Intelligent System

Architecture of learning agent. [V Imp]

Learning Agent :
A learning agent in AI is the type of agent that can learn from its past experiences or it has learning capabilities. It starts to act with basic knowledge and then is able to act and adapt automatically through learning.

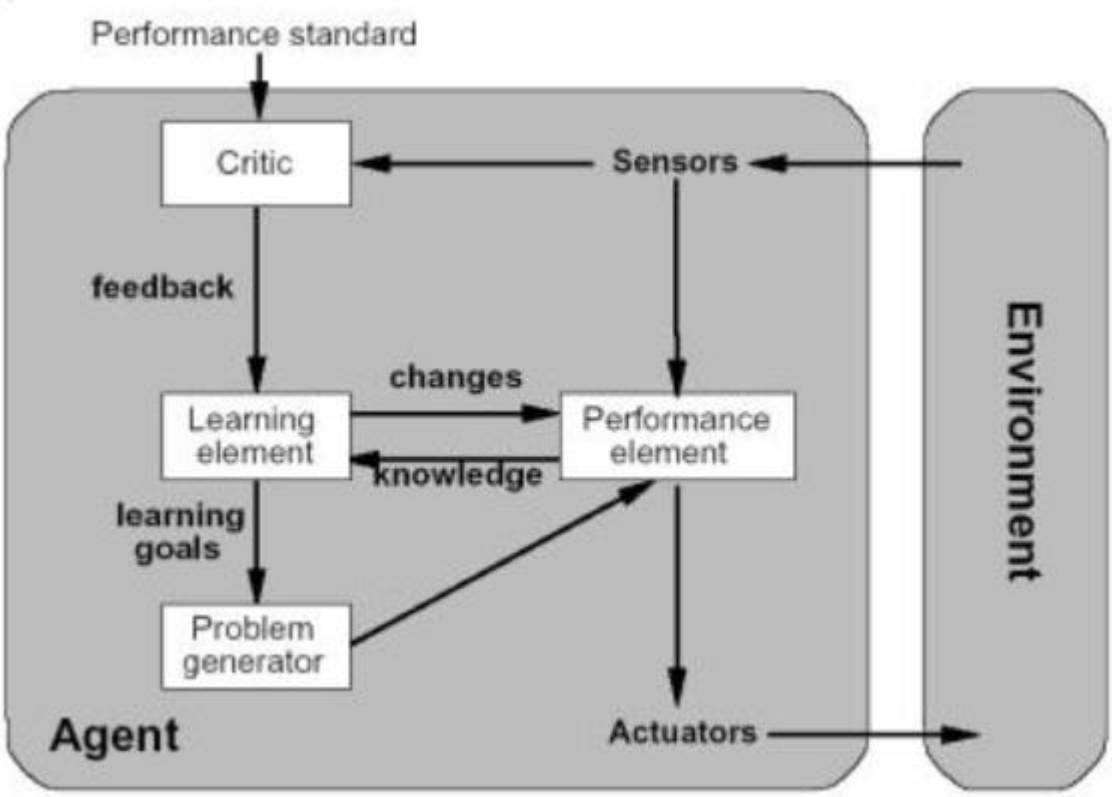
A learning agent has mainly four conceptual components, which are:

1. Learning element: It is responsible for making improvements by learning from the environment

2. Critic: The learning element takes feedback from critics which describes how well the agent is doing with respect to a fixed performance standard.

3. Performance element: It is responsible for selecting external action

4. Problem Generator: This component is responsible for suggesting actions that will lead to new and informative experiences.



A learning agent has mainly four conceptual components, which are:
1. Learning element: It is responsible for making improvements by learning from the environment
2. Critic: The learning element takes feedback from critics which describes how well the agent is doing with respect to a fixed performance standard.
3. Performance element: It is responsible for selecting external action

4. Problem Generator: This component is responsible for suggesting actions that will lead to new and informative experiences.

Describe each of learning component w.r.t. Medical diagnosis system

In the context of a medical diagnosis system, the components of a learning agent can be defined as follows:

1. **Environment**: The environment in this case would be the medical field, which includes various diseases, symptoms, medical tests, treatments, and patient data.
2. **Sensors**: The sensors in a medical diagnosis system would include data from various sources, such as medical records, laboratory test results, patient history, and medical imaging data.

3. **Learning Algorithms**: [This is a Combination of Critics , Learning Element and Problem generator]
The learning algorithms in a medical diagnosis system would be used to analyze the data from the sensors and identify patterns and relationships that can be used to diagnose diseases. These algorithms can include machine learning techniques such as decision trees, neural networks, and support vector machines.

4. **Actuators**: In a medical diagnosis system, the actuators would be the output of the system, which could include a diagnosis, a treatment plan, or a referral to a specialist.
Here's how each component would work in a medical diagnosis system:
1. Environment: The environment would provide the data for the medical diagnosis system. This data would include information about various diseases, their symptoms, the results of medical tests, and patient data.

2. **Sensors**: The sensors would collect data from various sources, such as medical records, laboratory test results, patient history, and medical imaging data. This data would be used as input for the learning algorithms.

3. **Learning Algorithms**: The learning algorithms would analyze the data from the sensors and identify patterns and relationships that can be used to diagnose diseases. For example, if a patient has a particular combination of symptoms and test results, the learning algorithm could identify a particular disease that is likely to be the cause.

4. **Actuators**: The actuator in a medical diagnosis system would be the output of the system. This could include a diagnosis, a treatment plan, or a referral to a specialist. The output would be based on the analysis of the data by the learning algorithms.

Overall, the learning components of a medical diagnosis system work together to analyse patient data and provide an accurate diagnosis based on the available information. As the system learns from new data, it can improve its accuracy and provide better diagnoses over time

List Down all the types of Agent Explain Each with a block diagram.[V Imp]

Agents can be grouped into four classes based on their degree of perceived intelligence and capability:

- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents

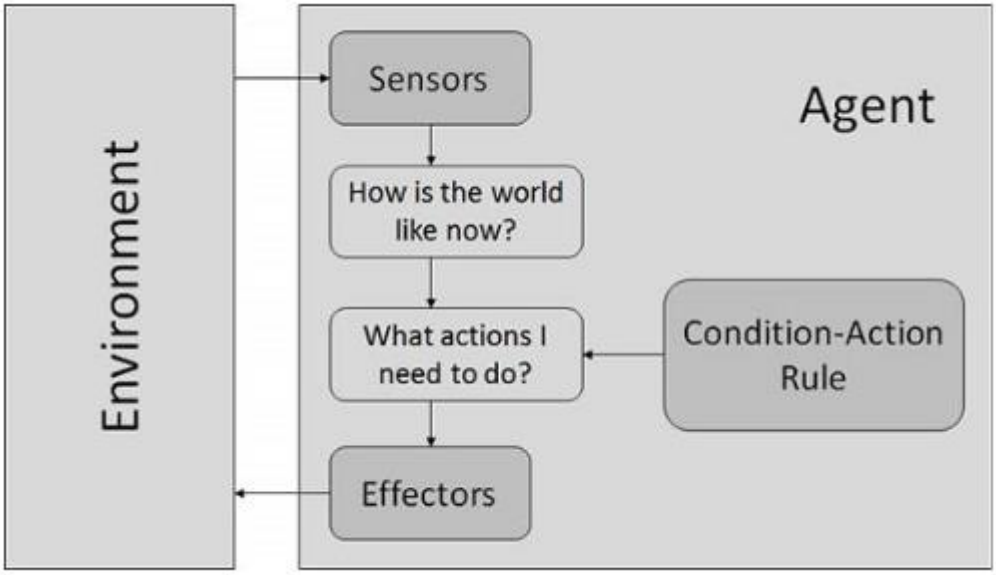
Simple Reflex Agent:

● This agent works only on the basis of current perception and it does not bother about the history or previous state in which the system was.

● This type of agent is based upon the condition-action rule. If the condition is true, then the action is taken, else not.

PROBLEMS FACED:

1. Very limited intelligence.
2. No knowledge about the non-perceptual parts of the state.
3. Operating in a partially observable environment, infinite loops are unavoidable.

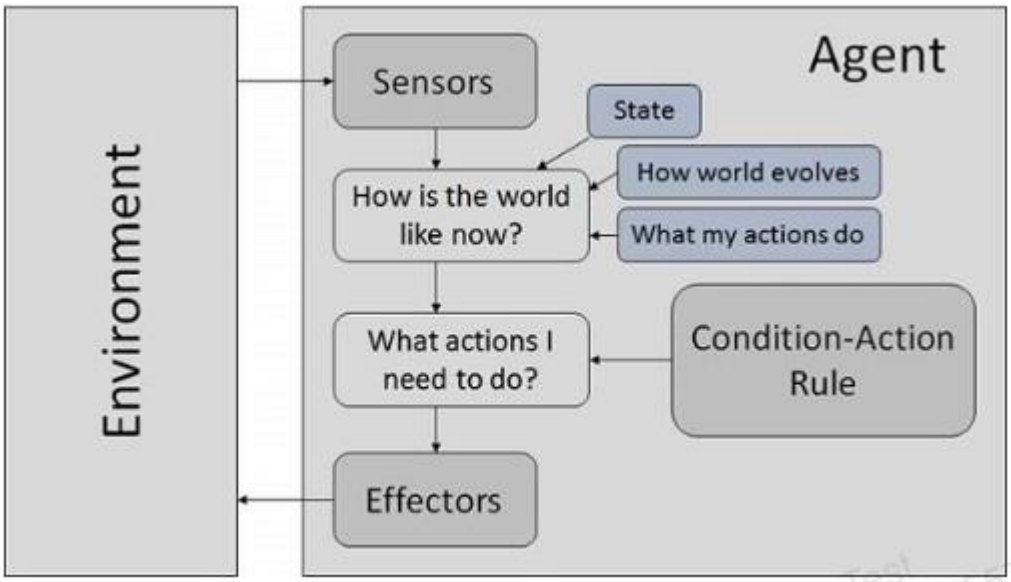


Model based reflex agents

● It works by finding a rule whose condition matches the current situation.

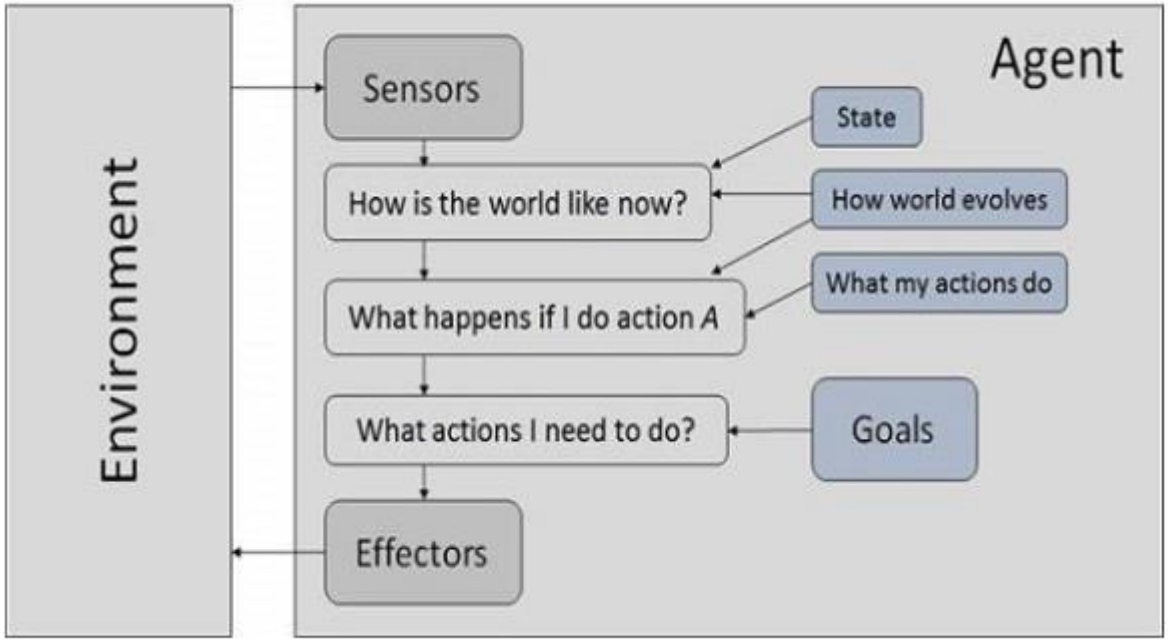
● It can handle partially observable environments.

● Updating the state requires information about how the world evolves independently from the agent and how the agent actions affect the world.



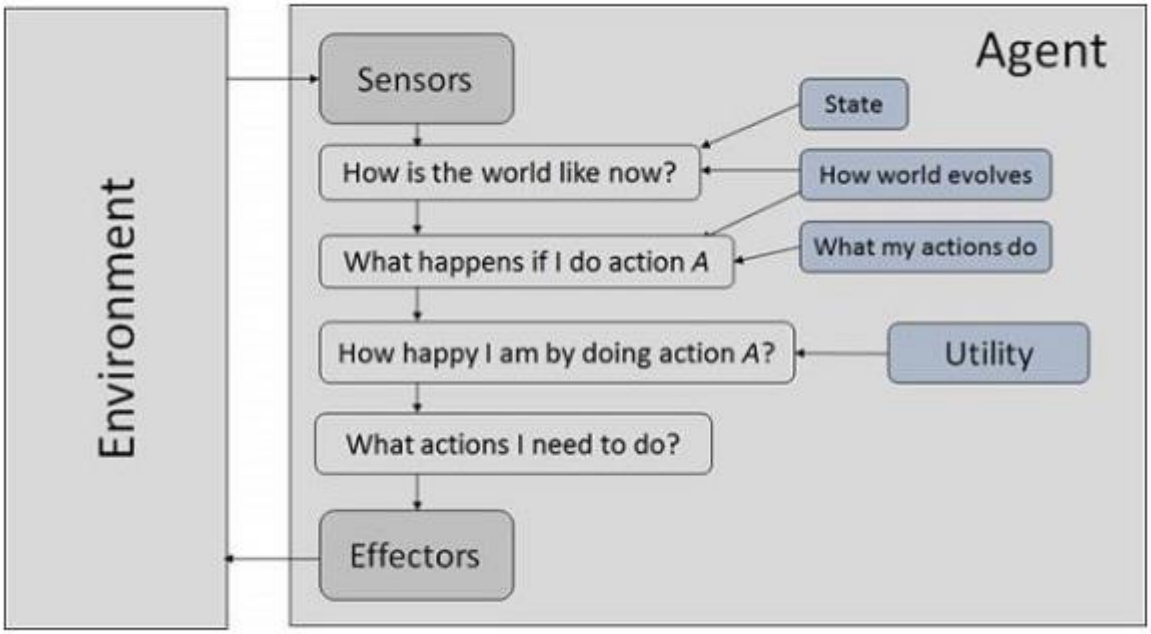
Goal based reflex agents

●The goal-based agent focuses only on reaching the goal set and hence the decision took by the agent is based on how far it is currently from their goal or desired state.
● Their every action is intended to minimize their distance from the goal.
● This agent is more flexible, and the agent develops its decision-making skill by choosing the right from the various options available.



Utility based agents

• These agents are more concerned about the preference(utility) for each state. When there are multiple options available, the utility-based agent takes the decision on the basis that how much satisfaction the agent gets from it.
• This approach was like somewhat adding emotions to the agent, because, after taking any decision, the agent ensures that "how happy I Am after taking this decision?".
• This agent was developed because sometimes achieving the desired goal is not enough. We may look for quicker, safer and cheaper alternate to reach the destination.



Compare Model Based, Goal based and Utility Based Agent

	Model Based Agent	Goal Based Agent	Utility Based Agent
Definition	Uses a model of the environment to make decisions	Achieves a set of predefined goals	Maximizes expected utility based on a predefined utility function
Decision-making process	Considers all possible actions and outcomes	Selects actions that achieve goals	Selects actions that maximize utility
Representation of the environment	Uses a model of the environment to predict outcomes of actions	Represents goals as a set of desirable states	Uses a utility function to evaluate outcomes of actions
Learning	Can learn the model of the environment through experience	Can learn new ways to achieve goals	Can learn the utility function through experience
Flexibility	Can adapt to changes in the environment by updating the model	Can adapt to changes in the environment by updating goals	Can adapt to changes in the environment by updating the utility function

Robustness	Can handle uncertainty in the environment by using probabilistic models	Can handle uncertainty in the environment by using alternative ways to achieve goals	Can handle uncertainty in the environment by using expected utility calculations
Computational complexity	Can be computationally expensive when the environment is complex	Can be computationally expensive when the goals are complex	Can be computationally expensive when the utility function is complex
Strengths	Suitable for complex environments with known models	Suitable for environments with well-defined goals	Suitable for environments with measurable outcomes
Weaknesses	Limited by the accuracy of the model	Limited by the ability to define goals	Limited by the accuracy of the utility function
Examples	Chess-playing agent	Home automation agent that achieves predefined goals such as energy efficiency or security	Shopping agent that maximizes discounts and rewards

Differentiate Between Informed Search and Uninformed Search Algorithm

	Informed Search	**Uninformed Search**
Definition	Uses heuristic information to guide the search	Does not use heuristic information to guide the search
Goal	Finds the optimal solution faster	Finds any solution
Knowledge	Requires additional knowledge about the problem domain	Can work with just the problem definition

Time complexity	Can have better time complexity if the heuristic is good	May have worse time complexity if the problem size is large
Space complexity	May use more memory to store additional information for the heuristic	Typically uses less memory
Completeness	Not guaranteed to find a solution if the heuristic is not admissible or consistent	Guaranteed to find a solution if one exists
Examples	A* search algorithm, Greedy Best First Search	Breadth First Search, Depth First Search
Efficiency	Can be more efficient for large, complex problems	Can be less efficient for large, complex problems
Optimality	Can find an optimal solution if the heuristic is admissible and consistent	May not find an optimal solution
Performance	The performance depends on the quality of the heuristic	The performance does not depend on additional knowledge

Compare Problem solving and Planning Agent

	Problem - Solving Agent	Planning Agent
Definition	Aims to find a solution to a problem by searching for a sequence of actions	Aims to generate a sequence of actions that will achieve a goal
Focus	Focuses on finding a solution to a specific problem	Focuses on planning a course of action to achieve a goal
Approach	Uses search algorithms to find a sequence of actions that solves the problem	Uses symbolic manipulation to generate a plan
Representation	Represents the problem domain using a state space	Represents the problem domain using logical rules and relationships

Domain Knowledge	Requires domain knowledge to define the problem space and search strategy	Requires domain knowledge to define the problem space and plan generation process
Execution	Executes the sequence of actions found by the search algorithm	Executes the plan generated by the planning process
Search Strategy	Employs a variety of search algorithms such as breadth-first, depth-first, and A* search	Employs planning techniques such as forward and backward chaining, and constraint satisfaction
Feedback	Feedback is based on the success or failure of the search algorithm to find a solution	Feedback is based on the success or failure of the plan execution
Flexibility	Can handle uncertain and changing environments, but may require additional search	Can handle uncertain and changing environments by generating alternative plans
Computational Efficiency	May require significant computational resources for complex problems	Can handle complex problems with efficient planning algorithms
Examples	Chess-playing agent, Route-planning agent	Robot task planning agent, Factory production scheduling agent

Overall problem-solving agents and planning agents have different approaches and requirements, and are suitable for different types of tasks.

● Problem-solving agents are focused on finding a solution to a specific problem using search algorithms to find a sequence of actions.

● Planning agents are focused on generating a sequence of actions that will achieve a goal using symbolic manipulation to generate a plan.

● The choice of agent type will depend on the specific requirements of the task and the nature of the problem domain