Introduction to Reinforcement Learning

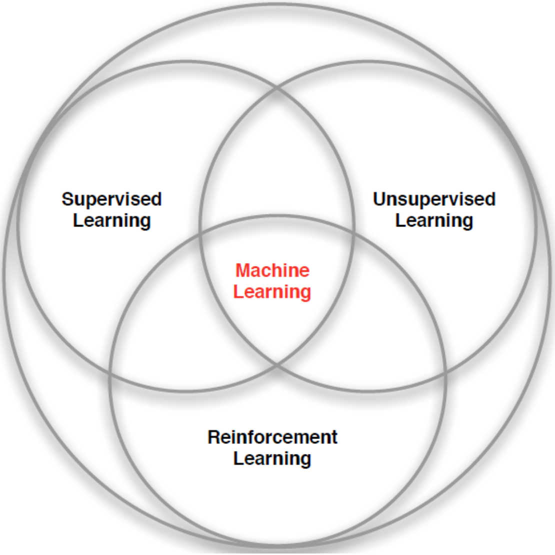
Koganti Nishanth

June 21, 2019

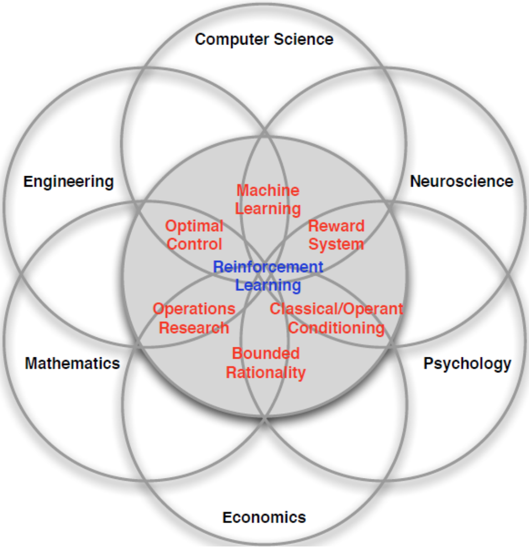
Overview

1. Why RL?
2. What is RL?
3. Components
4. Examples
5. Resources

Branches of Machine Learning



Domains of Reinforcement Learning



Examples of Reinforcement Learning

- Performing stunts on an helicopter[1].
- Managing an investment portfolio.
- Playing atari games better than humans[2].
- Defeating world champion of Go[3].
- Performing drug discovery[4].

[1] Abbeel, et al. "An application of reinforcement learning to aerobatic helicopter flight." in *NIPS 2007*.

[2] Mnih, et al. "Human-level control through deep reinforcement learning." in *Nature 2015*.

[3] Silver, et al. "Mastering the game of Go with deep neural networks and tree search." in *Nature 2016*.

[4] AlphaFold 2018. Available at `https://deepmind.com/blog/alphafold/`

Autonomous Helicopter Flight[1]

[1] Abbeel, et al. “An application of reinforcement learning to aerobatic helicopter flight.” in *NIPS 2007*.

Characteristics of Reinforcement Learning

What makes Reinforcement Learning different?

- There is no supervision, only a *reward* signal.
- Feedback is delayed, not instantaneous.
- Agent's actions effect subsequent data it receives.

Reward Hypothesis

Reinforcement learning is based on the *reward hypothesis*:

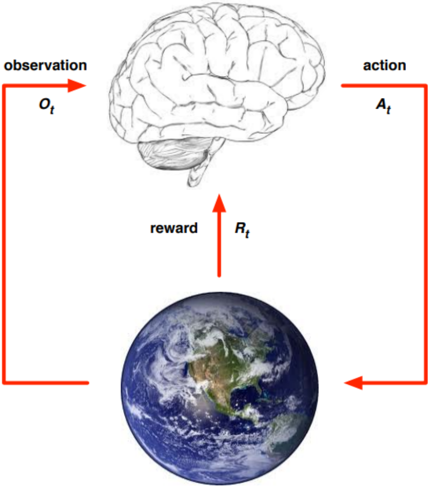
Reward Hypothesis

All goals can be described by maximization of expected cumulative reward.

Example of rewards:

- Helicopter Flight:
  - Reward for following desired trajectory.
  - Punishment for crashing.
- Play Atari games:
  - Reward/punishment for increasing/decreasing score.

Agent and Environment



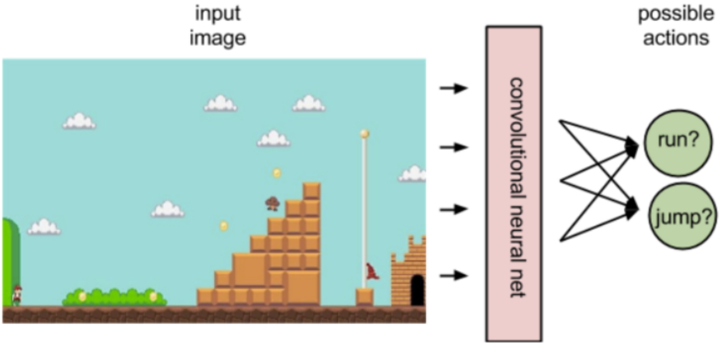
- At each step t, the agent:
    - Executes action a_t.
    - Receives states s_t.
    - Receives scalar reward r_t.
- The environment:
    - Receives action a_t.
    - Emits observation s_{t+1}.
    - Emits scalar reward r_{t+1}.

Major Components of RL Agents

- RL Agent may include following components:
  - Policy: Agent's behavior function.
  - Value function: Utility of each state and/or action.
  - Model: Agent's representation of the environment.

Policy

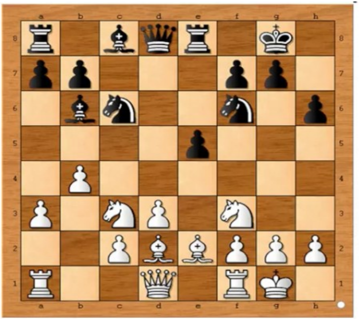
- A *policy* models the agent's behavior.
- Function from state to action.
- Deterministic policy: $a = \pi(s)$.
- Stochastic policy: $\pi(a|s) = \mathbb{P}[a_t = a|s_t = s]$.



Value Function

- Value function is a prediction of future reward.
- Used to evaluate the goodness/badness of states.
- Select between actions using this value function.

$$v_\pi(s) = \mathbb{E}_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s]$$



Is this a good state for white?

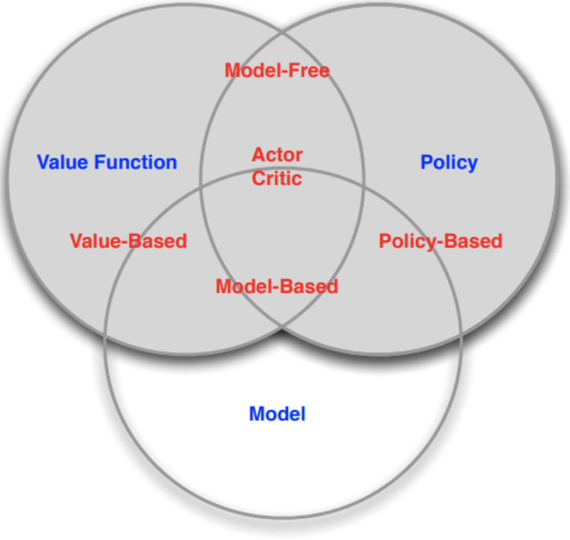
Model

- A *model* predicts what the environment will do next
- $\mathcal{P}$ predicts the next state
- $\mathcal{R}$ predicts the next reward

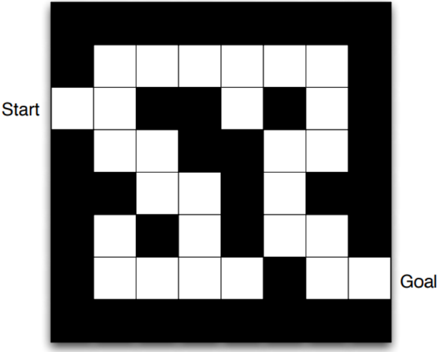
$$\mathcal{P}^{a}_{ss'} = \mathbb{P}[s_{t+1} = s' | s_t = s, a_t = a]$$

$$\mathcal{R}^{a}_{s} = \mathbb{R}[r_{t+1} | s_t = s, a_t = a]$$

Taxonomy of RL Algorithms

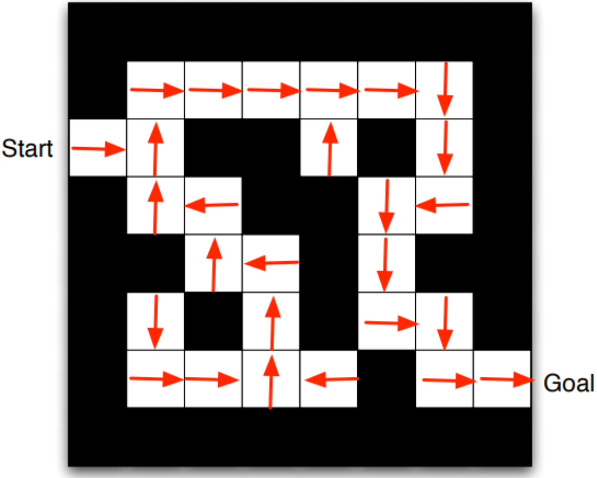


Maze Example



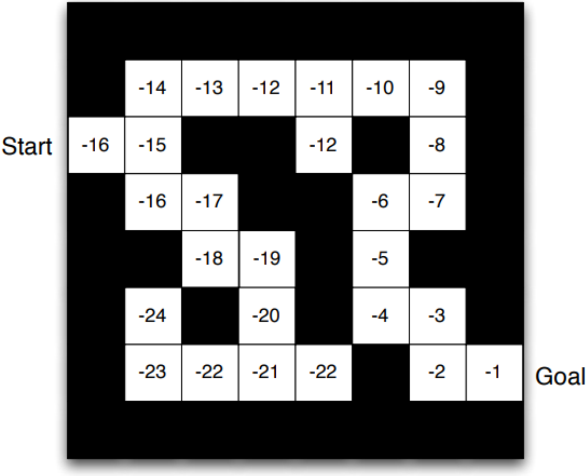
- Rewards: -1 per time step.
- Actions: N,S,E,W.
- State: Agent's position on maze.

Maze Example: Policy



Red arrows indicate actions taken as per policy.

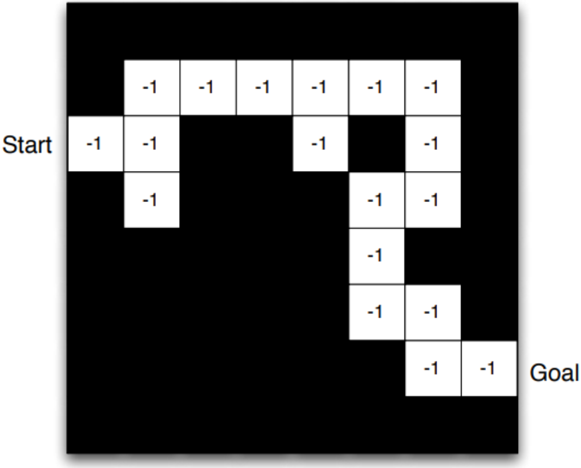
Maze Example: Value Function



		-14	-13	-12	-11	-10	-9		
Start	-16	-15			-12		-8		
		-16	-17			-6	-7		
			-18	-19		-5			
		-24		-20		-4	-3		
		-23	-22	-21	-22		-2	-1	Goal

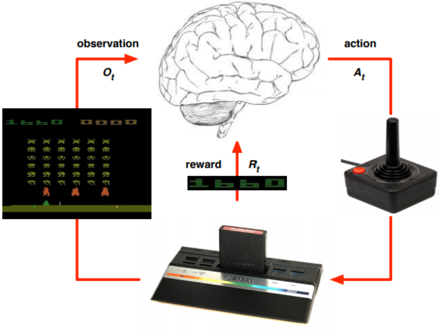
Numbers in grid indicate expected *long-term* rewards for that cell.

Maze Example: Model



$\mathcal{P}^a_{ss'}$ and $\mathcal{R}^a_s$ are shown by grid and numbers.
Note: Model can be imperfect!

Case Study: Atari Game Play



- Learn directly from interactive game play.
- Relies on value function based RL.
- Approximate value function using deep neural network.

[1] Mnih, et al. “Human-level control through deep reinforcement learning.” in *Nature 2015*.

Tutorials for RL

- Andrew Ng: CS 229 Course Lectures 16-20[1].
- David Silver: Reinforcement Learning Course[2].
- Spinning up with Deep Reinforcement Learning: OpenAI[3].

[1] https://www.youtube.com/watch?v=UzxYlbK2c7E

[2] http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html

[3] https://spinningup.openai.com/en/latest/

Software Resources

- Awesome Reinforcement Learning: github.com/aikorea/awesome-rl.

- OpenAI Gym: `github.com/openai/gym`.
- Unity ML: `github.com/Unity-Technologies/ml-agents`.

- garage: `github.com/rlworkgroup/garage`.
- trfl: `github.com/deepmind/trfl/`.

Thank You