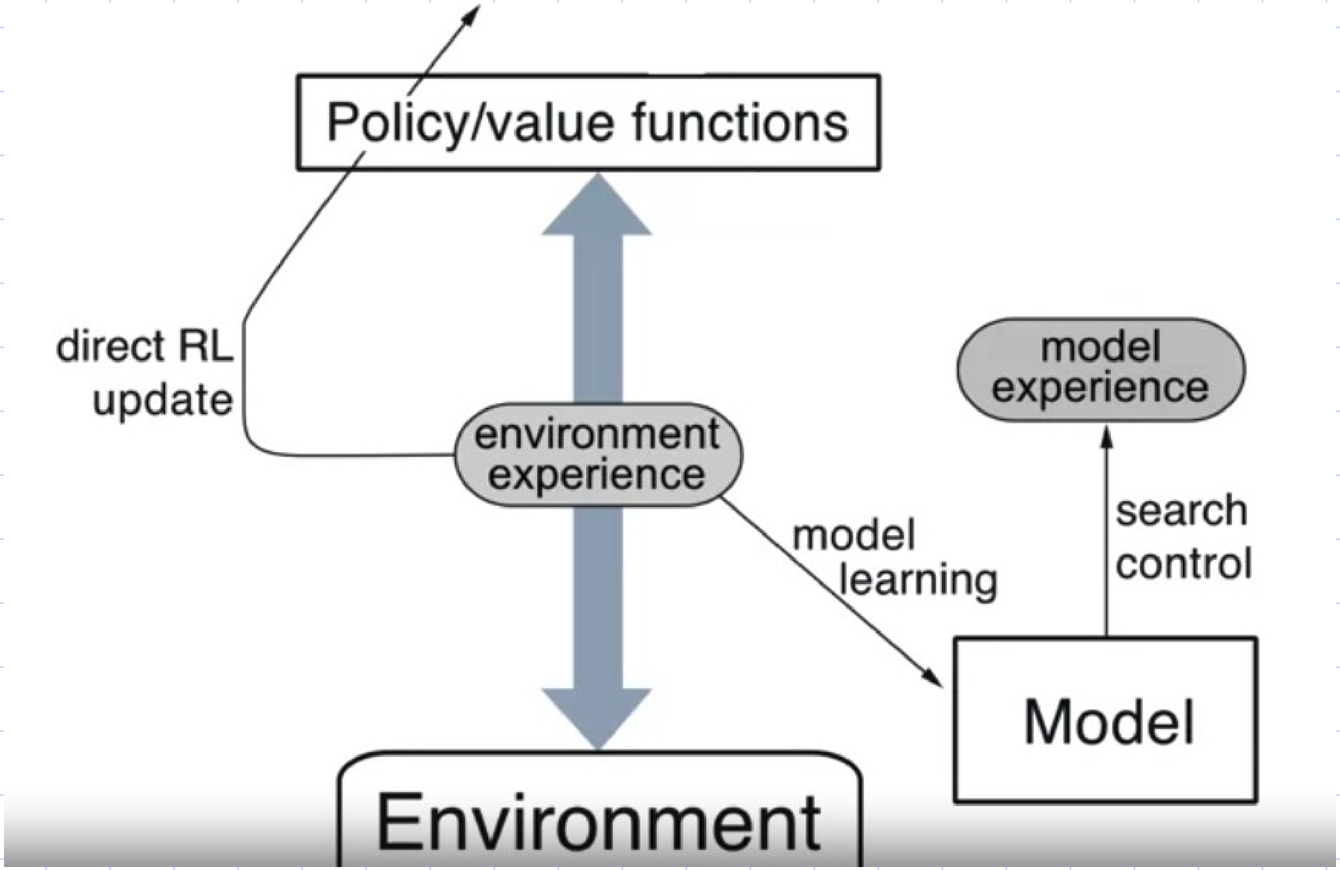
Dyna as a formalism for planning

Objectives

- Describe the Dyna architecture
- Describe the Tabular Dyna-Q algorithm
- Compare Dyna and Q-Learning

The Dyna Architecture



The Dyna Architecture

- Components:
  - Environment and policy: generate a experience.
  - Use experience to perform direct RL updates.
  - Model to do planning, .
  - The environment experience: learn the model. This model will be used to generate model experience.
  - Search Control: control how the model generates this simulated experience, what states the agent will plan from.
  - Planning updates are performed using the experience generated by the model.

The Dyna Architecture

- The Dyna architecture is a hybrid approach in reinforcement learning that combines model-based planning with model-free learning.
- It was introduced by Richard S. Sutton in 1990 as a way to leverage both the advantages of model-based planning and model-free learning to improve decision-making in reinforcement learning tasks.

Dyna Architecture-Components

- Model Learning:
  - The agent learns a model of the environment, which captures the dynamics of state transitions and the associated rewards.
- Planning:
  - Using the learned model, the agent performs planning by simulating possible trajectories of state-action pairs.

Dyna Architecture-Components

- Model-Free Learning:
  - Model-free learning allows the agent to learn directly from experience, updating its value estimates based on observed rewards and transitions.
- Integration:
  - The Dyna architecture integrates planning and model-free learning in a seamless manner.

Dyna Architecture-Components

- Experience Replay:
  - To enhance learning efficiency, the Dyna architecture often employs experience replay, where past experiences (both real and simulated) are stored in a replay buffer and sampled randomly for learning updates. This helps the agent to learn from a diverse set of experiences and avoid the issue of correlated updates.

Dyna Architecture- Advantages

- Improved Sample Efficiency:
  - Planning allows the agent to explore potential future trajectories without actually interacting with the environment, reducing the need for extensive exploration.
- Better Generalization:
  - Model-based planning enables the agent to generalize knowledge beyond its immediate experiences, leading to more robust decision-making in novel situations.

Dyna Architecture-Advantages

- Faster Learning:
  - By leveraging both model-based and model-free approaches, the agent can learn more efficiently and adapt to changes in the environment more quickly.

Dyna Algorithm

- The key steps of the Dyna algorithm between planning and real experience.
- During planning, the agent uses its learned model to simulate future states and rewards, and then updates its value estimates based on these simulations.
- This allows the agent to explore potential future trajectories without actually interacting with the real environment.

Dyna Algorithm

- Step 1: Initialize: Initialize the Q-values for all state-action pairs and optionally initialize the model of the environment.

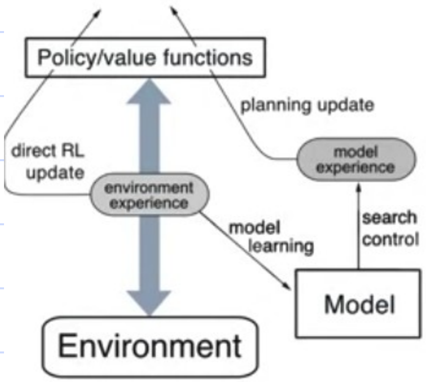
Dyna Algorithm

- Step 2: Loop:
  - Model Learning: If the model of the environment is not provided, learn the model from experience by observing state transitions and rewards.
  - Planning:
    - Sample a state-action pair from the agent's experience.
    - Use the model to simulate the next state and reward given the sampled state-action pair.
    - Update the Q-values based on the simulated experience using a model-free learning algorithm.
    - Repeat planning steps for a certain number of iterations or until convergence.

Dyna Algorithm

- Step 2: Loop:
  - Model Learning: .....
  - Planning:.....
  - Real Experience: Interact with the real environment to collect new experiences.
  - Model Update: If the model is learned from experience, update the model using the new real experiences.
- Step 3: Repeat the loop for a certain number of episodes or until convergence.

Dyna Algorithm



Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:

- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S, Q)
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
- (e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$

Dyna vs Q-learning Algorithm

- Dyna and Q-learning offer different approaches to reinforcement learning.
- Dyna combines model-based planning with model-free learning, potentially leading to more efficient and stable learning, while Q-learning learns directly from experience and can be simpler to implement.

Dyna vs Q-learning Algorithm

- Model Learning:
  - **Dyna**: In Dyna, the agent explicitly learns a model of the environment, including transition dynamics and immediate rewards. This model is then used for planning.
  - **Q-learning**: Q-learning does not explicitly learn a model of the environment. Instead, it learns directly from experience by updating Q-values based on observed state transitions and rewards.

Dyna vs Q-learning Algorithm

- Planning:
  - **Dyna**: Dyna incorporates planning by using the learned model to simulate future trajectories of state-action pairs. These simulated trajectories are used to update value estimates through model-free learning.
  - **Q-learning**: Q-learning does not involve planning. It selects actions based on the current policy and updates Q-values directly based on observed rewards and transitions.

Dyna vs Q-learning Algorithm

- Exploration-Exploitation:
  - **Dyna**: Dyna often explores the environment through its planning process. By simulating future trajectories, it can explore potential actions without taking them in the real environment.
  - **Q-learning**: Q-learning typically relies on exploration strategies, such as ε-greedy, to explore the environment. It balances exploration and exploitation by occasionally choosing random actions.

Dyna vs Q-learning Algorithm

- Sample Efficiency:
  - **Dyna**: Dyna can be more sample-efficient than Q-learning in some cases, especially when planning can lead to better decision-making without additional real experiences.
  - **Q-learning**: Q-learning learns directly from experience, which can require more samples to converge to an optimal policy, especially in complex environments.

Dyna vs Q-learning Algorithm

- Stability:
  - **Dyna**: Dyna updates its value estimates based on both real experiences and simulated trajectories, potentially leading to more stable learning.
  - **Q-learning**: Q-learning updates its Q-values based solely on observed experiences. While this can lead to stable learning, it may also result in more volatile updates, especially in environments with high variance.

Dyna vs Q-learning Algorithm

- Complexity:
  - **Dyna**: Dyna is often more complex to implement due to the additional step of learning a model and incorporating planning.
  - **Q-learning**: Q-learning is relatively straightforward to implement and understand, as it directly learns from experience without the need for a learned model or planning.

Summary

- Describe the Dyna architecture
- Describe the Tabular Dyna-Q algorithm
- Compare Dyna and Q-Learning

FPT Education

FPT UNIVERSITY

Q & A