate a set of actions or recommendations based on the information. The following are the main characteristics of a production system:

1. **Rule-Based:** A production system is built on a set of rules that govern its behavior. These rules are typically expressed in the form of if-then statements that define the conditions under which certain actions should be taken.

2. **Modular:** A production system is made up of several independent modules, each with its own set of rules and data. These modules work together to form a complete system that can process information and make decisions.
3. **Knowledge-Based:** A production system relies on a knowledge base that contains information about the problem domain it is designed to solve. This knowledge is used by the system to make decisions and generate recommendations.
4. **Inference Engine:** A production system uses an inference engine to process the rules and data in the knowledge base. The inference engine applies the rules to the data and generates conclusions based on the information.
5. **Goal-Oriented:** A production system is designed to achieve a specific goal or set of goals. It works towards this goal by processing the rules and data in the knowledge base and generating actions or recommendations that help achieve the goal.
6. **Reactive:** A production system is reactive in nature, which means it responds to changes in the environment or problem domain. It continuously monitors the input data and generates responses or actions based on the information it receives.
7. **Procedural:** A production system is procedural in nature, which means it follows a predefined set of steps or procedures to achieve its goals. These procedures are defined in the rules and data in the knowledge base.

Depth-First Search (DFS):

Depth-First Search (DFS) is a popular algorithm used in Artificial Intelligence (AI) for searching through large and complex data structures or graphs. It is a type of search algorithm that starts at the root node of a tree or graph, and explores as far as possible along each branch before backtracking. The following are the main characteristics of the DFS algorithm:

1. **Stack-Based:** DFS is a stack-based algorithm that uses a stack data structure to keep track of the nodes to be explored. It starts by pushing the root node onto the stack, and then pops off the top node to explore its children.
2. **Recursive:** DFS can be implemented using recursive functions. The function calls itself for each child node and continues until all nodes have been visited.
3. **Depth-First:** DFS explores the depth of the tree or graph first, which means it goes as far as possible along each branch before backtracking.
4. **Uninformed:** DFS is an uninformed search algorithm, which means it has no information about the location of the goal node. It systematically searches through all possible paths until it finds the goal.
5. **Memory Intensive:** DFS can be memory-intensive, especially in large and complex data structures. It can explore many branches and generate a large stack, which can lead to memory overflow.
6. **Non-Optimal:** DFS may not always find the optimal path to the goal. It may find a solution quickly, but it may not be the best solution.

Breadth First Search (BFS):

Breadth First Search (BFS) is a searching algorithm used to search for a node in a tree or graph data structure. It traverses the structure in a breadthwise manner, i.e., it explores all the nodes at a given level before moving to the next level. BFS starts with the root node and visits all the nodes at the same depth before proceeding to the next level.

BFS uses a queue data structure to maintain the order of the nodes to be visited. The algorithm starts with the root node and enqueues it into the queue. It then dequeues the node and explores its children by enqueuing them into the queue. The algorithm continues this process until it finds the desired node or the queue becomes empty.

The steps involved in BFS are as follows:

1. Enqueue the starting node into the queue.
2. Mark the starting node as visited.
3. While the queue is not empty, do the following:
4. Dequeue the node at the front of the queue.
5. Check if the node is the desired node. If it is, return it.
6. Otherwise, enqueue all its unvisited children into the queue and mark them as visited.
7. Repeat steps 4-6 until the queue is empty.

BFS is complete, i.e., it is guaranteed to find a solution if one exists, as long as the search space is finite. However, BFS may not be optimal as it may explore many nodes before finding the solution.

BFS is used in a variety of applications in AI, including route finding, puzzle solving, and game playing. It is also used in machine learning algorithms such as decision trees and random forests.