

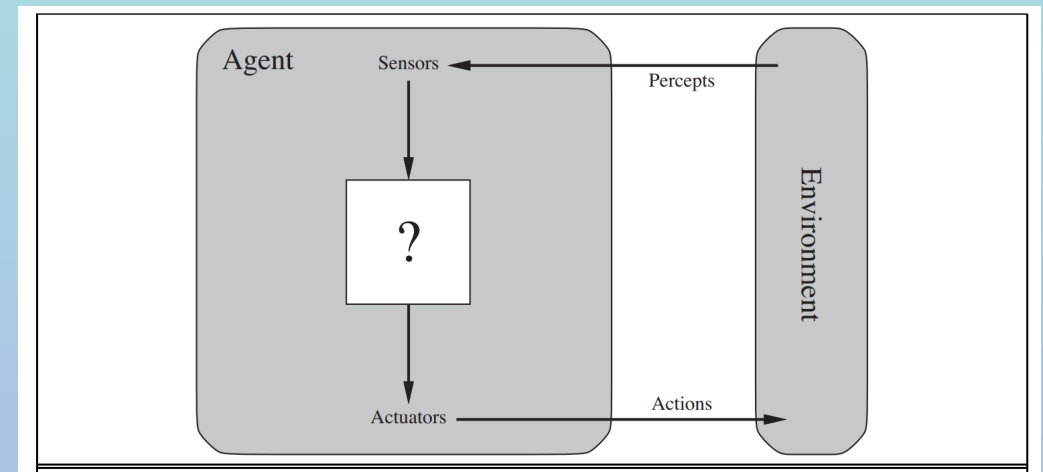
ACCA7026

AI : Agent, Environment and Learning

Dr. Seena Joseph

Artificial Intelligence Agents

- An artificial intelligence (AI) **agent** is a software program that can interact with its environment, collect data, and use the data to perform self-determined tasks to meet predetermined goals.
- **Agent** – Perceive **environment** through sensors and acts upon on environment using **actuators**



Agents

- A **human agent** has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for actuators (actions/behaviour).
- A **robotic agent** has cameras and infrared range finders for the sensors, and various motors for actuators.
- A **software agent** responds to keystrokes, file contents, and network packets as sensory inputs by creating files, showing on the screen, and delivering network packets.
- Agent's behaviour is the agent's function that maps the percept sequence to an action.
 - Agent function – abstract mathematical description
 - Agent program – implementation of the function and running within some physical system.

Agents

- **Performance Measure of Agent** – It is the criteria, which determines how successful an agent is.
- **Behavior of Agent** – It is the action that agent performs after any given sequence of percepts.
- **Percept** – It is agent's perceptual inputs at a given instance.
- **Percept Sequence** – It is the history of all that an agent has perceived till date.
- **Agent Function** – It is a map from the precept sequence to an action.
- **Rational Agent** – Capable of doing expected actions to maximize its performance measure, on the basis of –
 - Its percept sequence
 - Its built-in knowledge base

Agents

- The **performance measures**, which determine the degree of success.
- Agent's **Percept Sequence** till now.
- The agent's **prior knowledge about the environment**.
- The **actions** that the agent can carry out.
- An agent that is rational will always act in the best possible way, defined as acting in a way that maximizes the agent's success within the given perceptual sequence. **Performance Measure, Environment, Actuators, and Sensors (PEAS)** define the problem that the agent resolves.

Agents

- **Performance Measure** – Criteria used to evaluate agent's success
- **Environment** - The external context/world that agent interacts
- **Actuators** – Mechanism through which the agent takes the action pr performs operations in the environment.
- **Sensors** - The mechanism through which the perceives or gather information about the environment.

Agents

- PEAS description of the task environment of an automated taxi

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Agents

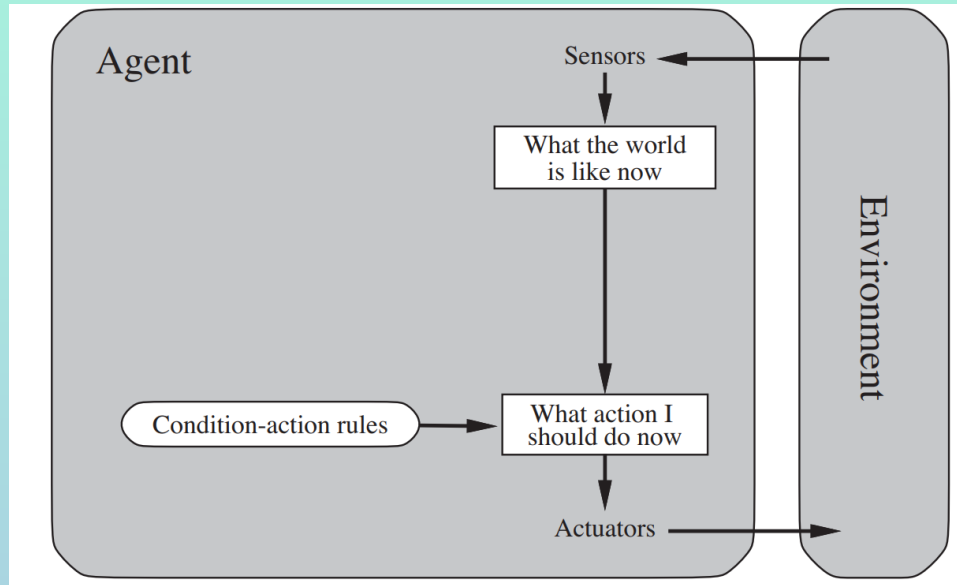
- Identify the PEAS description for the following agent types
 - Medical diagnosis system
 - Satellite image analysis system
 - Interactive English tutors

Agents

- Structure of intelligent agents
 - Agent = Architecture + Agent Program
 - Architecture = the machinery that an agent executes on.
 - Agent Program = an implementation of an agent function.

Agent Types

Simple reflex agents



- They choose actions only based on the current percept.
- They are rational only if a correct decision is made only based on current precept.
- Their environment is completely observable.

Agent Types

Simple reflex agents

Example

Vacuum cleaner Agent with two locations.

Condition – If the current location is dirty

Action – Clean

Condition – if the current location is clean

Action – Move to the other location

It only reacts to current state (clean or dirty)

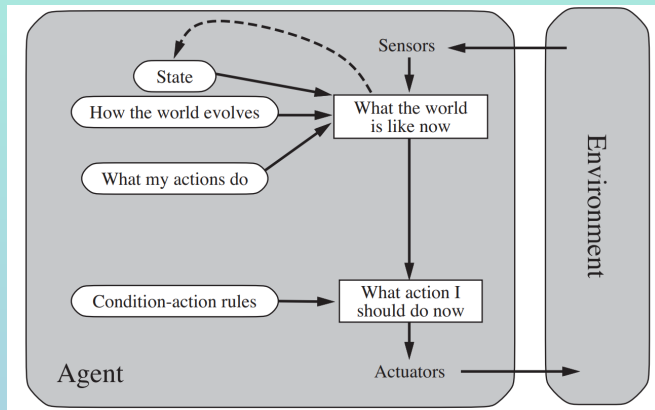
Agent Types

Model based reflex agent

- They use a model of the world to choose their actions. They maintain an internal state.
 - Model – knowledge about “how the things happen in the world”.
 - Internal State – It is a representation of unobserved aspects of current state depending on percept history.
 - State update – Agent has the knowledge of how the environment works.

Agent Types

Model bases reflex agent



- This agent can remember the structure of the environment and plan future actions based on its internal representation of the world.
- Example : Robot in a maze

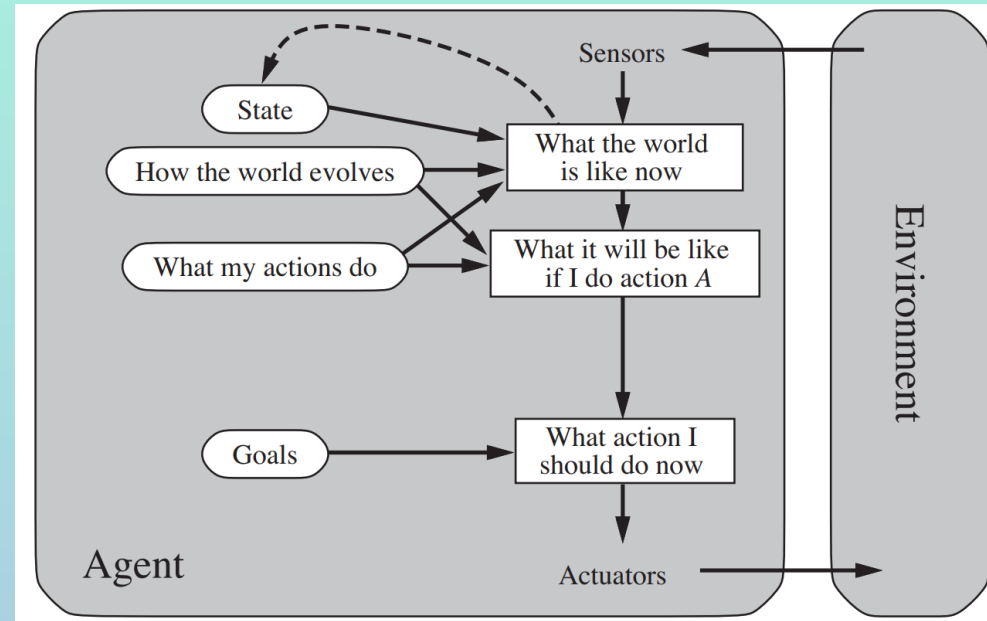
Agent Types

Goal based Agents

- Goal-based agents are AI agents that use information from their environment to achieve specific goals. They employ search algorithms to find the most efficient path towards their objectives within a given environment.
- Also known as rule based agent as they use predefined rules to achieve the goal.

Agent Types

Goal based agents



Agent Types

Goal based agents

- Perception: The agent perceives its environment using sensors or other input devices to collect information about its surroundings.
- Reasoning: The agent analyzes the information collected and decides on the best course of action to achieve its goal.
- Action: The agent takes actions to achieve its goal, such as moving or manipulating objects in the environment.
- Evaluation: After taking action, the agent evaluates its progress towards the goal and adjusts its actions, if necessary.
- Goal Completion: Once the agent has achieved its goal, it either stops working or begins working on a new goal.
- Example : Chess Game, Google Bard

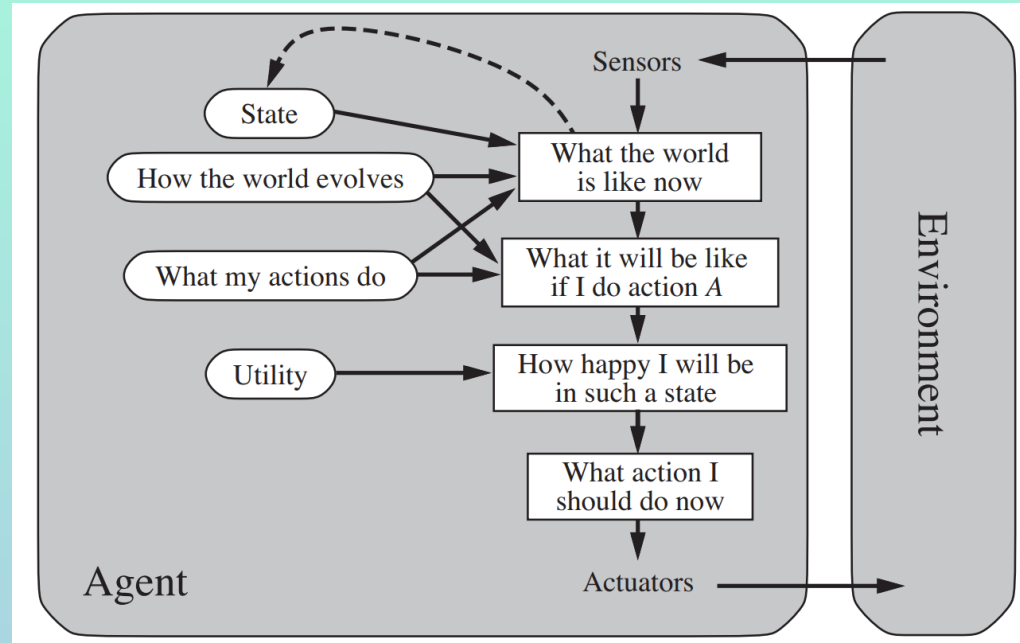
Agent Types

Utility based agents

- The goal of a utility-based agent is to make decisions that result in a high utility state (state of being useful/beneficial).
- The expected utility of each potential result is then calculated using the utility function and the probability distribution.
- At the end, it chooses the action that has the highest predicted utility, and it does this again at every time step.

Agent types

Utility based agents



- Example :Anthropic Claude, an AI tool whose goal is to help cardmembers maximize their rewards and benefits from using cards.
- Autonomous Drone delivery
- Autonomous Vehicles

Agent types

Learning Agents

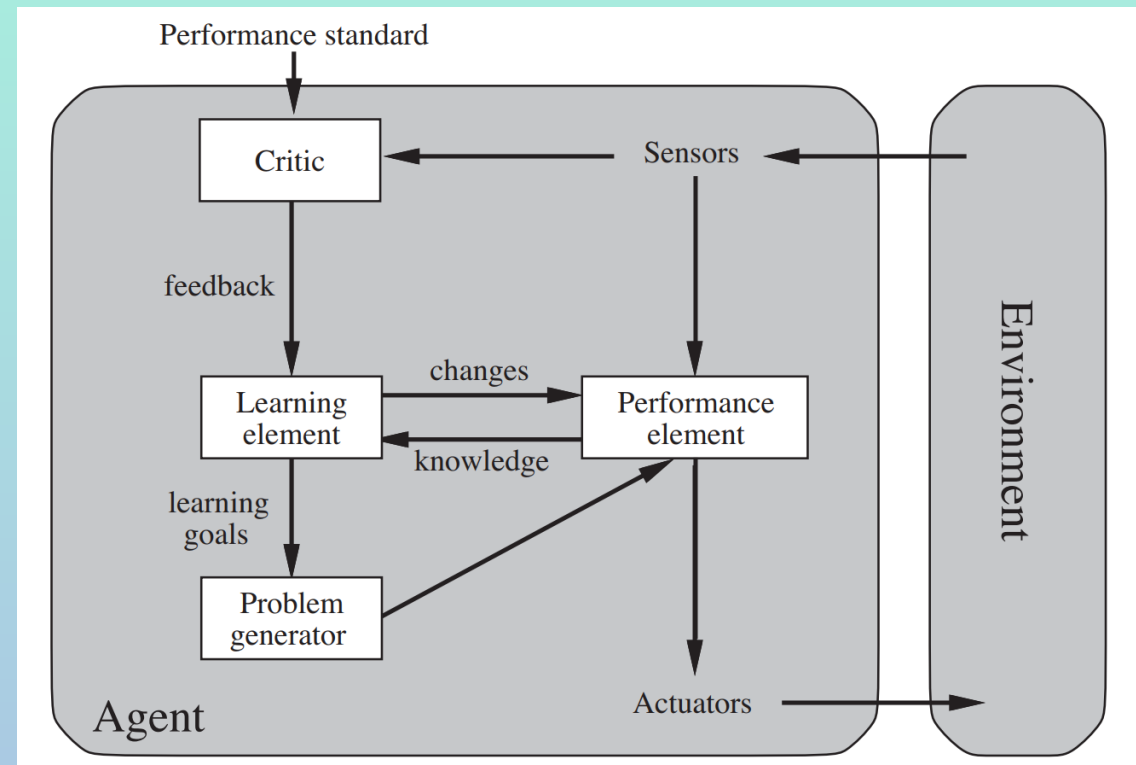
An AI learning agent is a software agent that can learn from past experiences and improve its performance. It initially acts with basic knowledge and adapts automatically through machine learning.

Four main components of a learning agent

1. **Learning Element:** It is responsible for learning and making improvements based on the experiences it gains from its environment.
2. **Critic:** It provides feedback to the learning element by the agent's performance for a predefined standard.
3. **Performance Element:** It selects and executes external actions based on the information from the learning element and the critic.
4. **Problem Generator:** It suggests actions to create new and informative experiences for the learning element to improve its performance.

Agent types

Learning agent



Agent types

Learning Agent

- Observation: Using sensors or other inputs, the learning agent looks around it.
- Learning: The agent uses statistical models and algorithms to analyze data; it gains knowledge via feedback on its performance and actions.
- Action: The agent acts in its surroundings to determine how to behave based on its acquired knowledge.
- Feedback: Through incentives, penalties, or indications from the surroundings, the agent gets information about their performance and behaviors.
- Adaptation: The agent updates its knowledge and adjusts to its surroundings by modifying its behavior and decision-making procedures based on input.

Example : Recommendation system

Environment

Properties of environment

- **Discrete / Continuous** – If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete; otherwise it is continuous).
 - Discrete – Chess Game
 - Continuous - Autonomous Driving
- **Observable / Partially Observable** – If it is possible to determine the complete state of the environment at each time from the percepts it is observable; otherwise it is only partially observable.
 - Observable – Chess Game
 - Partially Observable - Autonomous Driving

Environment

Properties of environment

- **Static / Dynamic** – If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.
 - Discrete – A crossword puzzle
 - Continuous - Autonomous Driving
- **Single agent / Multiple agents** – The environment may contain other agents which may be of the same or different kind as that of the agent.
 - Single – A vacuum cleaning robot
 - Multiple - Soccer match video game

Environment

Properties of environment

- **Accessible / Inaccessible** – If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.
 - Accessible - board game
 - Inaccessible - robotic exploration on space
- **Deterministic / Non-deterministic** – If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.
 - Deterministic - a robot picking up objects from conveyor belt
 - Non-deterministic – Weather forecasting system
- .

Environment

Properties of environment

- **Episodic / Non-episodic or sequential** – In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead.
- **Episodic** – Spam email filtering
- **Non episodic** – Autonomous drone navigation

Learning

- Designing **agents** that interact with **environment** and **learn** how to **solve** the environment by systematic trial and error approach.
- Problem - Goal based agent interact with an uncertain environment

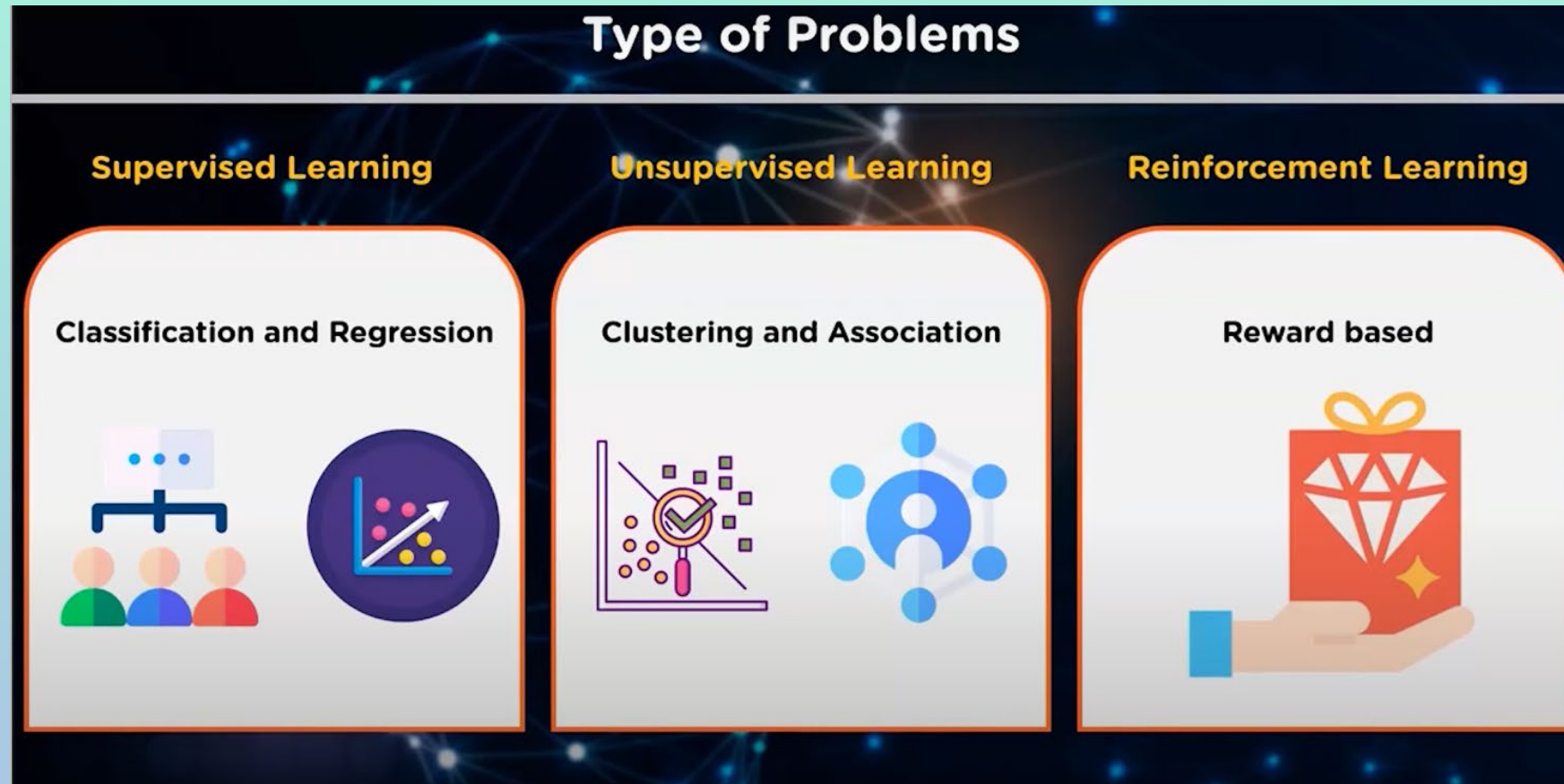
Learning

Learning – What to do -> How to map situations to action -> so as to maximize a numerical reward signal.

- Supervised learning – learning from a training set of labelled examples provided by a knowledgeable external supervisor.
 - Not adequate for learning from interaction
- Unsupervised learning – finding the hidden structures from unlabelled data.
- Reinforcement Learning trying to maximize the reward signal instead of finding the hidden structure.
 - Challenges : trade off between – exploitation (choosing the best solution) and exploration (trying out new option for better solution)

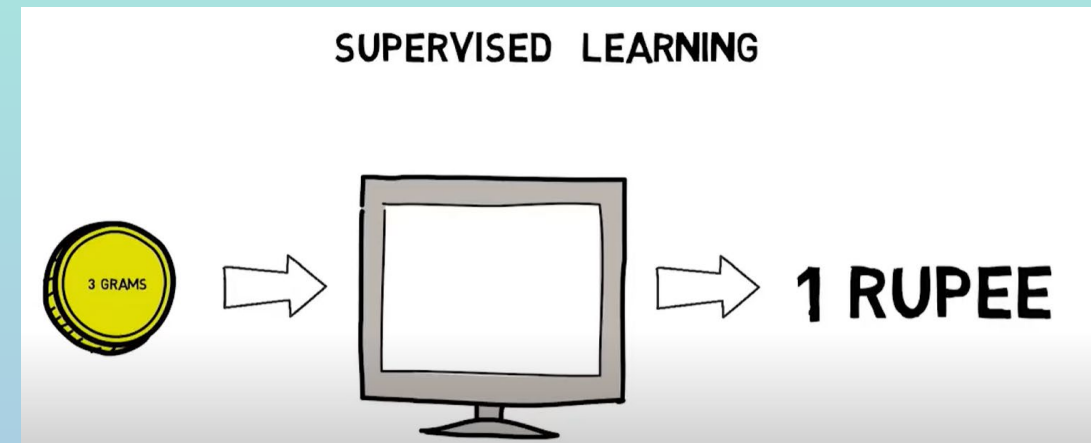
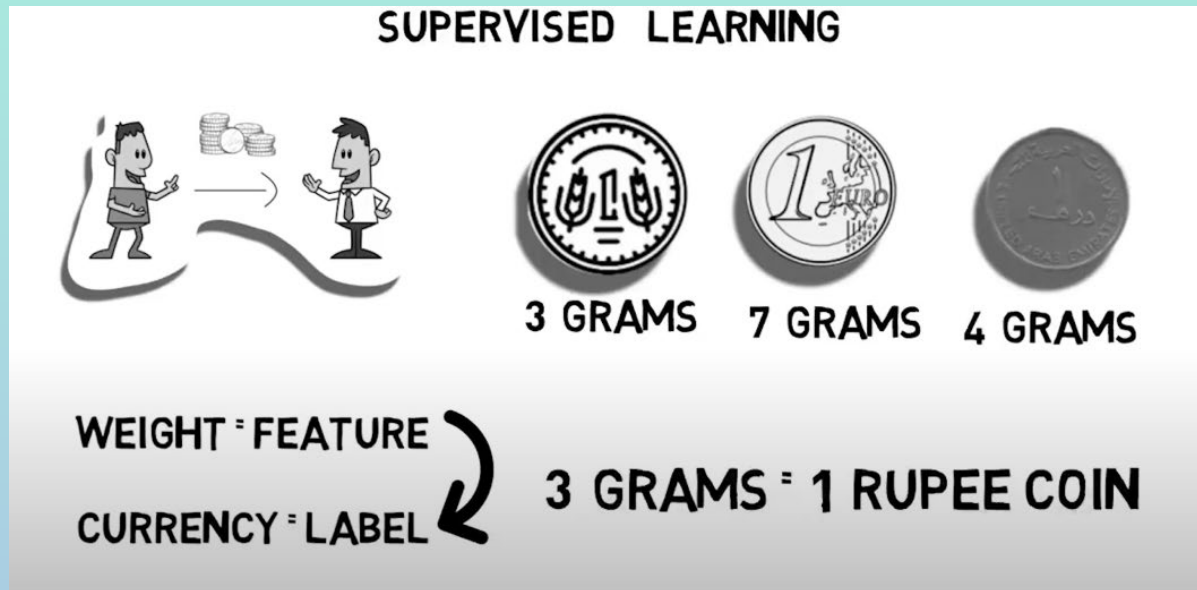
Learning

Types of problems



Learning

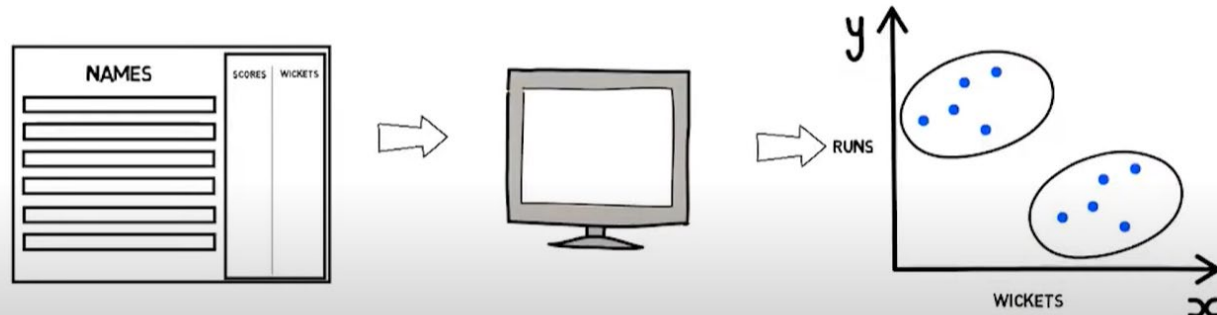
Supervised learning



Learning

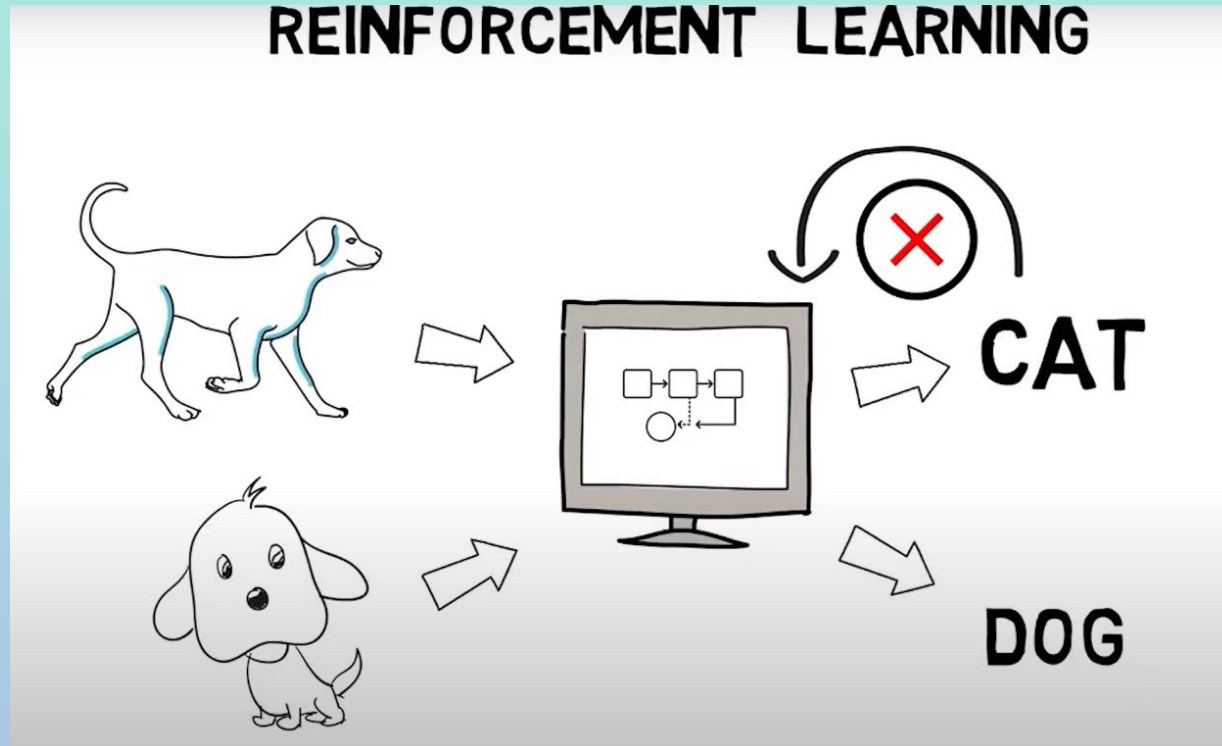
Unsupervised

UNSUPERVISED LEARNING



Learning

Reinforcement learning



Learning

Reinforcement



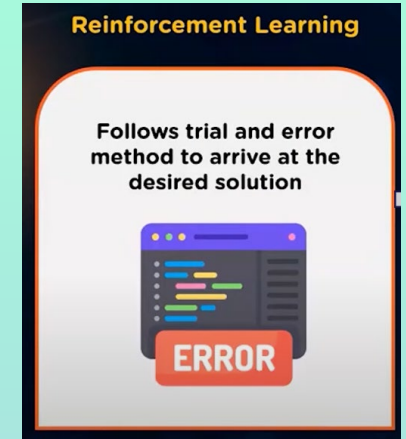
Learning

Applications



Reinforcement learning

- Train a dog to catch the ball, reward based.
- Machine learn from its environment using rewards and errors.
- Follows trial and error problem approach.
- Train the model to come to an optimal solution for a problem by taking decision by itself.
- Consist of an **environment**, which an **agent** will interact with to learn to reach a **goal** or **perform an action**.

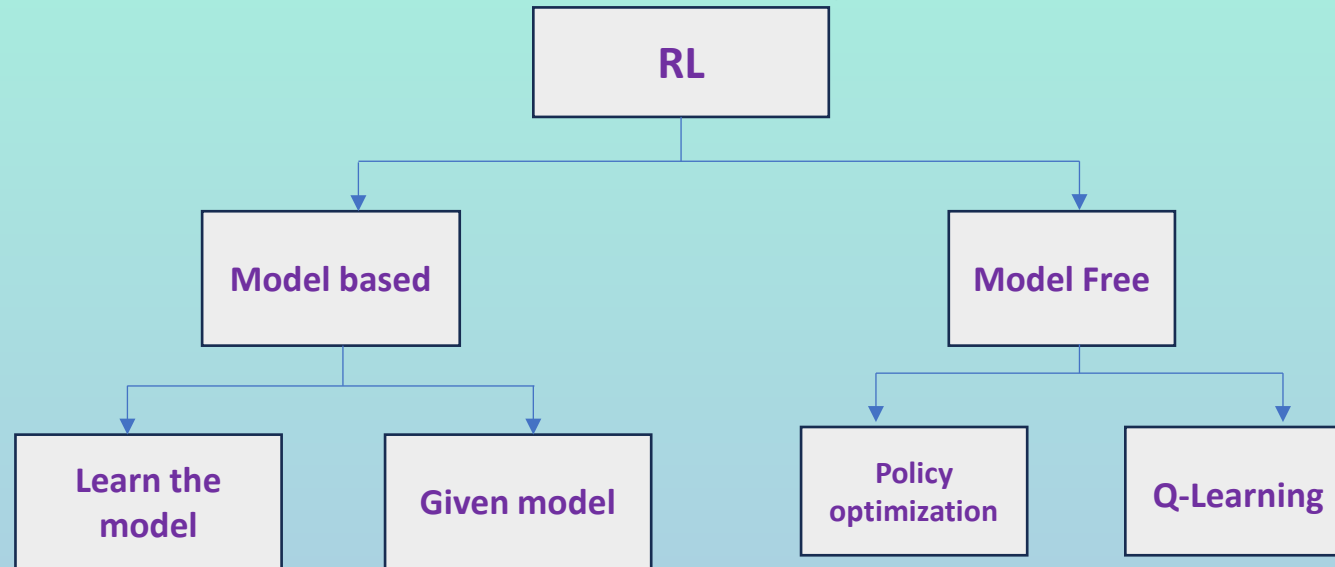


Reinforcement learning

- Terms :
 - Agent – an entity that interact with the environment; responsible for making decision
 - Environment – external system or the process
 - State – the current position of an agent in an environment; the transition to the next state is based on the current action in contrast to the previous actions
 - Action – the step taken by the agent when it is in a particular state
 - Reward – for every action the agent will get a positive or negative reward
 - Policy – mapping between an action and a present state; strategy of choosing an action given a state in expectations of better outcome. Simply an agent's way of behaving at a given time/recipe for acting in each state.
 - Episode – complete sequence of interaction between an agent and the environment; starting from the initial state to the termination state; when an agent ends up in a terminating state, and can't take a new action.
 - Value function – specify what is good in long run; value of a state is the total amount of reward an agent can expect to accumulate over the future.
 - Exploration
 - Exploitation

Reinforcement learning

- Types of reinforcement learning



- Examples of model based RL and Model Free RL ?

Reinforcement learning

Q-Learning

- RL policy that finds the next best action given a current state.
- Chooses the action at random and maximize the reward.
- Model-free environment – agent attempts to construct an optimal policy directly by interacting with the environment.
- Important terms in Q-Learning
 - State
 - Action
 - Reward
 - Episode
 - Q-value
 - Temporal difference

Reinforcement learning

States

The State, S , represents the current position of an agent in an environment



Action

The Action, A , is the step taken by the agent when it is in a particular state



Rewards

For every action, the agent will get a positive or negative reward



Episodes

When an agent ends up in a terminating state and can't take a new action



Q-Values

Used to determine how good an Action, A , taken at a particular state, S , is $Q(A,S)$



Temporal Difference

A formula used to find the Q-Value by using the value of current state and action and previous state and action

Reinforcement learning

- Elements of RL
 - Policy : What to do
 - Reward : What is good
 - Value : what is good because it predicts rewards
 - Model : what follows what
- **Q-values** guide the agent's decision-making process by estimating the long-term value of actions, whereas **rewards** provide immediate feedback to the agent's actions and drive the learning process by indicating the short-term consequences of those actions

Reinforcement learning

- Q-learning Characteristics
 - Characteristics of RL applies to Q-learning
 - An input and output system, rewards, environment, Markov decision process and training & interfaces.
 - Plus the additional characteristics
 - Finite number of states
 - Finite number of actions
 - Classic Q-learning problems
 - Maze problem
 - Other examples ?

Reinforcement learning

- Q-learning

- Q-value – indicates the quality of a particular action a in a given state, s :

$Q(s, a)$

Q-values are the current estimates of total rewards that can be obtained through all remaining steps in the current episode if the agent is in state, s and in action, a .

- Q-values estimates how much additional reward we can accumulate through all remaining steps in the current episode if the agent is in state, s and takes action, a .
 - It tells the agent how, good is to take a certain action in a specific situation
 - Q-values increase as the agent gets closer to the highest reward.
 - Q-values are stored in a Q-Table [one row for each possible state and one column for each possible action)
 - The Q-Table provides values that enable the AI agent to operate optimally in each given state, giving it the best path to the maximum reward.
 - Thus Q-Table therefore represents the AI agent's policy for acting in the current environment.

Reinforcement learning

- Temporal Difference (TD)

The diagram illustrates the Temporal Difference (TD) equation, $TD(s_t, a_t) = r_t + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$. It features five explanatory boxes with arrows pointing to the corresponding terms in the equation:

- Top-left box:** "The reward received for the action taken in the previous state" points to r_t .
- Top-right box:** "The largest Q-value available for any action in the current state (the largest predicted sum of future rewards)" points to $\max_a Q(s_{t+1}, a)$.
- Bottom-left box:** "Temporal difference for the action taken in the previous state" points to the entire equation.
- Bottom-middle box:** "The discount factor (between 0 and 1)" points to γ .
- Bottom-right box:** "The Q-value for the action taken in the previous state" points to $Q(s_t, a_t)$.

$$TD(s_t, a_t) = r_t + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$$

Reinforcement learning

- Temporal difference (TD)
 - TD learning compares the difference between the current estimate (based on the observed reward and the estimate of the value of the next state) and previous estimate to update the estimate of the predicted reward for doing a certain action in a given state. The value estimate is then modified using this difference to move it in the direction of the actual expected reward.
 - Method of calculating how much the Q-value for the action taken in the previous state should be changed based on what the AI agent has learned about the Q-values for the current state's actions.
 - Previous q-values are updated after each step
 - Discount factor : mechanism for discounting future rewards; concept: the value of receiving the reward in the future is less than the value of receiving the same reward today.

Reinforcement learning

- Bellman equation

The diagram illustrates the Bellman equation for Q-learning. It features the equation $Q^{new}(s_t, a_t) = Q^{old}(s_t, a_t) + \alpha \cdot TD(s_t, a_t)$ in the center. Four boxes with arrows explain the components: 'The old Q-value for the action taken in the previous state' points to $Q^{old}(s_t, a_t)$; 'The temporal difference for the action taken in the previous state' points to $TD(s_t, a_t)$; 'The learning rate (between 0 and 1)' points to α ; and 'The new Q-value for the action taken in the previous state' points to $Q^{new}(s_t, a_t)$.

The old Q-value for the action taken in the previous state

The temporal difference for the action taken in the previous state

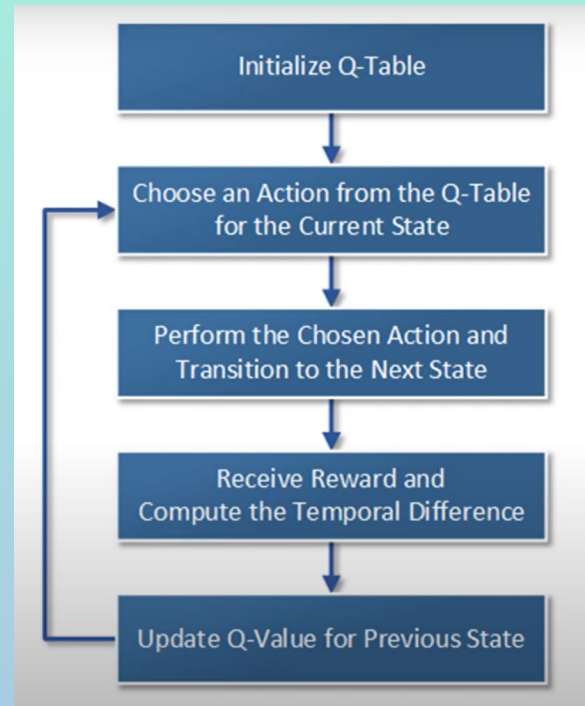
$$Q^{new}(s_t, a_t) = Q^{old}(s_t, a_t) + \alpha \cdot TD(s_t, a_t)$$

The new Q-value for the action taken in the previous state

The learning rate (between 0 and 1)

Reinforcement learning

- Q-learning process



Reinforcement learning

- Inference mode
 - Once the Q-learning model is fully trained, it can be used for inference .
 - In inference mode – Q values are no longer updated
 - Agent will take the action with the largest Q-value

Practical Task

- Practical task – Important
 - Implement and train a Q-learning agent to play Tic-Tac-Toe game.
 - Define environment (states, actions, transition probabilities, rewards)
 - Functions to represent the game state, execute actions, and determine the rewards
 - Implement and train the agent
 - Evaluate performance
 - Tune hyperparameters
 - Learning rate, discount factor and exploration rate

Extra Learning materials

- [Supervised and Unsupervised learning – GeeksforGeeks](#)
- [Machine Learning Basics | What is Machine Learning? | Introduction to Machine Learning | by Adonis Tejeda Rivera | Analytics Vidhya | Medium](#)
- [Reinforcement Q-Learning from Scratch in Python with OpenAI Gym – LearnDataSci](#)
- [Reinforcement Learning Full Course | Reinforcement Learning In Machine Learning | Simplilearn \(youtube.com\)](#)
- [RL 7: Monte-Carlo Method | Reinforcement Learning \(youtube.com\)](#)
- [Foundations of Q-Learning \(youtube.com\)](#)

References

- Russell, S. and Norvig, P. (2009) Artificial Intelligence: A Modern Approach (3rd edition), Prentice Hall