Stats AI

Reinforcement Learning

Author: Hussain Abbas, MSc

Motivation

- The goal of RL is to find the optimal strategy for a given problem:
  - Should I sell my stock or not?
  - Should the player stay in their position or keep moving?
  - Should we invest in this company or not?
  - Should the drone return to base?
  - Should we hire this person or not?

- The above problems are the same abstract problem:
  - A goal seeking agent operates in an environment consisting of states, taking actions in each state, so as to maximize its long term expected reward

- Rewards can be positive or negative

- Rewards can be delayed

Stats AI

Motivation

- We can formalize the previous problem into an MDP:
    - MDP: Markov Decision Process
    - MDP: consists of state, action, reward, state
- For each state, we have a set of actions we can take:
    - After we take an action, we observe a reward and move to the next state
    - The process continues until we hit the terminating state
    - The terminating state is basically the "end of the road"
- Thus, the optimal strategy consists of taking the best action in each state to maximize the total expected reward
- This optimal strategy in RL jargon has a very specific name: the optimal policy …

Optimal Policy

- The optimal policy is the strategy we want to obtain:
  - For each state, we want to take the best action to maximize the total expected reward
  - We call this “following the optimal policy”
- Thus, we maximize the total expected reward by following the optimal policy
- Since rewards may be delayed, we may have to take actions now for which we observe no immediate reward (study for midterms weeks in advance)
- Strategies which do not follow the optimal policy are sub-optimal, i.e., they obtain a reward lower than what is possible

Optimal Policy

Optimal Policy

In terms of return, a policy π is considered to be better than or the same as policy π' if the expected return of π is greater than or equal to the expected return of π' for all states. In other words,

$$\pi \geq \pi' \text{ if and only if } v_{\pi}(s) \geq v_{\pi'}(s) \text{ for all } s \in \boldsymbol{S}.$$

Remember, $v_{\pi}(s)$ gives the expected return for starting in state s and following π thereafter. A policy that is better than or at least the same as all other policies is called the *optimal policy*.

Optimal State-Value Function

The optimal policy has an associated *optimal* state-value function. Recall, we covered state-value functions in detail last time. We denote the optimal state-value function as v_* and define as

$$v_*(s) = \max_{\pi} v_\pi(s)$$

for all $s \in \boldsymbol{S}$. In other words, v_* gives the largest expected return achievable by any policy π for each state.

Stats AI

Optimal Action-Value Function

Optimal Action-Value Function

Similarly, the optimal policy has an *optimal* action-value function, or *optimal* Q-function, which we denote as q_* and define as

$$q_*(s,a) = \max_{\pi} q_\pi(s,a)$$

for all $s \in \boldsymbol{S}$ and $a \in \boldsymbol{A}(s)$. In other words, q_* gives the largest expected return achievable by any policy π for each possible state-action pair.

Stats AI

Bellman Optimality Equation

One fundamental property of q_* is that it must satisfy the following equation.

$$q_*(s,a) = E\left[R_{t+1} + \gamma \max_{a'} q_*(s',a')\right]$$

This is called the *Bellman optimality equation*. It states that, for any state-action pair (s,a) at time t, the expected return from starting in state s, selecting action a and following the optimal policy thereafter (AKA *the Q-value* of this pair) is going to be the expected reward we get from taking action a in state s, which is R_{t+1}, plus the *maximum* expected discounted return that can be achieved from any possible next state-action pair (s',a').

Since the agent is following an optimal policy, the following state s' will be the state from which the best possible next action a' can be taken at time $t+1$.

Stats AI

Finding the Optimal Policy

- Recall that the optimal strategy is the optimal policy
- We maximize the total expected reward by following the optimal policy
- The question is, "how do we actually find the optimal policy?"
- Recall the following:
  - The optimal policy has an associated optimal action-value function, q*
  - q* gives the largest expected return achievable by any policy for every possible state-action pair
  - q* must satisfy the Bellman equation
- Bottom line:
  - We can use the Bellman equation to find q*
  - We can use q* to find the optimal policy

Stats AI

Enter Q-Learning

- Recall the following:
  - We can use the Bellman equation to find q*
  - We can use q* to find the optimal policy

- Given q*, we can determine the optimal policy by applying an RL algorithm to find the action that maximizes q* for each state

- Q learning is one such algorithm that can be used to solve for the optimal policy in an MDP

- Bottom line:
  - We use Q-learning to find the optimal q-values for each state-action pair
  - Thus, Q-learning is how we “train the ML model”
  - For deployment, the optimal strategy is the optimal policy
  - We maximize the total expected reward by following the optimal policy
  - That is, for each state, we take the action that has the highest q-value

Stats AI