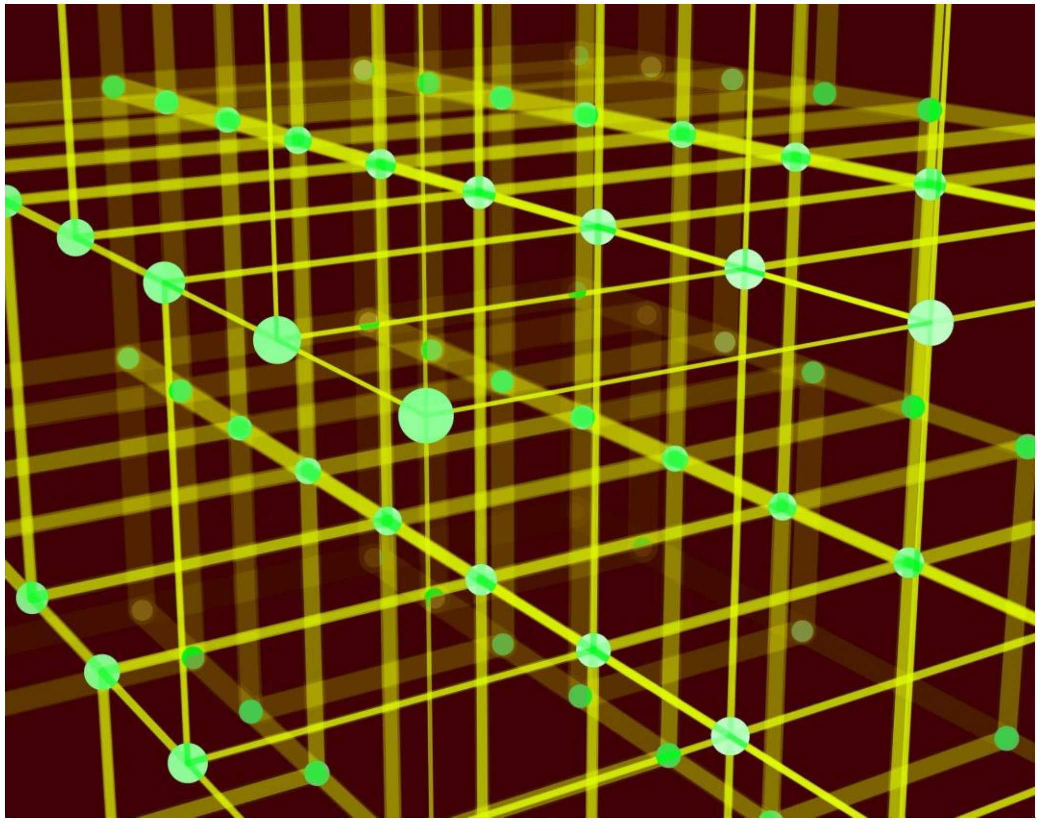
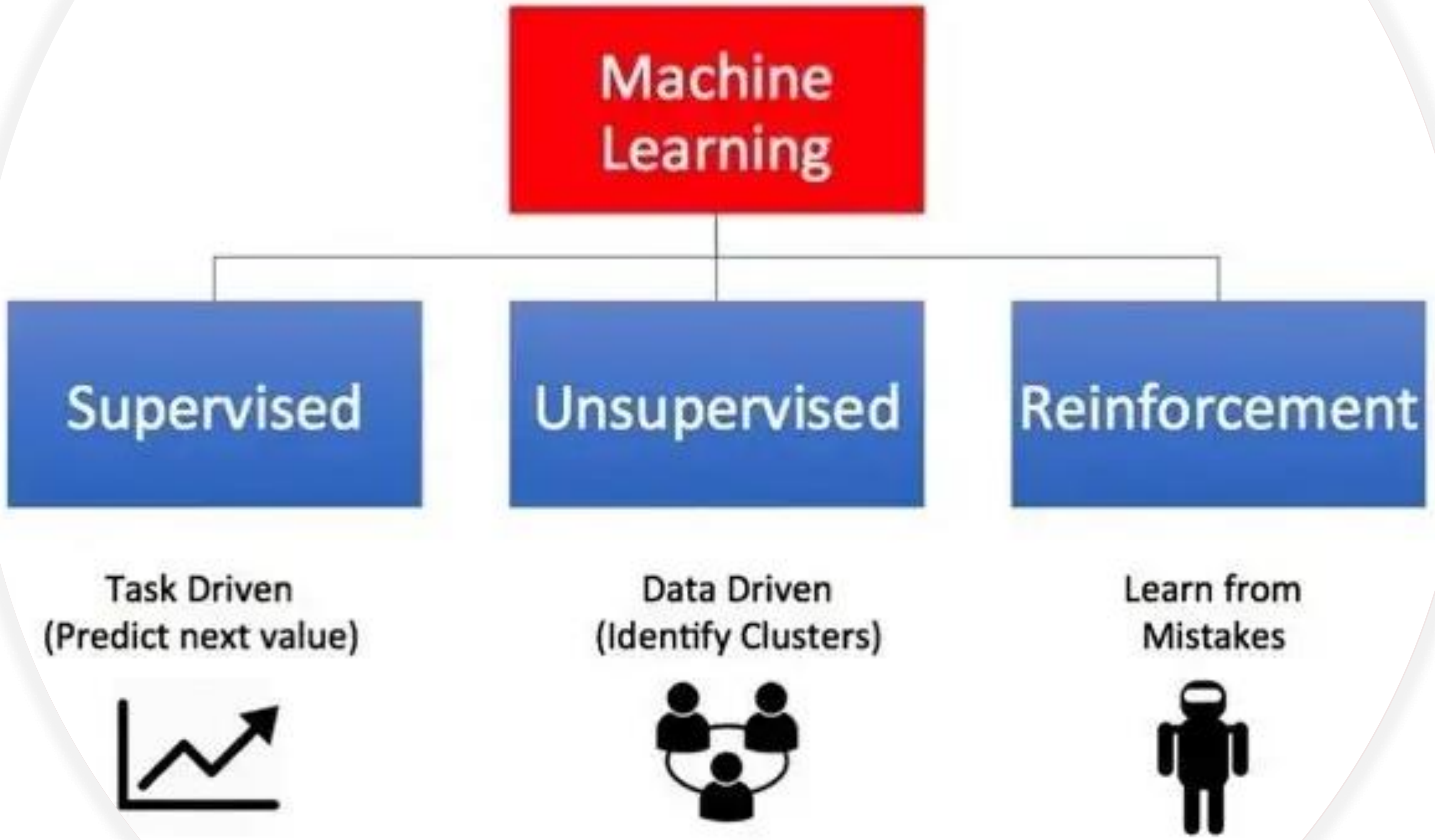
Reinforcement Learning

Dr. Venkataramana Veeramsetty



Types of Machine Learning



What is Reinforcement Learning?

Reinforcement Learning(RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences.

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions.

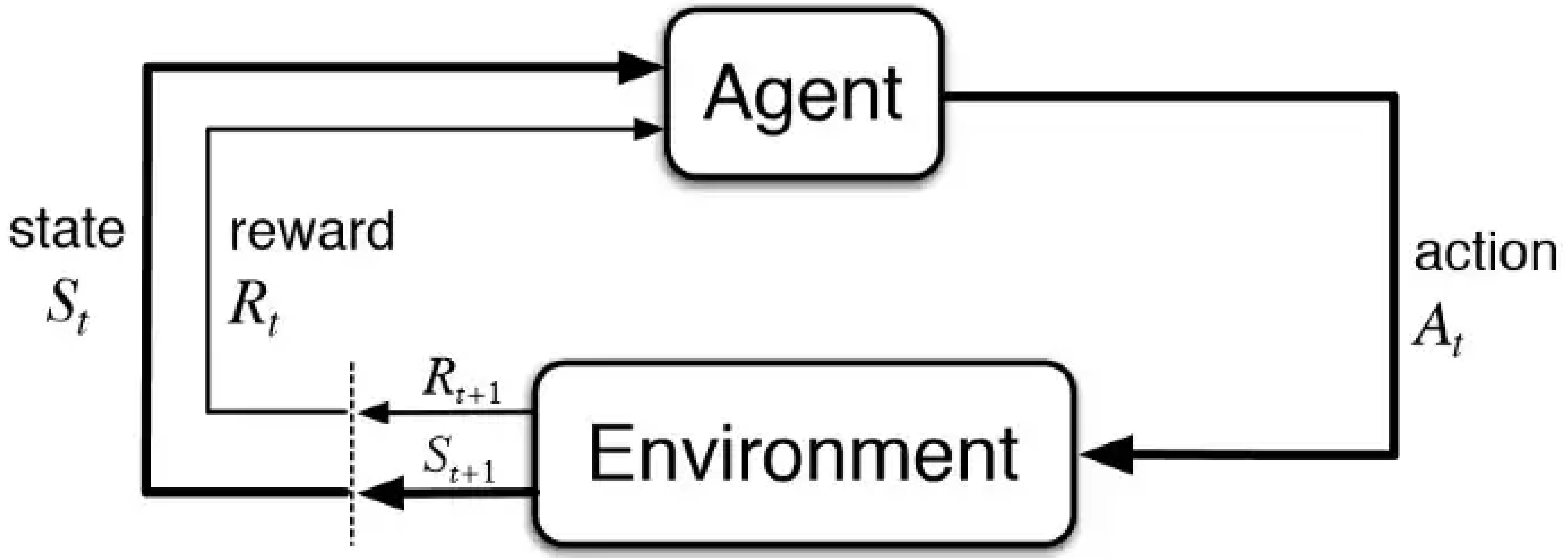
For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.

Difference between learning algorithms

The goal in unsupervised learning is to find similarities and differences between data points, in the case of reinforcement learning the goal is to find a suitable action model that would maximize the **total cumulative reward** of the agent.

In supervised learning where the feedback provided to the agent is **correct set of actions** for performing a task, reinforcement learning uses **rewards and punishments** as signals for positive and negative behavior.

Action-Reward feedback loop of a generic RL model.



Basic elements of an RL problem

Environment – Physical world in which the agent operates

State – Current situation of the agent

Reward – Feedback from the environment

Policy – Method to map agent’s state to actions

Value – Future reward that an agent would receive by taking an action in a particular state

Classification of Reinforcement Learning

- Active reinforcement learning
- Passive reinforcement learning

Passive Reinforcement Learning

The agent's policy is fixed which means that it is ***told what to do***.

The goal of a passive RL agent is to execute a fixed policy (sequence of actions) and evaluate it

Active Reinforcement Learning

An agent ***needs to decide what to do*** as there's no fixed policy that it can act on.

In active reinforcement learning, the goal of an active RL agent is to act and learn an optimal policy.

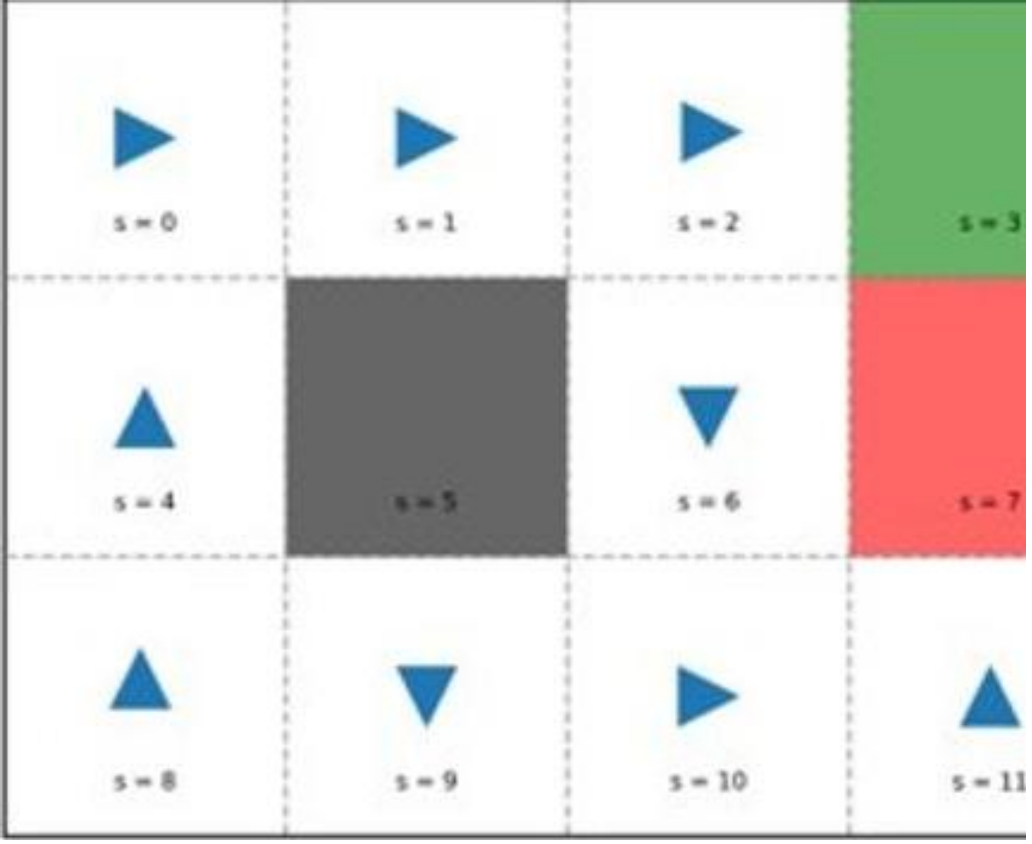
Passive Learning Techniques

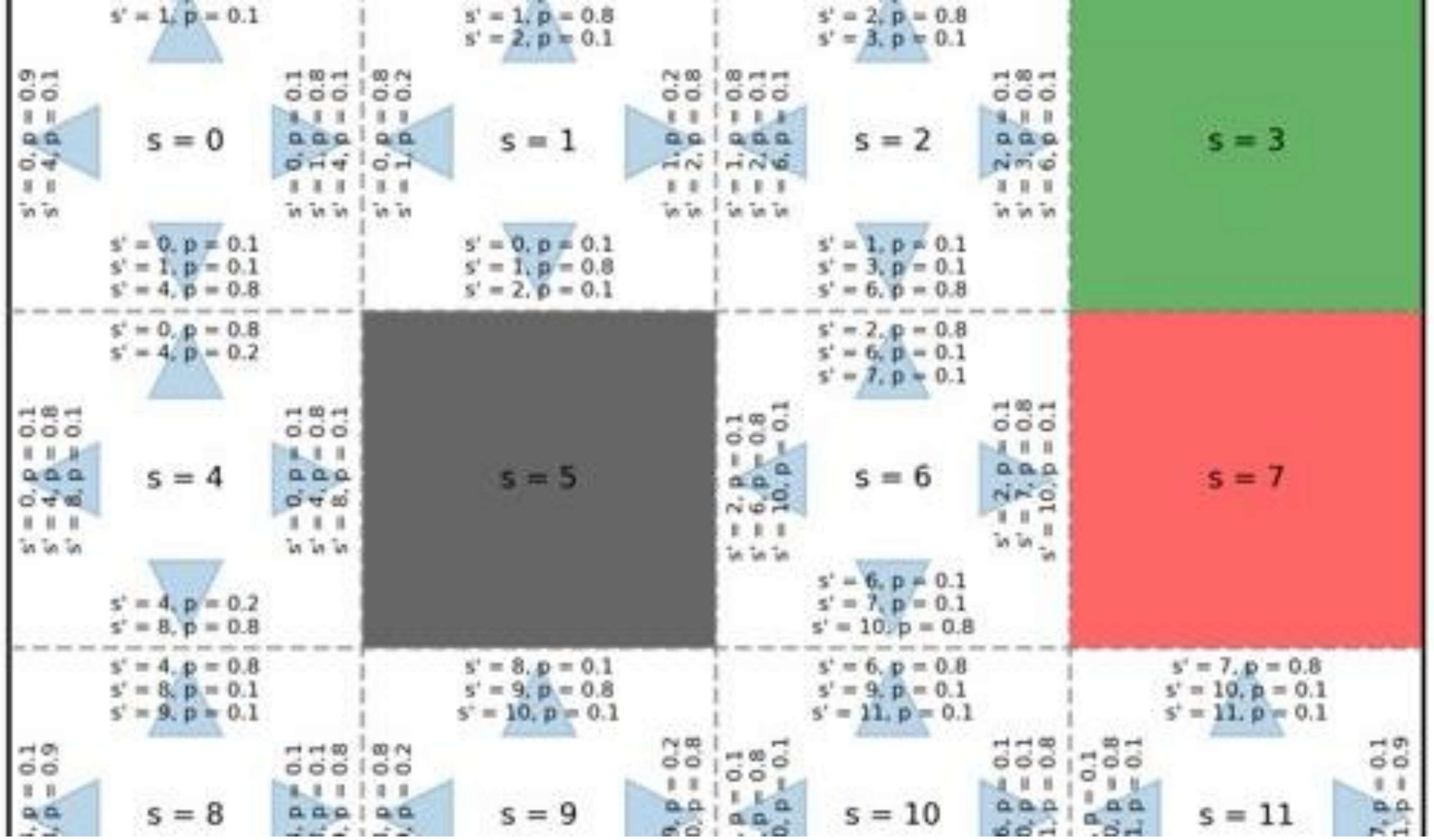
- ***Direct Utility Estimation***
- ***Adaptive Dynamic Programming(ADP)***
- ***Temporal Difference Learning (TD)***

Direct Utility Estimation

In this method, the agent executes a **sequence of trials or runs** (sequences of states-actions transitions that continue until the agent reaches the terminal state).

Each trial gives a sample value and the agent estimates the utility based on the samples values. Can be calculated as **running averages of sample values**.





Direct Utility Estimation

8-4-0-4-0-1-2-3

-0.04-0.04-0.04-0.04-0.04-0.04-0.04+1=0.72

Adaptive Dynamic Programming(ADP)

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) U^{\pi}(s')$$

ADP is a smarter method than Direct Utility Estimation as it runs trials to learn the model of the environment by estimating the utility of a state as a sum of reward for being in that state and the expected discounted reward of being in the next state.

It uses value-iteration method to compute utility value of state in the game

The algorithm converges fast but can become quite costly to compute for large state spaces.

ADP is a model based approach and requires the transition model of the environment

Temporal Difference Learning (TD)

Temporal Difference learning does not require the agent to learn the transition model.

While ADP adjusts the utility of s with all its successor states, TD learning adjusts it with that of a single successor state s'

TD is slower in convergence but much simpler in terms of computation.

$$U^{\pi}(s) = U^{\pi}(s) + \alpha(R(s) + \gamma U^{\pi}(s') - U^{\pi}(s))$$

Active Reinforcement Learning Techniques

Adaptive Dynamic Programming(ADP)

Q-Learning

Adaptive Dynamic Programming(ADP)

- As the goal of an active agent is to learn an optimal policy, the agent needs to learn the expected utility of each state and update its policy. Can be done using a passive ADP agent and then using value or policy iteration it can learn optimal actions.

$$
\begin{aligned}
v(s) &= \max_{a_i} \left[r(s) + \gamma \sum_{s'} p(s'|s, a = a_i) v(s') \right] \\
&= r(s) + \gamma \max_{a} \left[\sum_{s'} p(s'|s, a) v(s') \right]
\end{aligned}
$$

Q-Learning

- Q-learning is a TD learning method which does not require the agent to learn the transitional model, instead learns Q-value functions *Q(s, a)* .

$$\boxed{\text{New } Q(s,a)} = \boxed{Q(s,a)} + \boxed{\alpha}\boxed{[R(s,a)} + \boxed{\gamma}\boxed{\max Q'(s',a')} - \boxed{Q(s,a)}]$$

- 🟥 New Q Value for that state and the action
- ⬛ Learning Rate
- 🟫 Reward for taking that action at that state
- 🟪 Current Q Values
- 🟩 Maximum expected future reward given the new state (s') and all possible actions at that new state.
- 🟧 Discount Rate

$$a_{next} = arg \max_{a'} f(Q(s', a'), N(s', a'))$$

References

- https://towardsdatascience.com/explaining-reinforcement-learning-active-vs-passive-a389f41e7195
- https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292
- https://www.simplilearn.com/tutorials/machine-learning-tutorial/reinforcement-learning
- https://www.javatpoint.com/reinforcement-learning

Thank You