Reinforcement Learning
Summary

Yin Li

yin.li@wisc.edu

University of Wisconsin, Madison

Key concepts of (deep) neural networks

- Reinforcement learning task
- Markov decision process
- Value functions & Bellman equation
- Value iteration

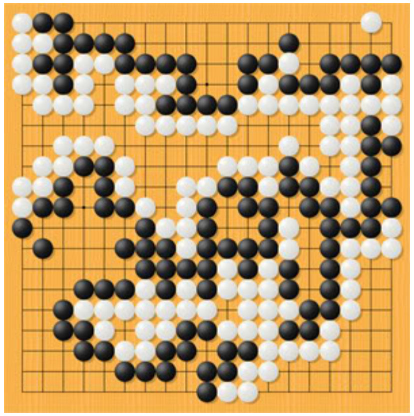
Reinforcement Learning (RL)

Task of an agent embedded in an environment

repeat forever

1) sense world
2) reason
3) choose an action to perform
4) get feedback (usually reward = 0)
5) learn

the environment may be the physical world or an artificial one



Formalism: Markov Decision Processes

- **States** S, beginning with initial state s_0
- **Actions** A
- **Transition model** $P(s_{t+1} | s_t, a_t)$
  - *Markov assumption*: the probability of going to s_{t+1} from s_t depends only on s_t and a_t and not on any other past actions or states
- **Reward function** $r(s_t)$

- **Policy** $\pi(s) : S \rightarrow A$ the action that an agent takes in any given state
  - The "solution" to an MDP

Defining the optimal policy

- Given a policy π, we can define the *expected utility* over all possible state sequences from s_0 produced by following that policy:

$$V^{\pi}(s_0) = \sum_{\substack{\text{sequences} \\ \text{starting from } s_0}} P(sequence)U(sequence)$$

- The value function of s_0 w.r.t. policy π
- The utility of a state sequence is defined as the sum of discounted rewards
- The optimal policy should maximize this utility

Discounted rewards

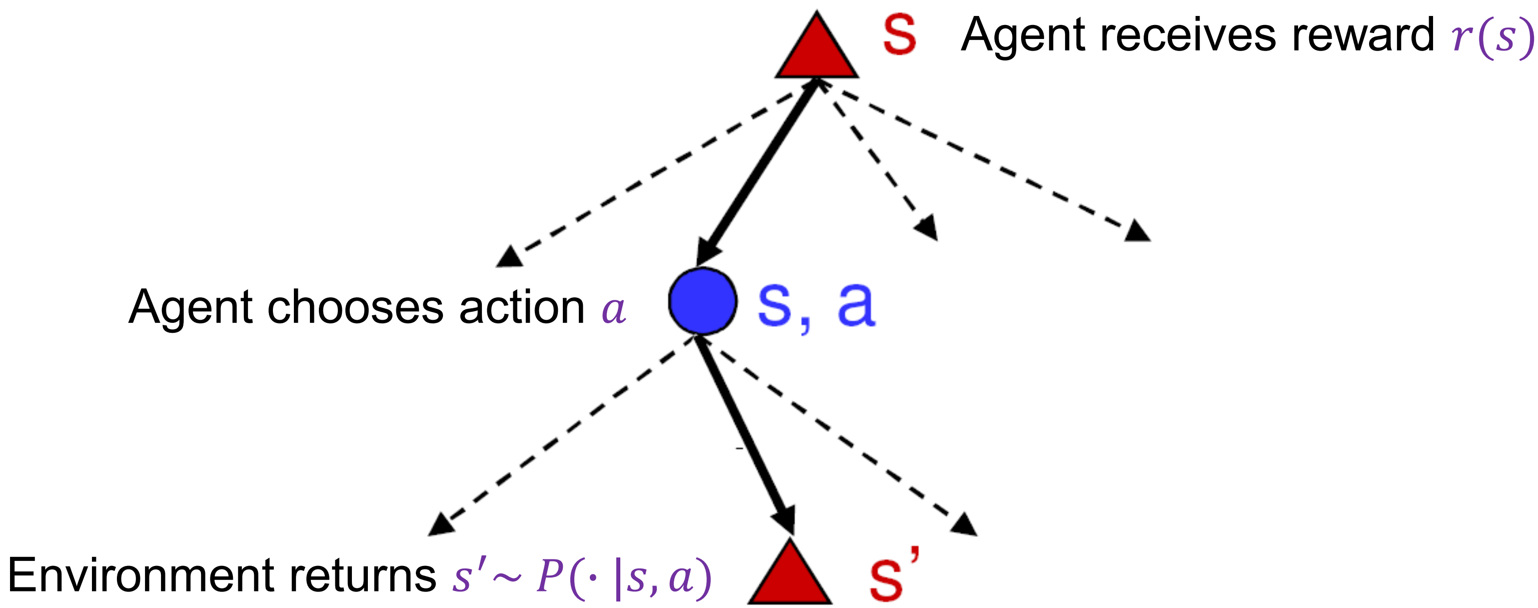
- To define the utility of a state sequence, *discount* the individual state rewards by a factor γ between 0 and 1

$$U(s_0, s_1, \ldots) = r(s_0) + \gamma\, r(s_1) + \gamma^2 r(s_2) + \cdots$$

$$= \sum_{t \geq 0} \gamma^t\, r(s_t)$$

Image source: P. Abbeel and D. Klein

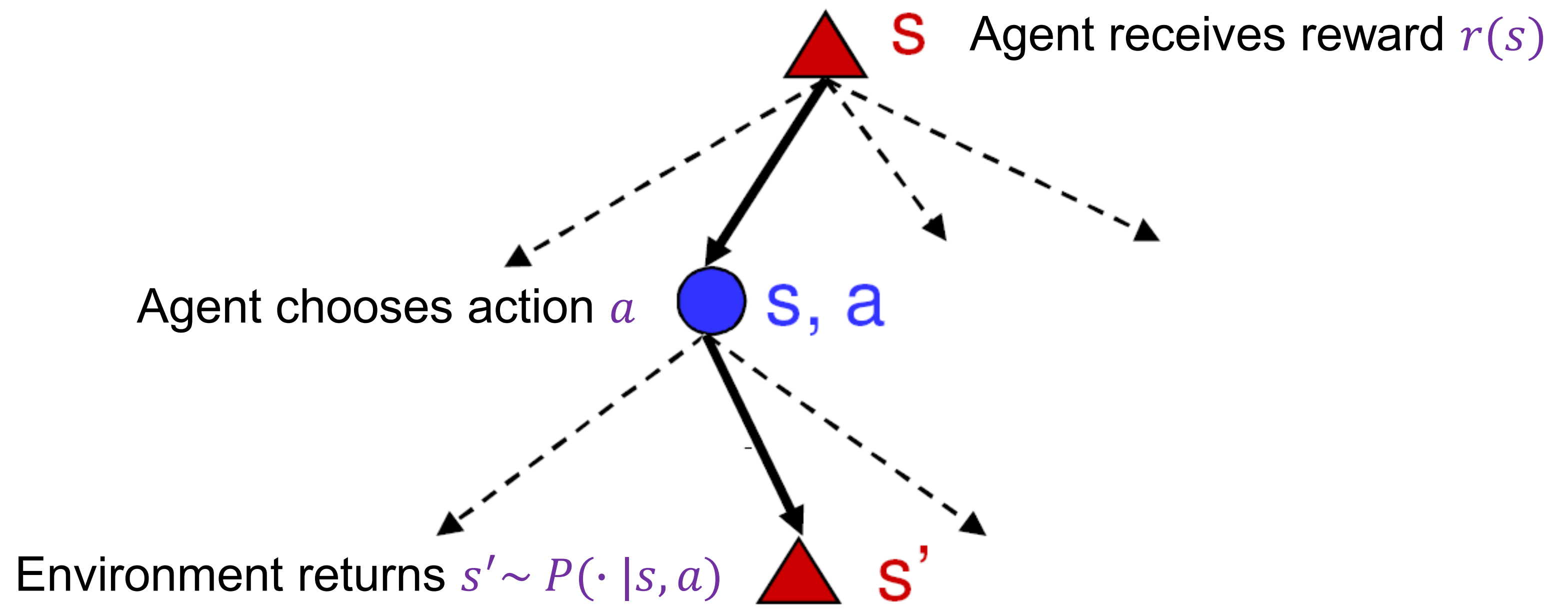
The Bellman equation



- Define state utility $V^*(s)$ as the expected sum of discounted rewards if the agent executes an *optimal* policy starting in state s

Image source: L. Lazbenik

The Bellman equation



- What is the recursive expression for $V^*(s)$ in terms of $V^*(s')$ - the utilities of its successors?

$$V^*(s) = r(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^*(s')$$

Image source: L. Lazbenik

The Bellman equation

- Recursive relationship between optimal values of successive states:

$$V^*(s) = r(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^*(s')$$

- The best policy to the MDP from s_0 is given by $V^*(s)$
- The solution is

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s,a) V^*(s')$$

Value iteration

- Start out with every $V_0(s)$ = 0
- Iterate until convergence
  - During the *i*th iteration, update the utility of each state according to the equation:

$$V_{i+1}(s) = r(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_i(s')$$

- With infinitely many iterations, guaranteed to find the correct utility values $V^*(s)$
  - Even if we randomly traverse environment instead of looping through each state and action
  - In practice, don't need infinitely many iterations…