Reinforcement Learning

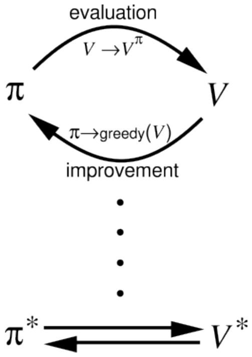
Reinforcement learning is a machine learning paradigm in which an agent is trained to maximize a reward signal by observing many agent-environment interactions. It does not require labelled input/output pairs as does supervised learning. Reinforcement learning is often studied in the context of **Markov decision processes (MDP)** for which the following terminology are important:

- State, S: The environmental variables observable to the agent. In an MDP, the state contains all of the information that affects the environment's interaction with the agent and future state(s).
- Action, A: The set of behaviors available to the agent given the environment's current state.
- Reward, R: The signal associated with an agent's action and its effect on the environment.
- Policy, $\pi(S)$: A function that maps an action to each state. An agent is controlled by a policy.
- Value function, $v_\pi(S)$: The (possibly discounted) sum of reward signals received by an agent that starts in a state S and follows policy π.
- Action-Value function, $Q_\pi(S, A)$: The (possibly discounted) sum of reward signals received by an agent that starts in a state S, immediately takes action A, and thereafter follows policy π.

The goal of reinforcement learning is to derive an optimal policy, for which the value of each state is maximal. This guarantees a maximal reward signal. The optimal policy is usually computed iteratively though **generalized policy iteration (GPI)** which repeatedly computes and then uses the (action-)value function for a policy to improve it until converging (simultaneously) on optimal V(s)/Q(S,A) and $\pi(S)$.

Implementations of GPI can be divided into **on-policy** and **off-policy** approaches. On-policy methods improve the same policy which is being used to control the agent, while off-policy methods improve a **target policy** based on observations of a separate **behavior policy** which is used to control the agent.

Off-policy methods can be used to ensure that every state-action pair is evaluated. This is typically achieved by using an ε-soft behavior policy, wherein an action other than the one specified by the policy is selected with probability ε.



Some common historical approaches to reinforcement learning are as follows:

Dynamic Programming

- Iterative approach to solving the Bellman equation for the optimal (action-)value function.
- Requires knowledge of the environment's states, actions, rewards and transition probabilities.

Monte Carlo Methods

- Iterative approach to approximating the optimal (action-)value function.
- Updates occur only at the end of an episode, using the sum of all rewards as the target.

Temporal Difference Methods

- Iterative approach to approximating the optimal (action-)value function.
- Updates occur within episodes, using one or more rewards summed with the (action-)value of the resulting state(-action) as the target. This is a **bootstrapping** method.