Reinforcement Learning

AI+X
INSTITUTE FOR ARTIFICIAL INTELLIGENCE

Forms of learning

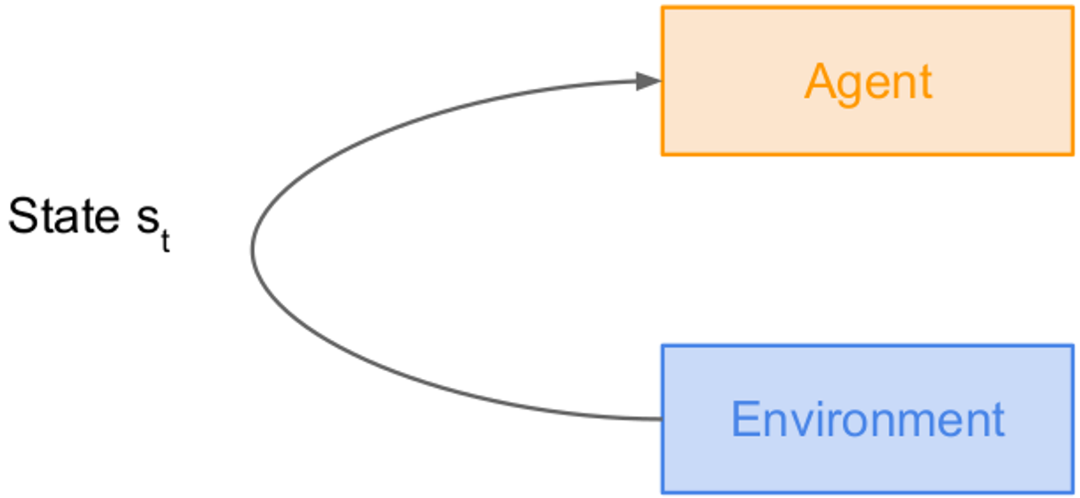
- Supervised learning
    - agent observes input-output pairs and learns a function that maps from input to output
- Unsupervised learning
    - agent learns patterns in the input without any explicit feedback
- **Reinforcement learning**
    - **agent learns from a series of rewards and punishments**

Reinforcement Learning

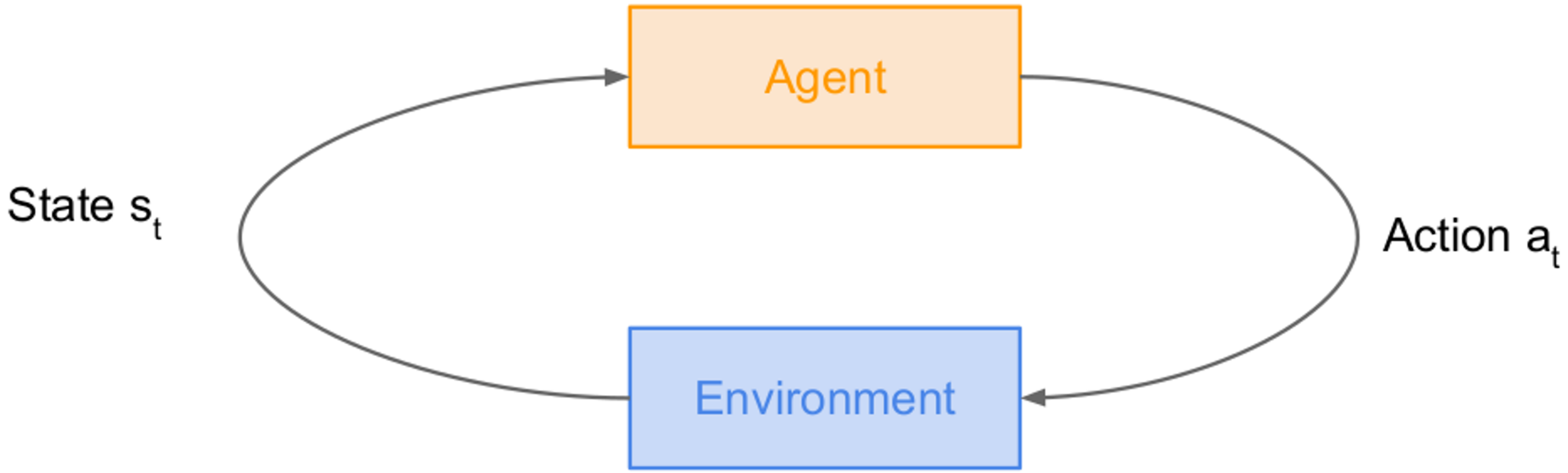
- Why reinforcement learning is different from supervised/unsupervised learning?
  - no supervision, just a reward signal
  - time matters (data is sequential and not independent)
    - feedback is delayed, not instantaneous
    - actions affect the subsequent states

Reinforcement Learning

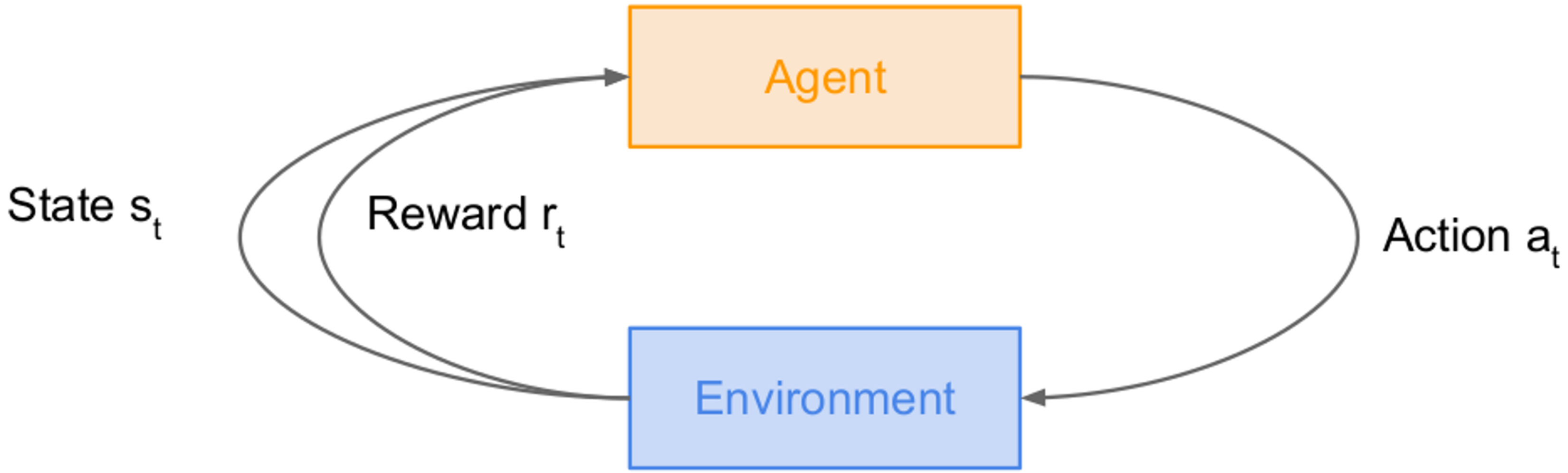
Reinforcement Learning



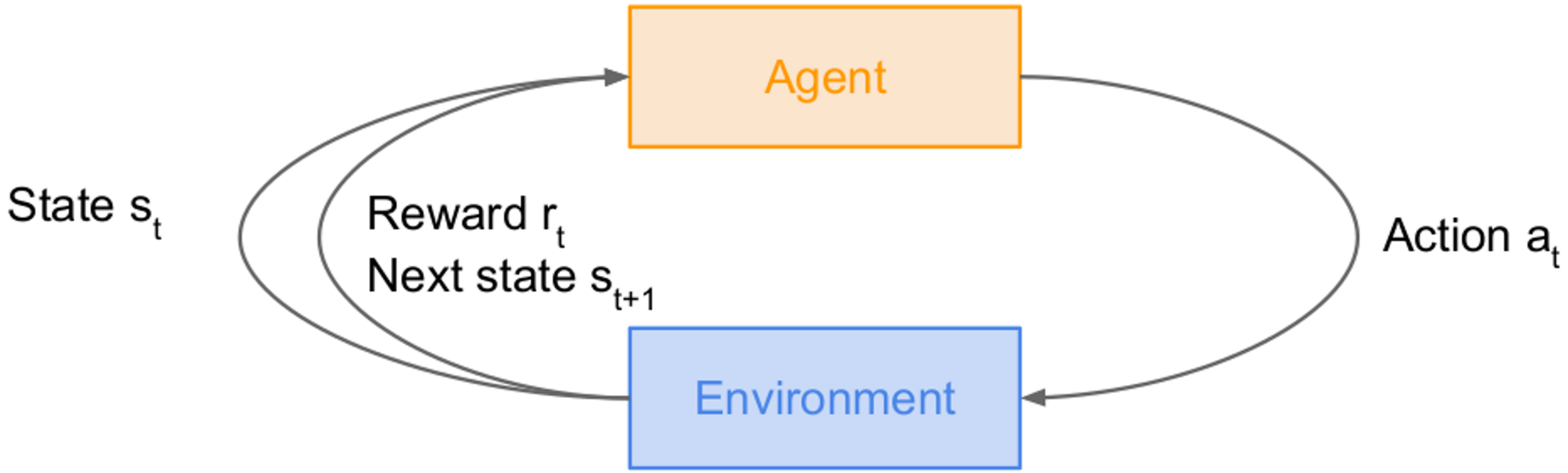
Reinforcement Learning



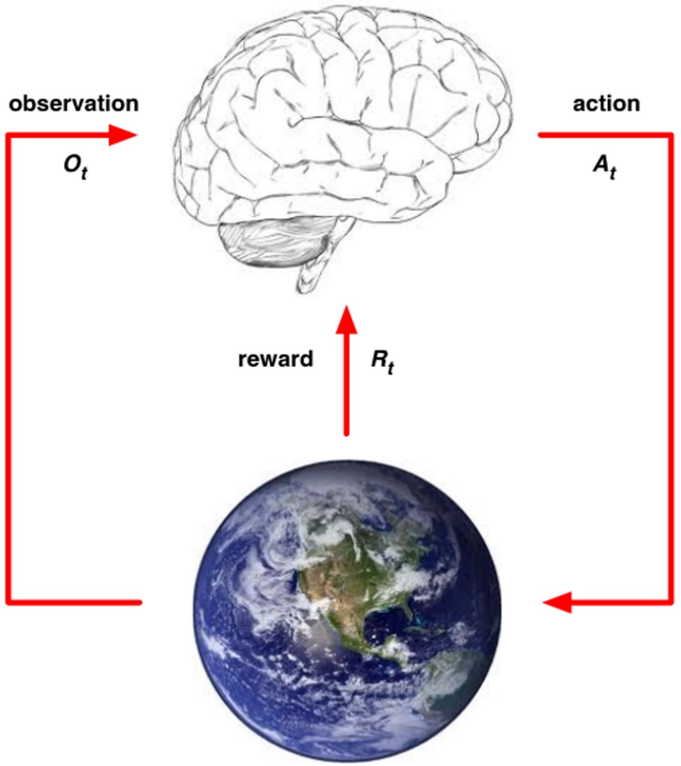
Reinforcement Learning



Reinforcement Learning



Reinforcement Learning



Knowledge Check 1

Why is reinforcement learning considered different than supervised learning?

A. Unlike supervised learning, reinforcement learning does not need any kind of feedback.

B. While supervised learning requires the correct answer for training, reinforcement learning uses a mix of correct and incorrect answers.

C. Unlike supervised learning, reinforcement learning does not need to know the answer during training, but requires some feedback in the form of a reward for its decisions instead.

Reinforcement Learning

- When can we use reinforcement learning?
  - Whenever we can describe our goal in terms of a reward function

Reinforcement Learning

- Fly a helicopter
  - positive reward for reaching target location
  - negative reward for crashing
- Play chess
  - positive/negative reward for winning/losing a game
- Manage an investment portfolio
  - positive reward is the money in bank
- Make a humanoid robot walk
  - positive reward for forward motion
  - negative reward for falling over
- Play many different Atari games better than humans
  - positive/negative reward for increasing/decreasing score

2017: OpenAI Five Beats Top Professional Dota 2 Players

Markov Decision Process

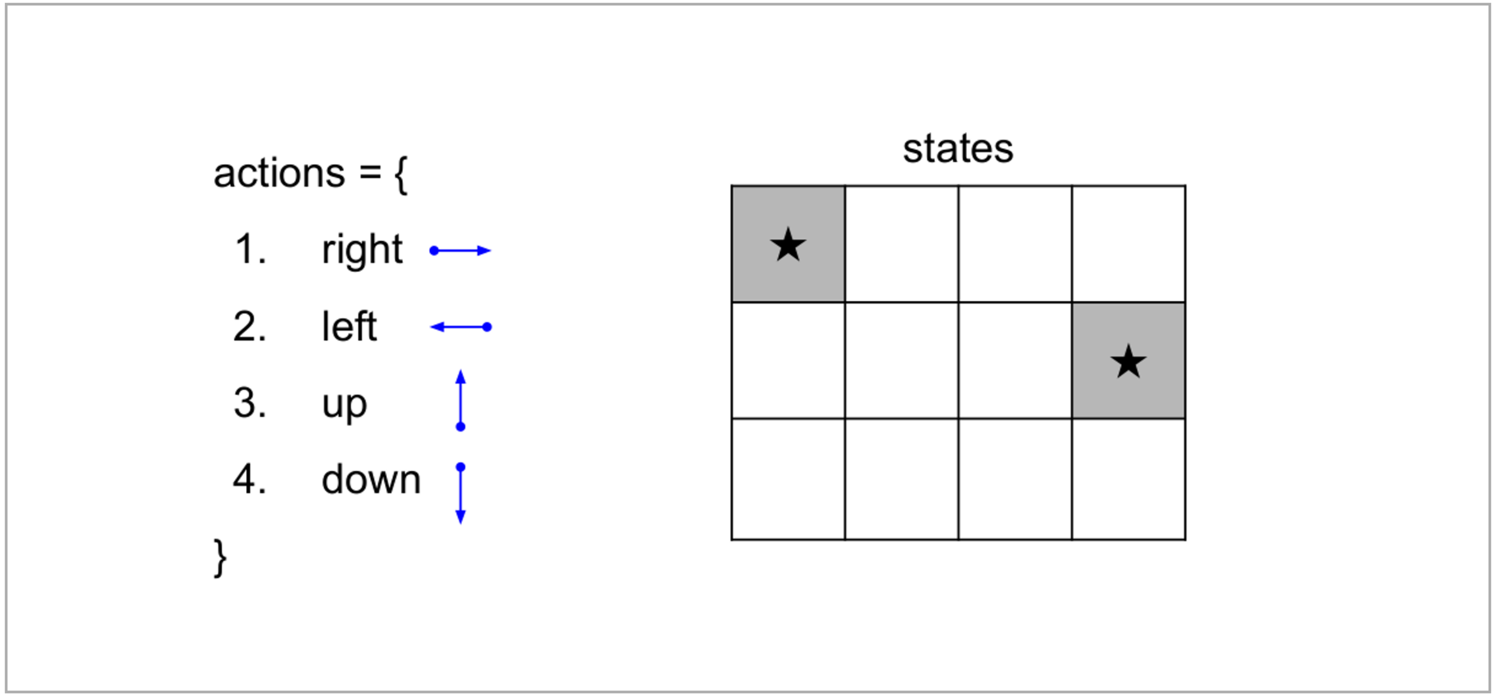
- A Markov process is a random process in which the future is independent of the past, given the present.
  - Mathematical formulation of the RL problem
  - Current state completely characterizes the state of the world
- Defined by:
- set of possible states (S)
- set of possible actions (A)
- distribution of reward given (state, action) pair (R)
- transition probability (P) - distribution over next state given (state, action) pair
- discount factor (γ)

Markov Decision Process

- At time step t=0, environment samples initial state s_0~ p(s_0)
- Then, for t=0 until done:
  - Agent selects action a_t
  - Environment samples reward $r_t \sim R(\,.\,|s_t\,,a_t\,)$
  - Environment samples next state $s_{t+1} \sim P(\,.\,|s_t\,,a_t)$
  - Agent receives reward rt and next state s_{t+1}
- A policy π is a function from S to A that specifies what action to take in each state
- Objective: find policy π* that maximizes cumulative discounted reward
  - $\sum_{t\geq 0} \gamma^t r_t$

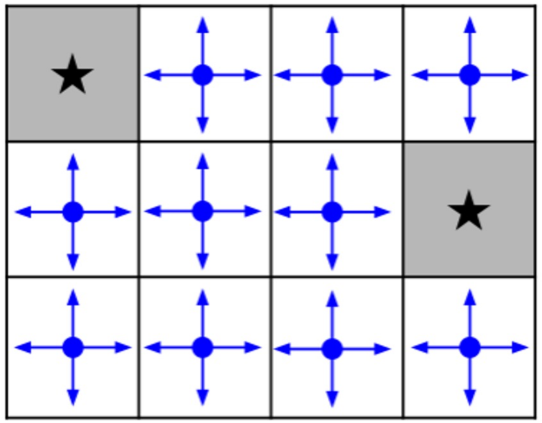
Markov Decision Process

- Goal: reach the stars in the least amount of actions
  - Reward function: negative reward for each transition

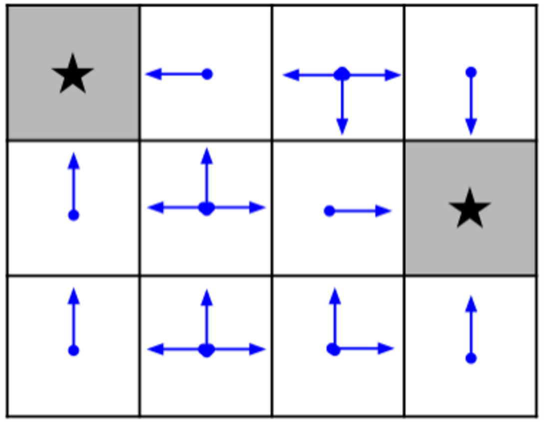


Markov Decision Process

- Goal: reach the stars in the least amount of actions
  - Reward function: negative reward for each transition



Random Policy



Optimal Policy

Q-Learning

- Agent use the environment's rewards to learn the best action to take in each state
- A Q-value for a state-action pair represents the "quality" of an action taken from that state
  - $Q^{\pi}(s, a) = \mathbb{E} [\sum_{t \geq 0} \gamma^t r_t \mid s_0 = s, a_0 = a, \pi]$
  - High Q-values imply greater rewards
- Q-values are initialized to an arbitrary value and are updated with the outcome of environment simulations
  - $Q(s, a) = (1-\alpha)Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a')]$
  - s' is the state reached from s after taking action a
  - α is the learning rate
- Q(s, a) is a large table
  - For huge tables, we can approximate this table with a neural network!

You have reached the end of the lecture.