CS 369: Introduction to Robotics

Prof. Thao Nguyen

Spring 2026

HAVERFORD
COLLEGE

Outline for today

- Reinforcement learning (RL)

- Types of RL algorithms

Outline for today

- Reinforcement learning (RL)

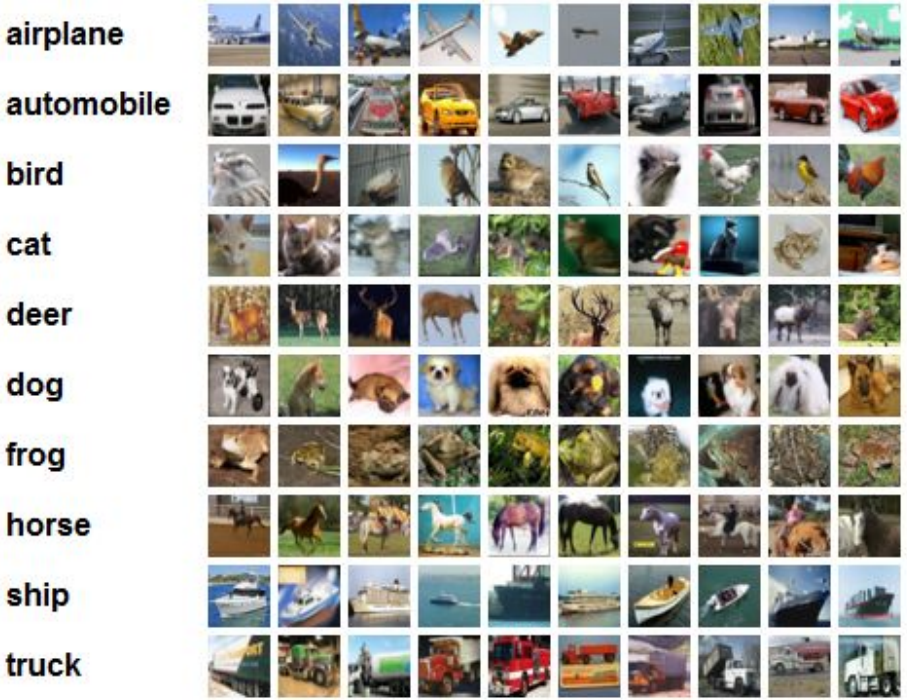
- Types of RL algorithms

Machine learning

- Concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data
- Perform tasks without being explicitly programmed for them
- Types of ML:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning

Supervised learning

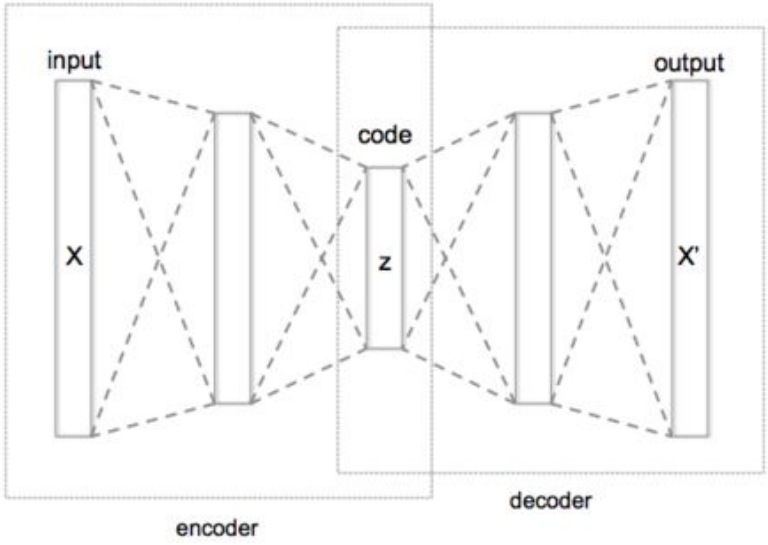
- Training data includes the ground truth (label) for each datapoint
- Learn to predict labels for new data points



CIFAR-10

Unsupervised learning

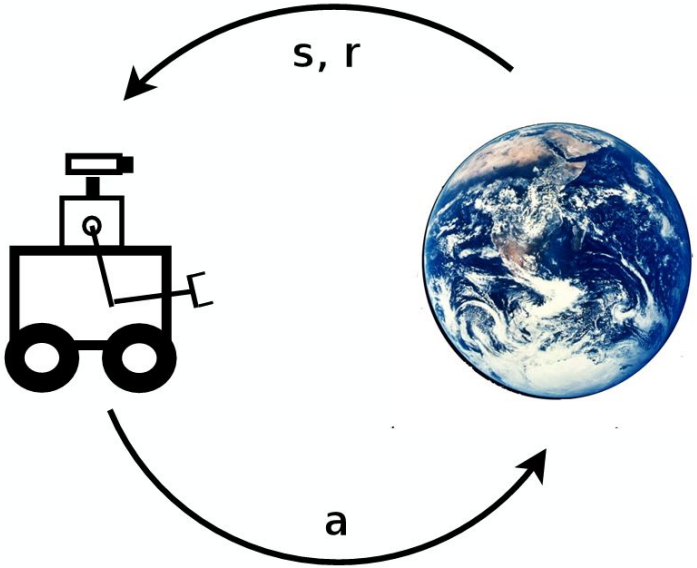
- No labels are included with the training data
- Learn the structure of the data



Wikipedia

Reinforcement learning

- Agent interacting with a dynamic environment
- Training data includes rewards
- Learn to achieve a goal



Markov decision process (MDP)

An MDP is a tuple$\langle S, A, P, R, \gamma \rangle$where:

- S is the set of all valid states
- A is the set of all valid actions
- P is the transition function $P(s_{t+1} \mid s_t, a_t)$
- R is the reward function $R(s_t, a_t, s_{t+1})$
- γ is the discount factor $\gamma \in (0,1]$

<u>Goal:</u> Find policy that maximizes expected return.

Policy

- Parameterized mapping from states to actions

$$\pi_\theta: S \rightarrow A$$

- Can be deterministic or stochastic

- Trajectory: sequence of states and actions in the world

$$\tau = (s_0, a_0, s_1, a_1, \ldots)$$

Return

- Cumulative discounted reward over a trajectory

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$$

- Expected return: $\underset{\tau \sim \pi}{\mathrm{E}} [R(\tau)] = \int_{\tau} P(\tau|\pi) R(\tau) = J(\pi)$

- Optimal policy: $\pi^* = \arg\max_{\pi} J(\pi)$

Value functions

1. The **On-Policy Value Function**, $V^{\pi}(s)$, which gives the expected return if you start in state s and always act according to policy π:

$$V^{\pi}(s) = \underset{\tau \sim \pi}{\mathrm{E}} \left[R(\tau) \,|s_0 = s \right]$$

2. The **On-Policy Action-Value Function**, $Q^{\pi}(s, a)$, which gives the expected return if you start in state s, take an arbitrary action a (which may not have come from the policy), and then forever after act according to policy π:

$$Q^{\pi}(s, a) = \underset{\tau \sim \pi}{\mathrm{E}} \left[R(\tau) \,|s_0 = s, a_0 = a \right]$$

3. The **Optimal Value Function**, $V^{*}(s)$, which gives the expected return if you start in state s and always act according to the *optimal* policy in the environment:

$$V^{*}(s) = \max_{\pi} \underset{\tau \sim \pi}{\mathrm{E}} \left[R(\tau) \,|s_0 = s \right]$$

4. The **Optimal Action-Value Function**, $Q^{*}(s, a)$, which gives the expected return if you start in state s, take an arbitrary action a, and then forever after act according to the *optimal* policy in the environment:

$$Q^{*}(s, a) = \max_{\pi} \underset{\tau \sim \pi}{\mathrm{E}} \left[R(\tau) \,|s_0 = s, a_0 = a \right]$$

Inverse reinforcement learning

- Imitation learning technique
- Learn reward function from expert demonstrations then perform RL
- Generalize better than behavioral cloning

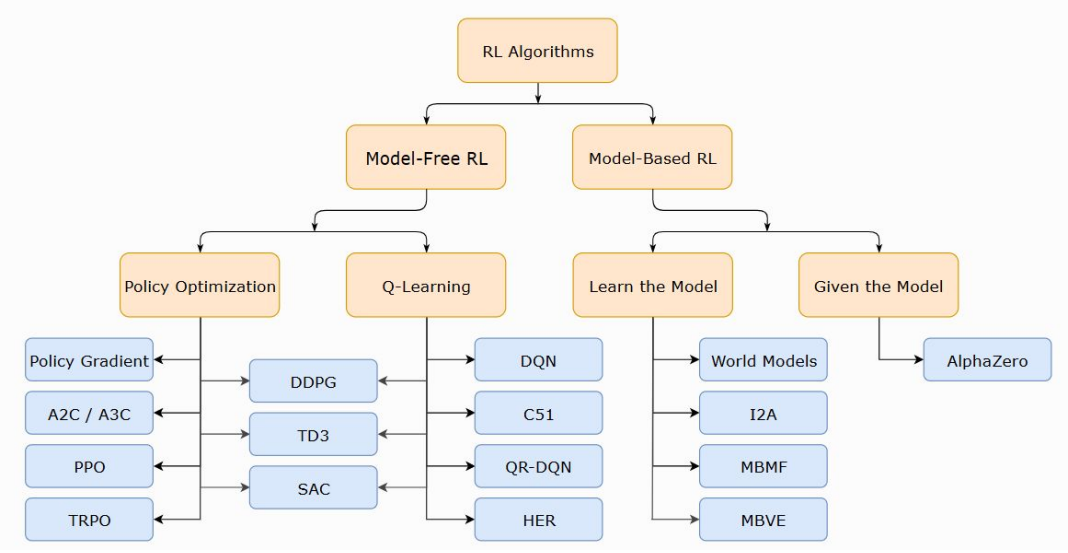
Outline for today

- Reinforcement learning (RL)

- Types of RL algorithms

Model-free vs. Model-based RL

- Environment model: transition and reward functions
- Planning: given environment model, simulate interactions



Spinning up RL

Policy optimization vs. Q-learning

- Policy optimization: “directly” optimize policy parameters θ
- Q-learning: learn an approximator $Q_\theta(s,a)$ for the optimal action-value function $Q^*(s,a)$. Actions given by: $a(s) = \arg\max_a Q_\theta(s,a)$
- On-policy: only uses data collected while acting according to the most recent version of the policy
- Off-policy: can use data collected at any point during training