COMP 532
Machine Learning and BioInspired Optimization

Lecture 8 Reinforcement Learning

Dr. Shan Luo

Department of Computer Science

shan.luo@liverpool.ac.uk

Questions

- What can *I* tell you about RL?
- What is common to all three classes of methods? – DP, MC, TD
- What are the principle strengths and weaknesses of each?
- In what sense is our RL view complete?
- In what senses is it incomplete?
  - What are the principal things missing?
- The broad applicability of these ideas…
- What does the term bootstrapping refer to?
- What is the relationship between DP and learning?

Today's Lecture

- Yesterday, we discussed *one-step tabular model-free* TD methods

- Today, we will look at methods that are:
  - multi-step,
  - approximate, and
  - model-based

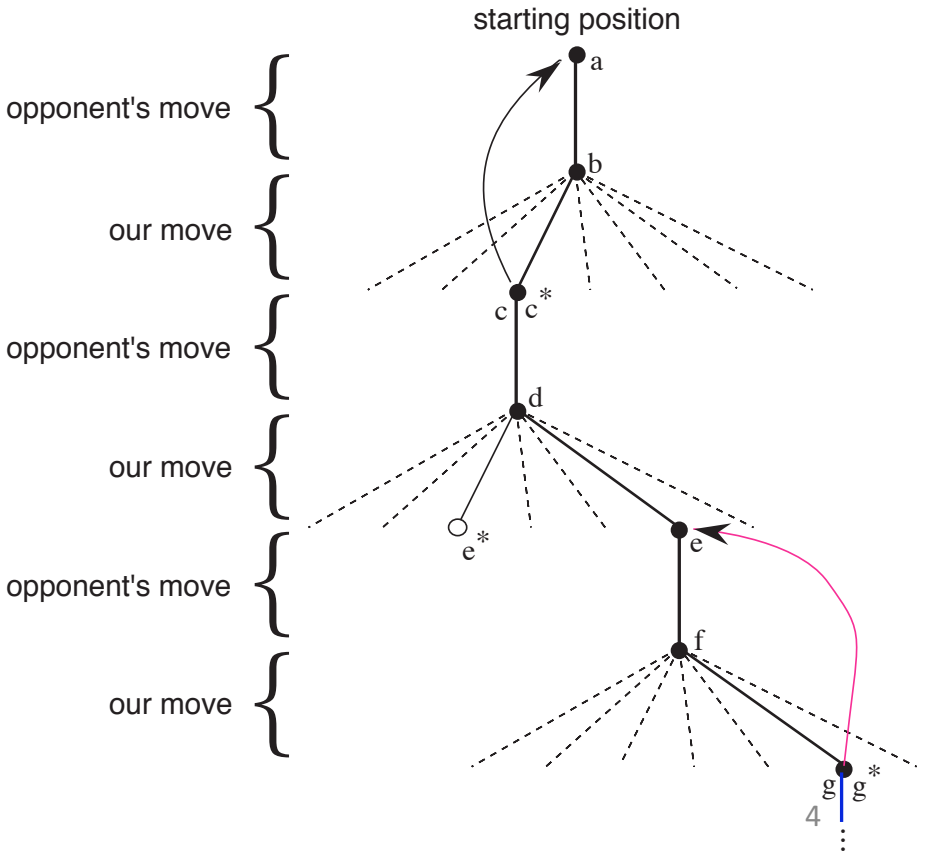
- But first, some additional "advanced" topics

Afterstates

- Usually, a state-value function evaluates states in which the agent can take an action.

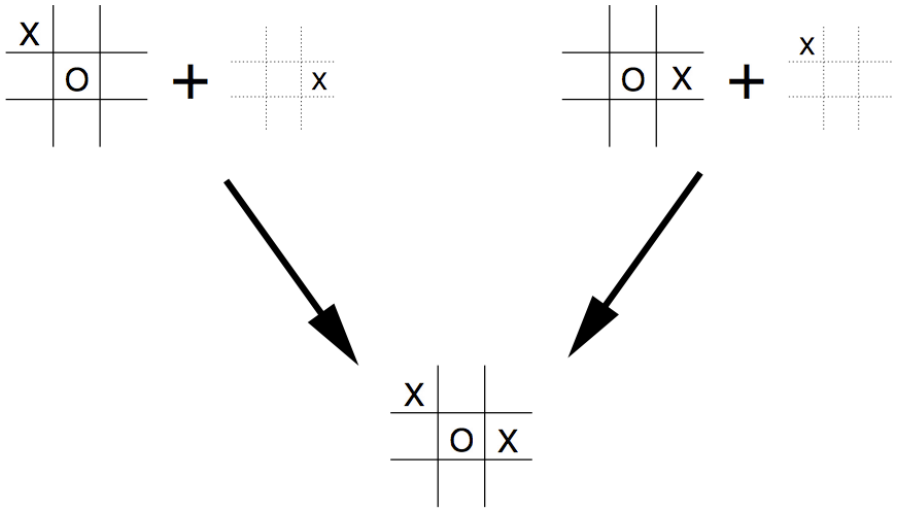
- Sometimes it is useful to evaluate states after the agent has acted.

- Remember tic-tac-toe:



Afterstates

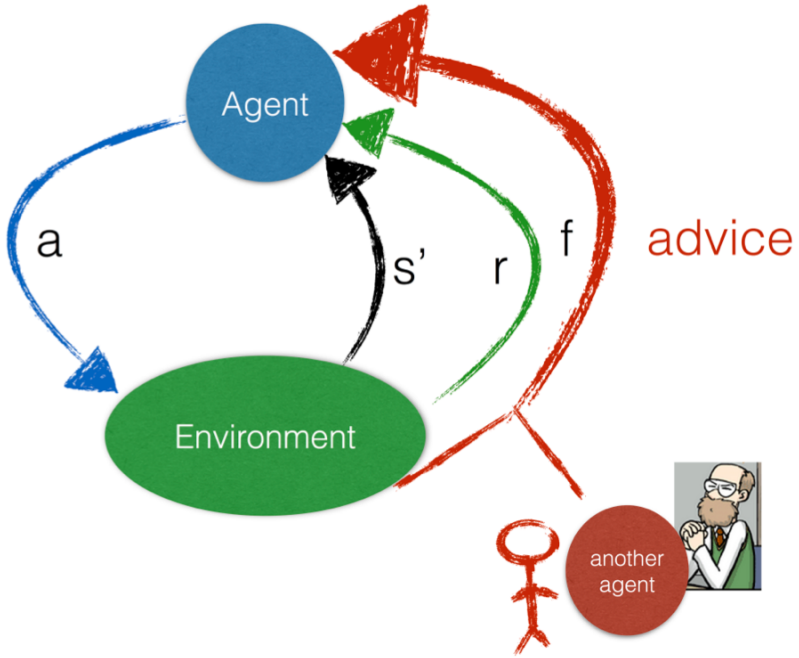
- Why is this useful?



- What is this in general?

Reward Shaping

- Incorporate domain knowledge to provide additional rewards during an episode
- Guide the agent to learn faster
- (Optimal) policies preserved give a potential-based shaping function



Reward Shaping

Q-learning:

$$\mathrm{Q}(s_t, a_t) \leftarrow \mathrm{Q}(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a \mathrm{Q}(s_{t+1}, a) - \mathrm{Q}(s_t, a_t) \right]$$

Reward shaping :

- Provide heuristic knowledge by an additional reward

$$\mathrm{Q}(s_t, a_t) \leftarrow \mathrm{Q}(s_t, a_t) + \alpha \left[r_{t+1} + F(s, s') + \gamma \max_a \mathrm{Q}(s_{t+1}, a) - \mathrm{Q}(s_t, a_t) \right]$$

- Potential-based reward shaping:

$F(s, s') = \gamma(\Phi(s') - \Phi(\mathrm{s})$

$\mathrm{F}(\mathrm{s}, \mathrm{s}')$: Additional reward when moving from state s to s'

γ: Discount factor

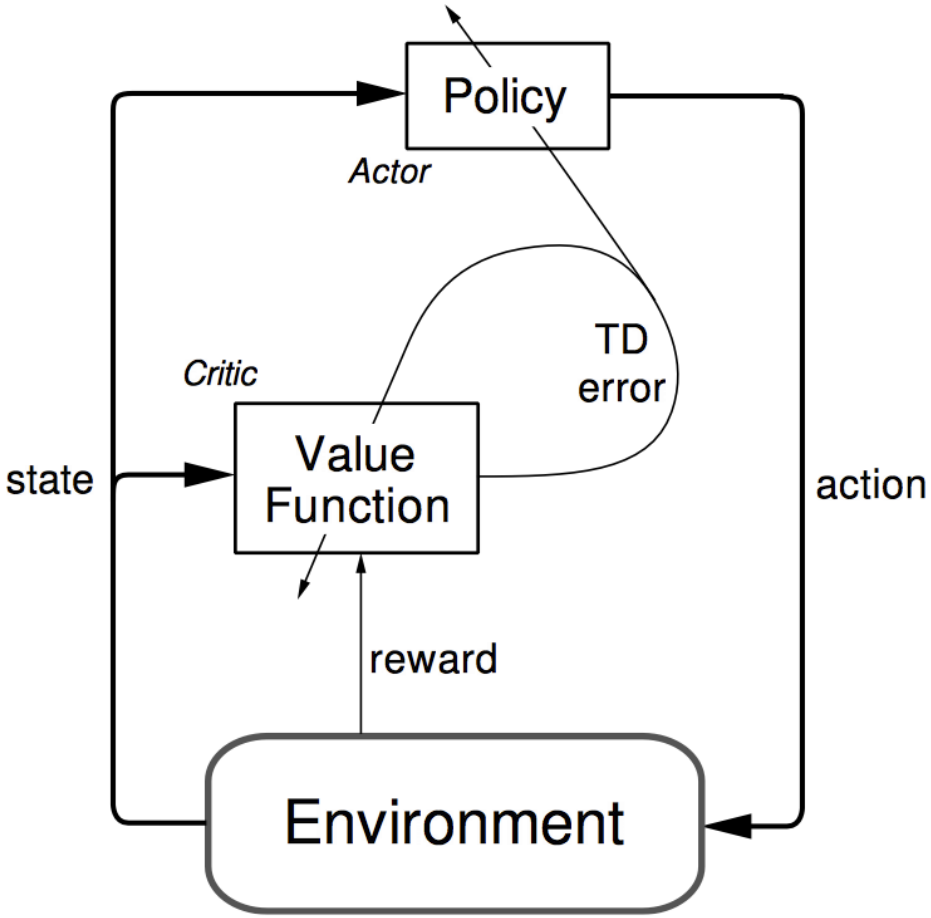
$\Phi(s)$: Potential of state s

Ng *et al.* "Policy Invariance Under Reward Transformations: Theory And Application To Reward Shaping." ICML, 1999.

Learning Without Value Functions

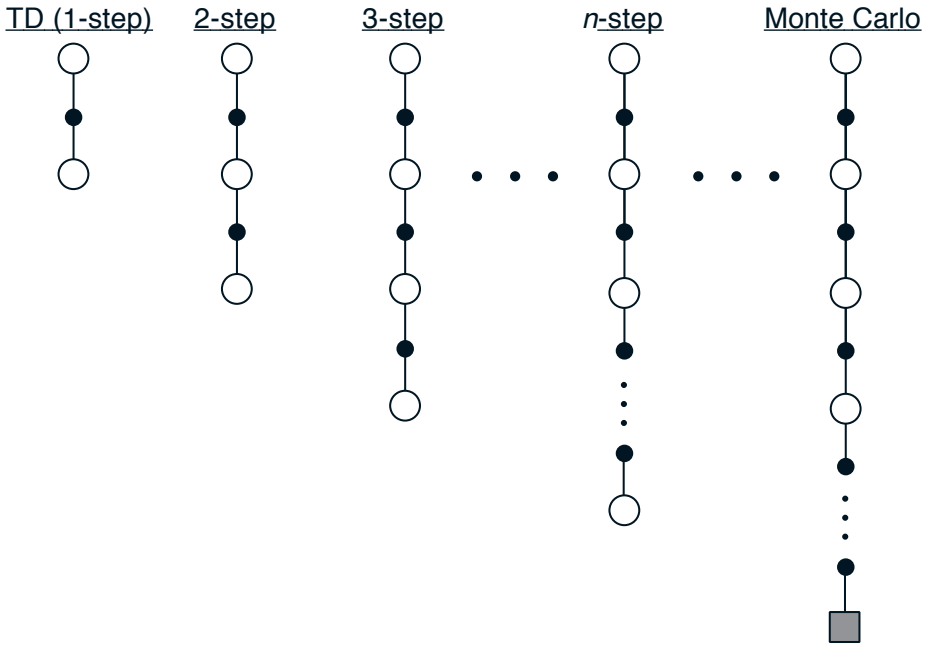
- Common among all methods we have discussed:
  - DP, MC, TD (Sarsa, Q-learning, R-learning)
  - They all use value functions!
- Policy π is based on value functions
- We can also learn by directly manipulating the policy instead of first looking for the optimal value function
- Examples: learning automata, evolutionary algorithms

Actor-Critic Methods



Multi-Step TD

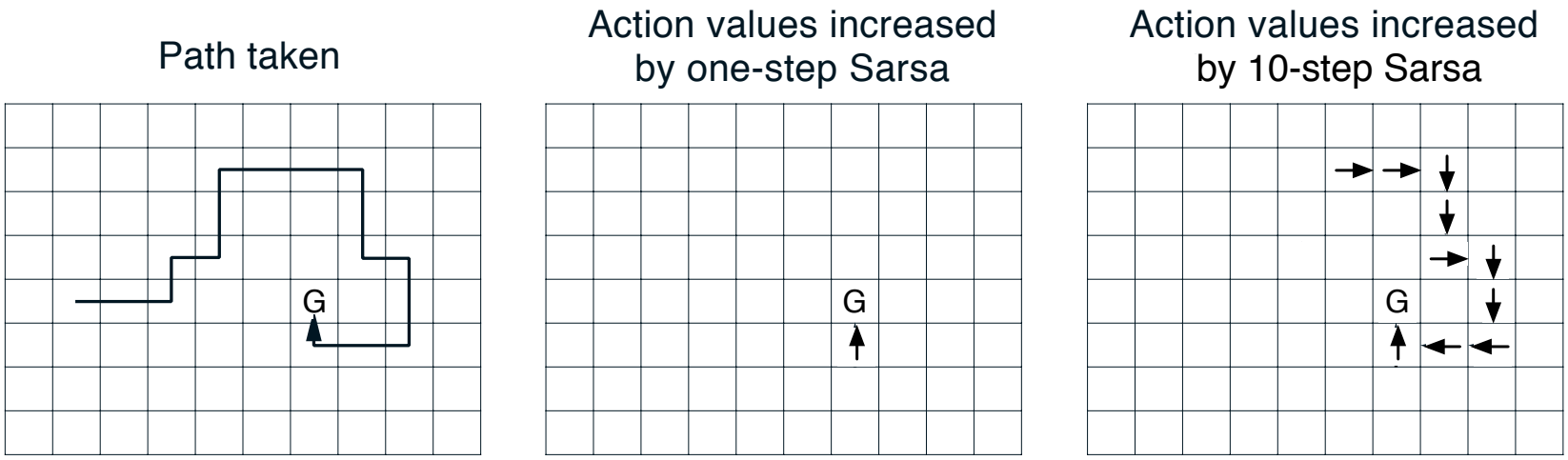
- So far we have looked at TD methods that back-up values after each single time step.
- In general, we could back up values over a longer range of steps.



Multi-Step TD

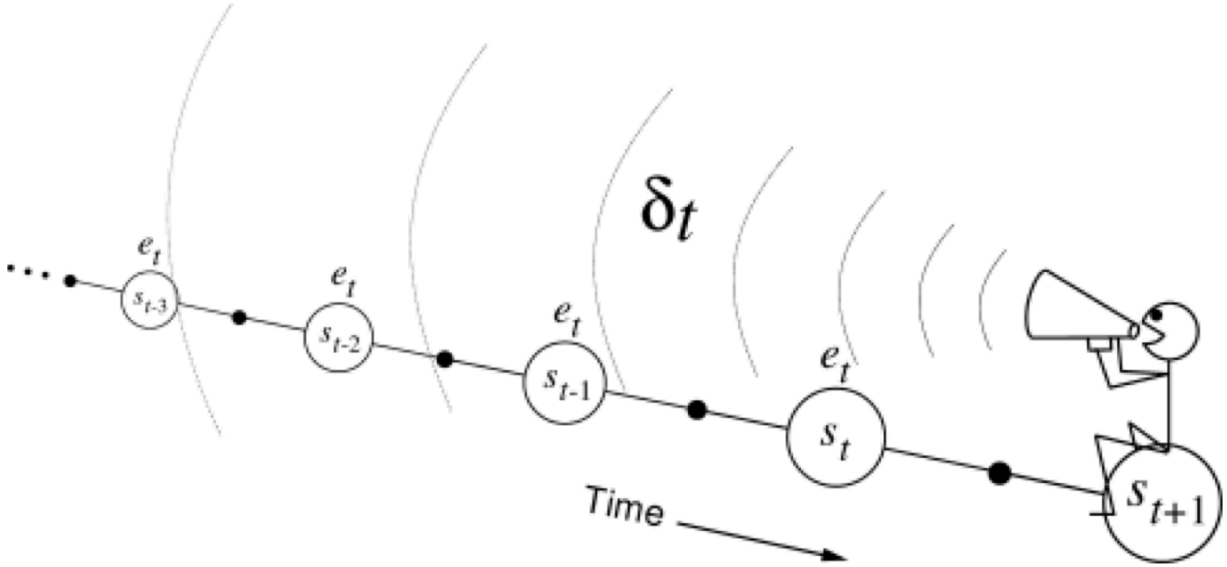
Can be applied for control as well, e.g.

n-step Sarsa



Eligibility Traces

- Most famous implementation of *n*-step TD
- State-action pairs are eligible for future rewards, with more recent states getting more credit
  - Keep a trace or history of state-action pairs
  - Trace decays over time



Function Approximation

- So far we have used a tabular notation for value functions
- For large state and actions spaces this approach becomes intractable
- Function approximators can be used to generalize over large or even continuous state and action spaces

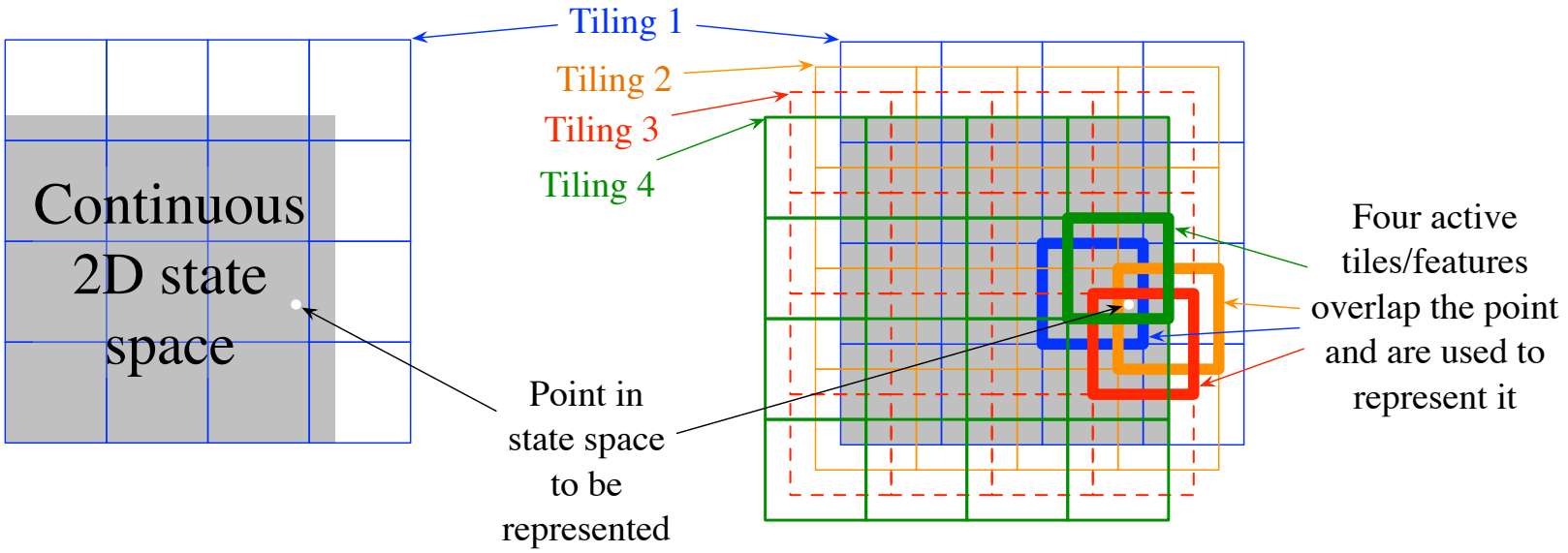
Examples:

- Tile coding
- Gaussian processes
- Neural networks

Function Approximation

For example, Tile Coding

- Uses overlapping tilings to represent a continuous multi-dimensional space
- Each state is represented by the combination of tiles that is active



Learning and Planning: Model-based TD

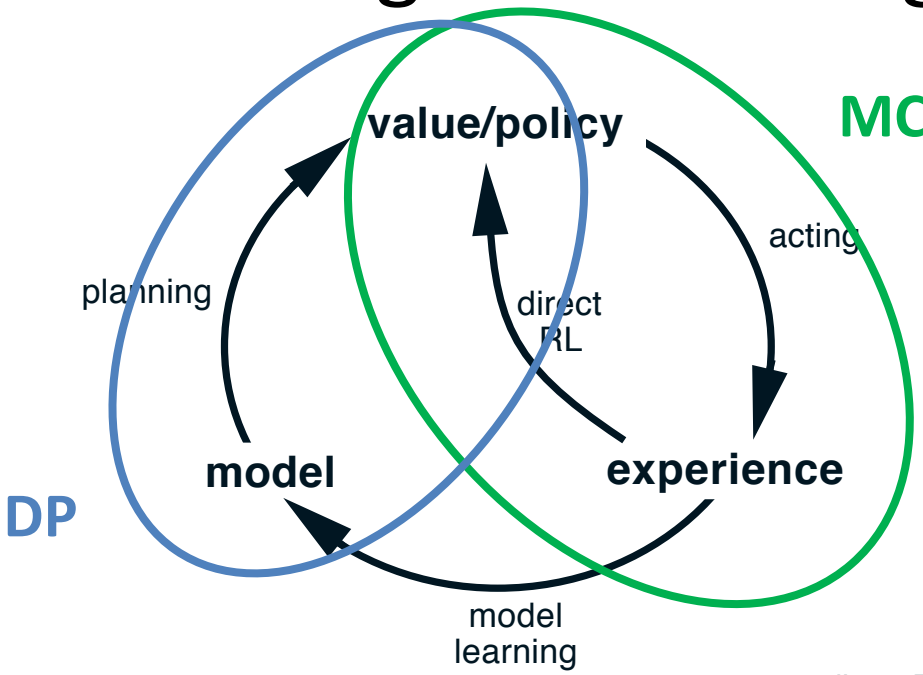
- Planning: any computational method that takes a model as input and produces or improves a policy for interacting with the modelled environment

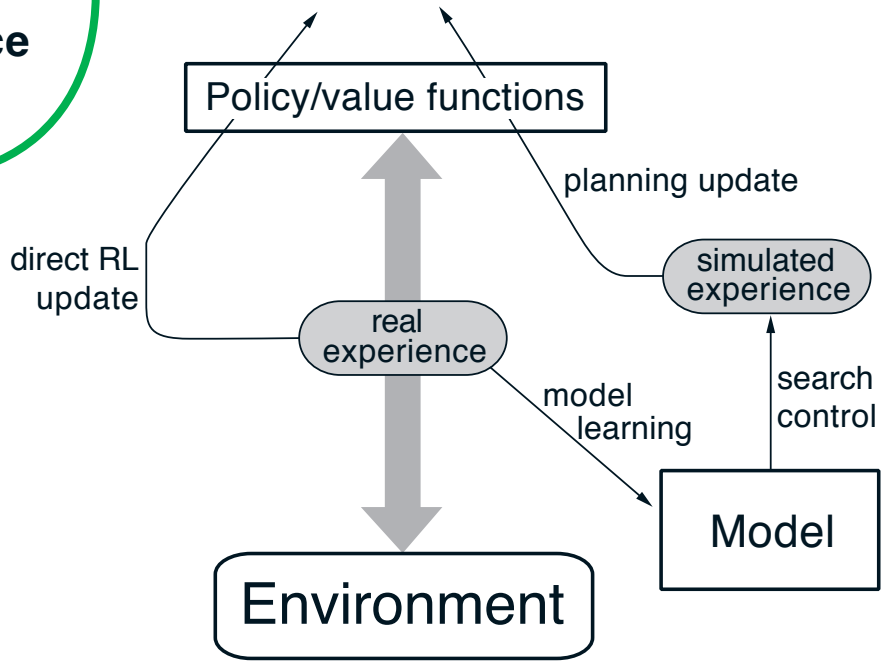
- Dynamic Programming is a planning method

- Monte Carlo and TD methods are not

- But: many commonalities! Can we combine both?

Learning and Planning: Model-based TD





Dyna-Q framework

Wrap-up

- We discussed different learning architectures:
  - Based on advice
  - Actor-critic
  - Direct policy learners
  - Multi-step TD & Eligibility Traces
  - Function approximation
  - Model learning & Planning

- Of course, there is more…
  - Deep learning (coming up next!)
  - Multi-agent learning (in March)