LG Advanced Data Scientists Program
Deep Learning

[8: Introduction to Reinforcement Learning]

Prof. Sungroh Yoon

Electrical & Computer Engineering | Seoul National University

(last compiled at 21:16:00 on 2020/02/26)

Outline

References

- books/papers:
  - Reinforcement Learning (2nd edition)[1] ▸ Link
  - Artificial Intelligence: A Modern Approach[2]
  - A brief survey of deep reinforcement learning[3]

- online resources:
  - Silver UCL class ▸ Link & ICML tutorial ▸ Link
  - Schulman MLSS tutorial ▸ Link
  - Abbeel & Schulman NIPS tutorial ▸ Link
  - UC Berkeley CS188 (AI) ▸ Link & CS294 (DRL) ▸ Link
  - Stanford CS234 (RL) ▸ Link

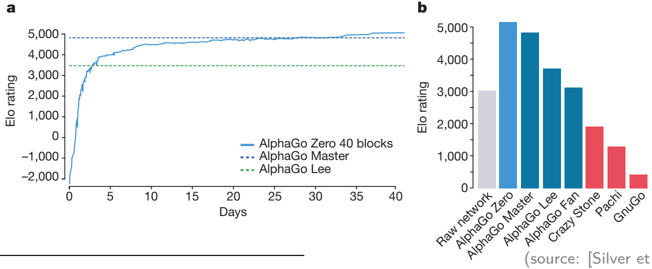
---

[1] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction.* MIT press

[2] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach.* Pearson Education Limited

[3] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*

Outline

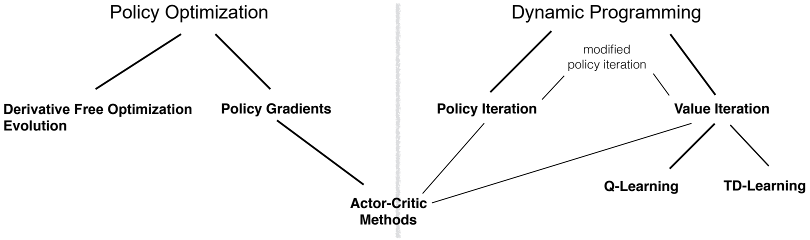


(source: [Silver et al., 2017][4])

[4] Silver, D. et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354

Reinforcement learning in a nutshell (Silver, 2016)

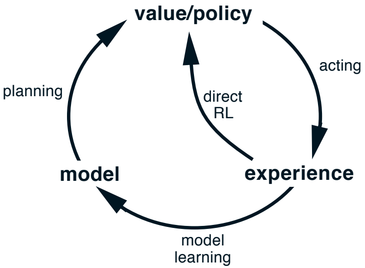
- RL: a general-purpose framework for ____________
  - RL is for an agent with the capacity to act
  - each action influences the agent's future state
  - success is measured by a scalar reward signal
  - goal: select actions to maximize future reward



(source: Schulman and Abbeel)

Core issues in RL (Littman, 2009)

- generalize experience ("learn by interaction")
  - use knowledge gained in similar situations
  - "learning"
- sequential decisions
  - deal properly with delayed gratification
  - "planning"
- exploration/exploitation
  - must strike a ______
  - exploration-exploitation trade-off



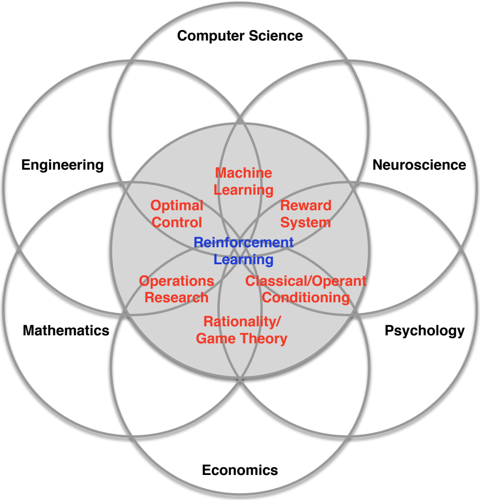
(source: [Sutton and Barto, 2018][5])

[5] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press

When to use RL

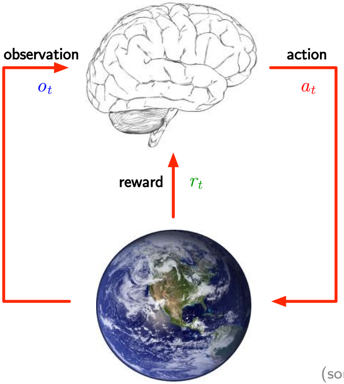
- Pineau (2017):
  - data in the form of __________
  - need to make a sequence of (related) decisions
  - observe (partial, noisy) feedback to choice of actions
  - tasks that require both learning and planning

Many faces of RL



(source: Silver)

Agent and environment



(source: Silver)

- at each step t, _____:
  - executes action a_t
  - receives observation o_t
  - receives scalar reward r_t
- **environment**:
  - receives action a_t
  - emits observation o_{t+1}
  - emits scalar reward r_{t+1}

Experience and state

- **experience**:
  - a sequence of observations, rewards, actions

$$o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t$$

- **state**:
  - a ________ of experience

$$s_t = f(o_1, r_1, a_1, \ldots, a_{t-1}, o_t, r_t)$$

  - in a fully observed environment

$$s_t = f(o_t)$$

Elements of RL

- an agent may include one or more of the following components:

1. policy
   - agent's behavior function
2. value function
   - how good is each **state** and/or **action**
3. model
   - agent's representation of the ___________

Element #1: Policy

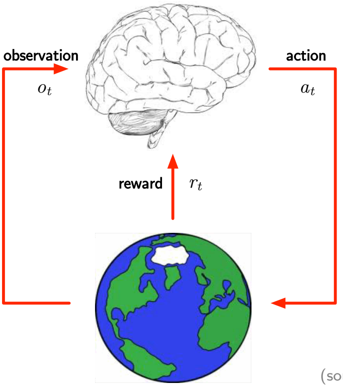
- a policy: the agent's behavior
- it is a map from state to _____
  - deterministic policy: $a = \pi(s)$
  - stochastic policy: $\pi(a \mid s) = \mathbb{P}(a \mid s)$

Element #2: Value function

- a prediction of future ______
  - "how much reward will I get from action a in state s?
- two types[6]
  1. state value function: $V(s)$
  2. state-action value function: $Q(s, a)$

[6] advantage function (slide 28): $A(s, a) = Q(s, a) - V(s)$

Element #3: Model



(source: Silver)

- model:
  - learned from experience
  - acts as proxy for environment
- agent:
  - interacts with model

Approaches to RL

- value-based
  - estimate optimal value function $\underbrace{V^*(s) \text{ or } Q^*(s,a)}_{\substack{\uparrow \\ \text{maximum value achievable under any policy}}}$
  - DP learning, TD learning, Sarsa (on-policy), Q-learning (off-policy)
- policy-based
  - search directly for optimal policy $\underbrace{\pi^*}_{\substack{\uparrow \\ \text{policy achieving maximum future reward}}}$
  - policy ______ (PG), actor-critic/A3C, TRPO, PPO, DDPG[7]
- model-based
  - build a model of environment
  - plan (*e.g.* by lookahead) using the model

[7] A3C: asynchronous advantage actor-critic, TRPO: trust region policy optimization [Schulman et al., 2015], PPO: proximal policy optimization [Schulman et al., 2017], DDPG: deep deterministic PG [Silver et al., 2014]

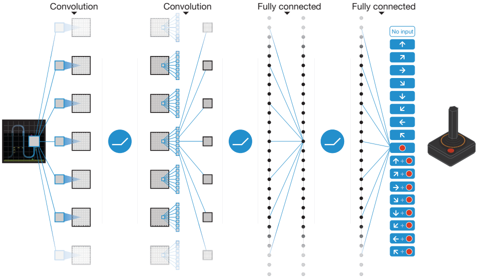
Model-based vs model-free RL (Pineau, 2017)

- model-based:
  - collect large amounts of observed trajectories
  - learn an approximate model of dynamics (*e.g.* with supervised learning)
  - pretend the model is correct and apply value iteration
- model-free:
  - use data to ______ learn value function or optimal policy

Deep RL

- use deep neural nets to represent
  - value function
  - policy
  - model

- optimize loss function by ____



(source: [Mnih et al., 2015][8])

[8]Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529

Outline

Markov decision process (MDP)

- Markov property (memorylessness)
  - state s_t is Markov iff

$$\mathbb{P}[s_{t+1} \mid \underbrace{s_1, \ldots, s_{t-1},}_{\text{history}} s_t] = \mathbb{P}[s_{t+1} \mid s_t]$$

  - state captures all relevant information from ______
  - "the future is independent of the past given the present"

- Markov decision process (MDP)
  - can formalize almost all RL problems
  - describes a $\underbrace{\text{(fully observable) environment}}_{\uparrow}$ for RL

the current state completely characterizes the process

- MDP definition: a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$
  - $\mathcal{S}$ a finite set of Markov states
  - $\mathcal{A}$ a finite set of actions
  - $\mathcal{P}$ a transition probability matrix $\rightarrow$ transition model/dynamics

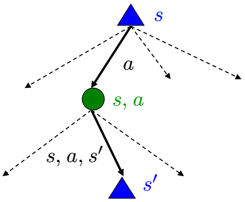
$$\mathcal{P}_{ss'}^{a} = \mathbb{P}[s_{t+1} = s' \mid s_t = s, a_t = a] \triangleq T(s, a, s')$$

  - $\mathcal{R}$ a _____ function $\rightarrow$ immediate/instantaneous reward

$$\mathcal{R}_{s}^{a} = \mathbb{E}[r_t \mid s_t = s, a_t = a] \triangleq r(s, a)$$
$$\mathcal{R}_{ss'}^{a} = \mathbb{E}[r_t \mid s_t = s, a_t = a, s_{t+1} = s'] \triangleq r(s, a, s')$$

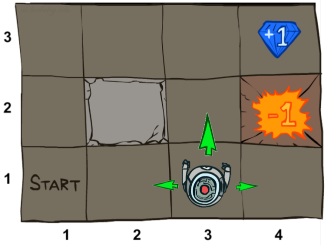
  - γ a discount factor ($\gamma \in [0, 1]$)

- MDP search tree
  - s : a *state*
  - (s, a) : a *q-state*
  - (s, a, s') : a *transition*



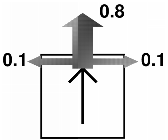
Example: grid world[9]

- a maze-like problem
  - agent lives in a grid
  - walls block agent's path
- ______ movement
  - actions do not always go as planned



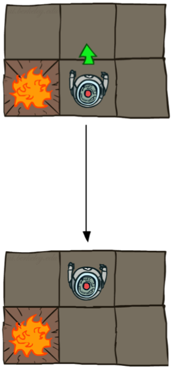
(source: Abbeel & Klein)

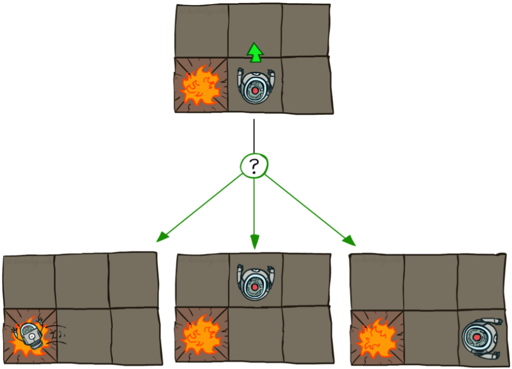
- agent receives rewards $r(s)$ each time step
  - small "living" reward each step (can be negative)
  - big rewards come at the end (good or bad)
- goal: maximize sum of rewards



[9] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson Education Limited

- deterministic grid world
- stochastic grid world





(source: Abbeel & Klein)

Policies

- policy π:
  - fully defines the behavior of an agent
  - stochastic policy: __________ over actions given states

$$\pi(a \mid s) = \mathbb{P}[a_t = a \mid s_t = s]$$

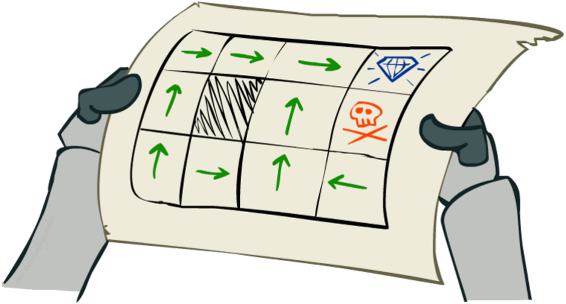
  - deterministic policy: function of given state

$$a = f(s)$$

- MDP policies
  - depend on the current state (not history)

  i.e. stationary (time-independent)

- **example:** grid world



(source: Abbeel & Klein)

  - a policy π for the 4×3 world[10]

[10] the policy shown above happens to be optimal when $r(s) = -0.03$ for all non-terminal states

Return

- return R_t: total discounted reward from time step t

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

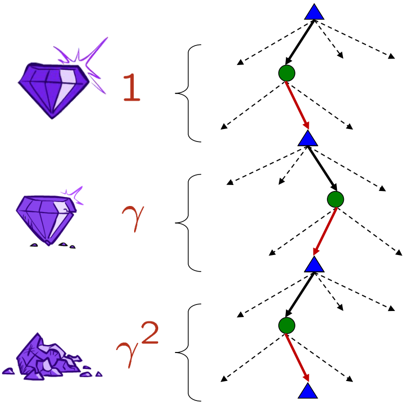
- discount $\gamma \in [0, 1]$: to estimate the present value of future rewards
  - (value of receiving reward r after k time steps) $= \gamma^k r$

- we value immediate reward above delayed reward
  - $\gamma \approx 0$ leads to "myopic" evaluation
  - $\gamma \approx 1$ leads to "__________" evaluation

※ goal of RL: find optimal policy π^* that maximizes the expected return

$$\pi^* = \operatorname*{argmax}_{\pi} \mathbb{E}[R \,|\, \pi]$$

- why discount?
  - sooner rewards probably do have higher utility than later rewards
  - also helps our algorithms ________



(source: Abbeel & Klein)

Value function

- gives the __________ **value** of state s

1. state-value function $V^{\pi}(s)$:
   - expected return, starting from state s, and then following policy π

$$V^{\pi}(s) = \mathbb{E}_{\pi}[R_t \mid s_t = s]$$

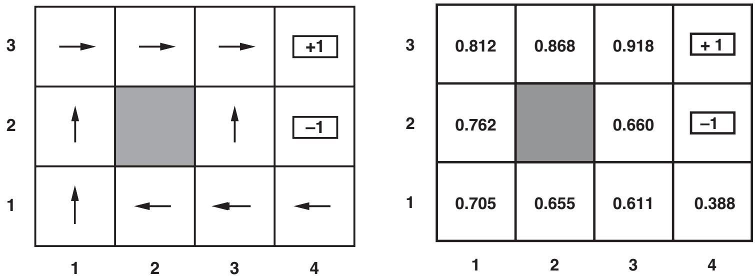
2. action-value function $Q^{\pi}(s, a)$:
   - expected return, starting from s, taking action a, and then following π

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[R_t \mid s_t = s, a_t = a]$$

3. advantage function A^{π}
   - measures how much better action a is than what policy π would've done

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s)$$

- **example:** grid world



(source: [Russell and Norvig, 2016][11])

  - (left) a policy π for the 4×3 world[12]
  - (right) values of the states, given policy π

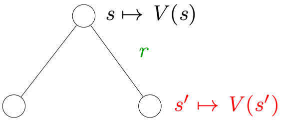
[11] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson Education Limited

[12] this π happens to be optimal when $r(s) = -0.04$ for all non-terminal states and no discounting

Bellman equations

- value function can be decomposed into two parts:
  1. immediate reward r_t
  2. discounted value of successor state $\gamma V(s_{t+1})$

$$
\begin{aligned}
V(s) &= \mathbb{E}[R_t \mid s_t = s] \\
&= \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid s_t = s] \\
&= \mathbb{E}[r_t + \gamma (r_{t+1} + \gamma r_{t+2} + \cdots) \mid s_t = s] \\
&= \mathbb{E}[r_t + \gamma R_{t+1} \mid s_t = s] \\
&= \mathbb{E}[\underbrace{r_t}_{①} + \underbrace{\gamma V(s_{t+1})}_{②} \mid s_t = s]
\end{aligned}
$$



"one-step "

$$V(s) = \mathcal{R}_s + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'} V(s')$$

Solving Bellman equations

- Bellman equation: a _____ equation
  - can be solved directly

$$\begin{bmatrix} V(1) \\ \vdots \\ V(n) \end{bmatrix} = \begin{bmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_n \end{bmatrix} + \gamma \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix} \begin{bmatrix} V(1) \\ \vdots \\ V(n) \end{bmatrix}$$

$$\begin{aligned} \mathbf{V} &= \mathbf{R} + \gamma \mathbf{P} \mathbf{V} \\ &= (\mathbf{I} - \gamma \mathbf{P})^{-1} \mathbf{R} \end{aligned}$$

  - $O(n^3)$ complexity for n states $\Rightarrow$ possible only for *small* problems

- **iterative** methods for *large* MDPs
  - dynamic programming (DP)
  - Monte-Carlo (MC) evaluation
  - temporal difference (TD) learning

Optimal value function

- optimal value function specifies the best possible performance
  - an MDP is "solved" when we know the optimal ____ function
- optimal state-value function $V^*(s)$
  - the maximum value function over all policies

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

- optimal action-value function $Q^*(s, a)$
  - the maximum action-value function over all policies

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

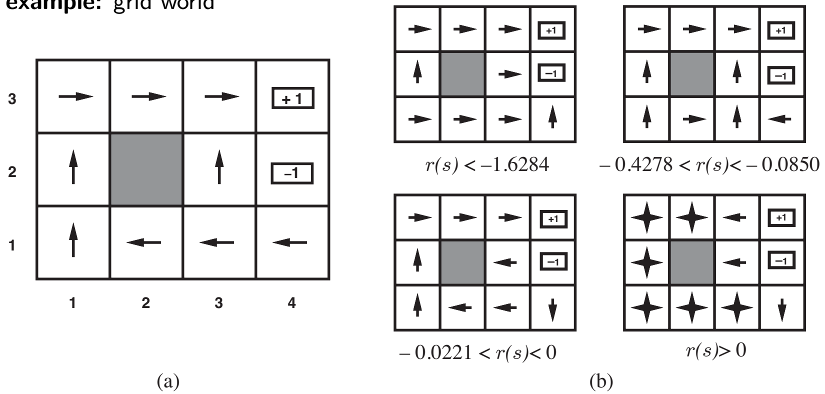
Optimal policy

- theorem: for any MDP
  - there exists an optimal policy π^*
    - is better than or equal to all other policies[13]: $\pi^* \geq \pi, \forall \pi$
    - achieves the optimal value function: $V^{\pi^*}(s) = V^*(s)$
    - achieves the optimal action-value function: $Q^{\pi^*}(s, a) = Q^*(s, a)$
- optimal policy can be found by maximizing over $Q^*(s, a)$
  - if we know $Q^*(s, a)$, we immediately have the optimal policy

$$\pi^*(a \mid s) = \begin{cases} 1 & \text{if } a = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q^*(s, a) \\ 0 & \text{otherwise} \end{cases}$$

[13] partial ordering over policies: $\pi \geq \pi'$ if $V^{\pi}(s) \geq V^{\pi'}(s), \forall s$

- **example:** grid world



(source: [Russell and Norvig, 2016][14])

(a) an optimal policy for the stochastic environment with $r(s) = -0.04$[15]

(b) optimal policies for four different ranges of $r(s)$

[14] Russell, S. J. and Norvig, P. (2016). *Artificial intelligence: a modern approach*. Pearson Education Limited

[15] in the nonterminal states

Clarification

- similar but distinct concepts:
  - reward: 1-step numerical feedback
  - return: sum of rewards over the agent's trajectory
  - value: expected sum of rewards over the agent's trajectory (= avg ______)
- episodic vs continuing:
  - **episodic** task
    - consider return over finite horizon (*e.g.* games, maze)
  - **continuing** task
    - consider return over infinite horizon (*e.g.* juggling, balancing)

Outline

Summary

- reinforcement learning (RL): "learn by interaction"
  - a general-purpose framework for decision making
  - elements: policy (agent behavior), value (utility of state/action), model
  - deep RL: use of neural nets + backprop for any of these elements

- approaches to reinforcement learning
  - value-based: estimate optimal value function $V^*(s)$ or $Q^*(s, a)$
    - DP learning, TD learning, Sarsa (on-policy), Q-learning (off-policy)
  - policy-based: search directly for optimal policy π^*
    - policy gradient, actor-critic/A3C, TRPO, PPO, DDPG
  - model-based: build a model of environment and plan using it

- deep reinforcement learning (DRL): use deep neural nets
  - to represent value function, policy, and/or model (trained via SGD)