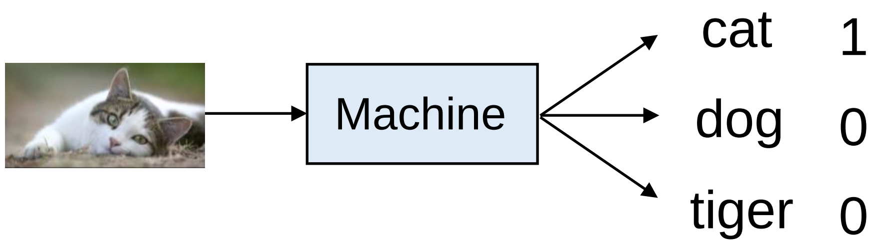
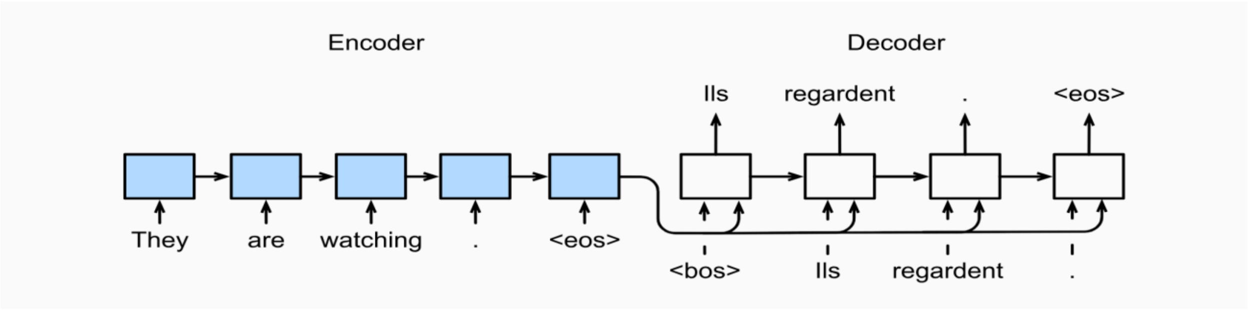
Summary

Chen-Yu Wei

Scenarios we focused on in this course

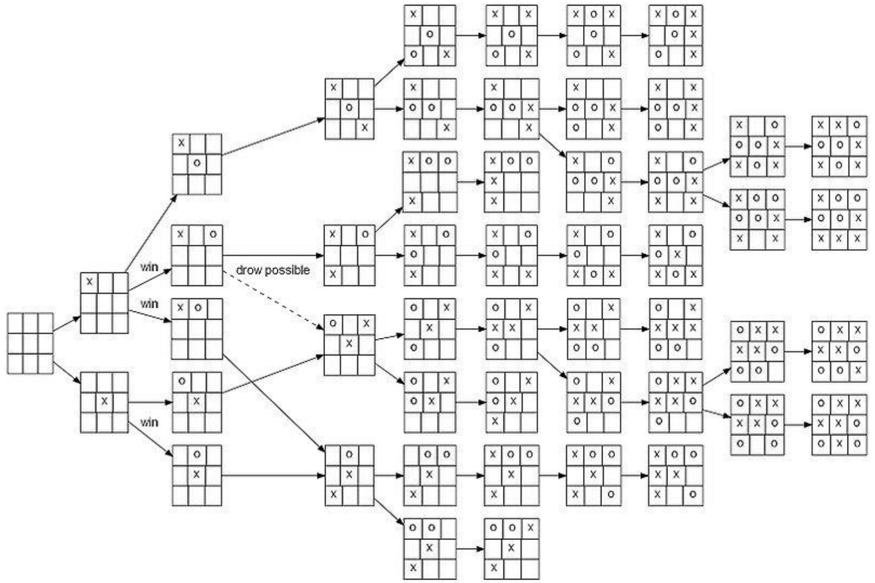


Learning from reward



Learning to make sequential decisions

Scenarios we focused on in this course



Learning from reward

… with bandit feedback

Exploration

Learning to make sequential decisions

… with delayed and aggregated feedback

Credit Assignment

Challenges in RL

- Generalization
- Exploration-exploitation tradeoff
- Credit assignment
- Distribution mismatch

.. and more

Course Content

(Focusing on exploration-exploitation tradeoff)

Part I. Learning in Bandits

- Multi-armed bandits
- Linear bandits
- Contextual bandits
- Adversarial multi-armed bandits
- Adversarial linear bandits

Part II. Basics of MDPs

- Bellman (optimality) equations
- Value iteration
- Policy iteration

(Focusing on credit assignment and distribution mismatch)

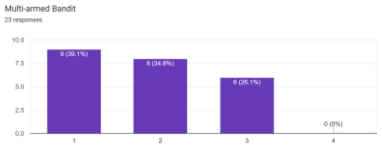
Part III. Learning in MDPs

- Approximate value iteration and variants
  - Least-square value iteration
  - Q-Learning
  - DQN
- Policy evaluation
  - Temporal difference
  - Monte Carlo
- Approximate policy iteration and variants
  - Least-square policy iteration
  - (Natural) policy gradient and actor-critic
  - REINFORCE, A2C, PPO, SAC
  - DDPG

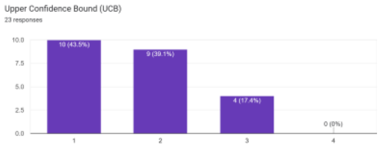
~~**Part IV. Offline RL**~~

Student Project Presentation

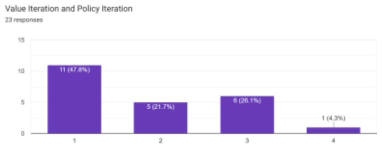
Prior Knowledge Before the Course



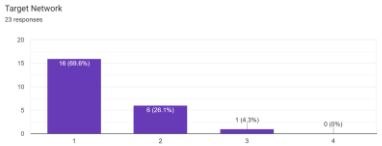
Multi-armed Bandit



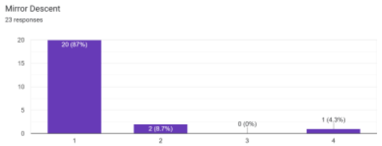
UCB



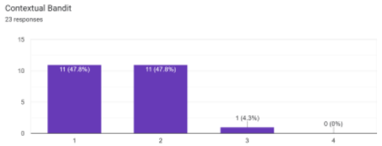
VI & PI



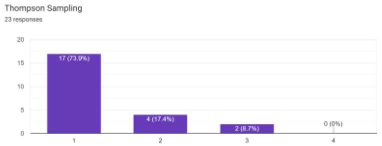
Target Network



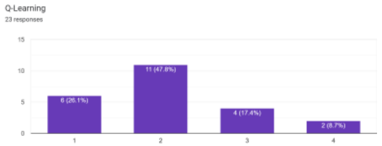
Mirror Descent



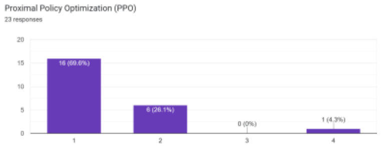
Contextual Bandit



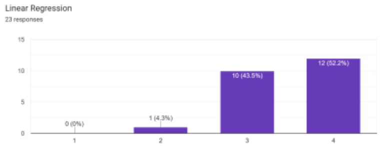
Thompson Sampling



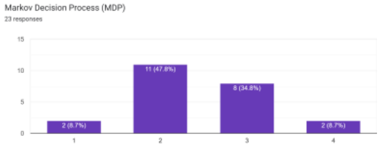
Q-Learning



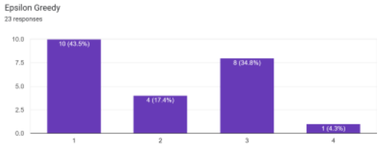
PPO



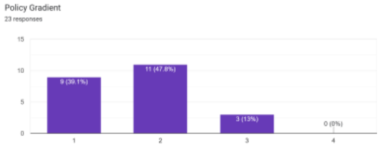
Linear Regression



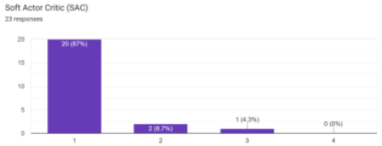
MDP



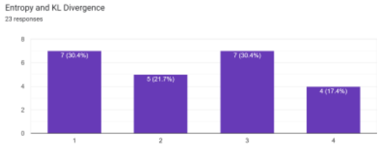
ϵ-greedy



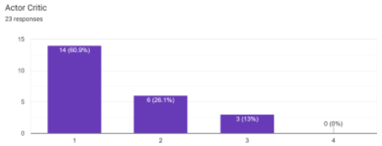
Policy Gradient



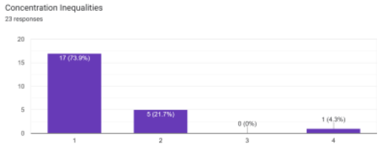
SAC



Entropy & KL Divergence



Actor Critic



Concentration Inequality

Exploration in Bandits

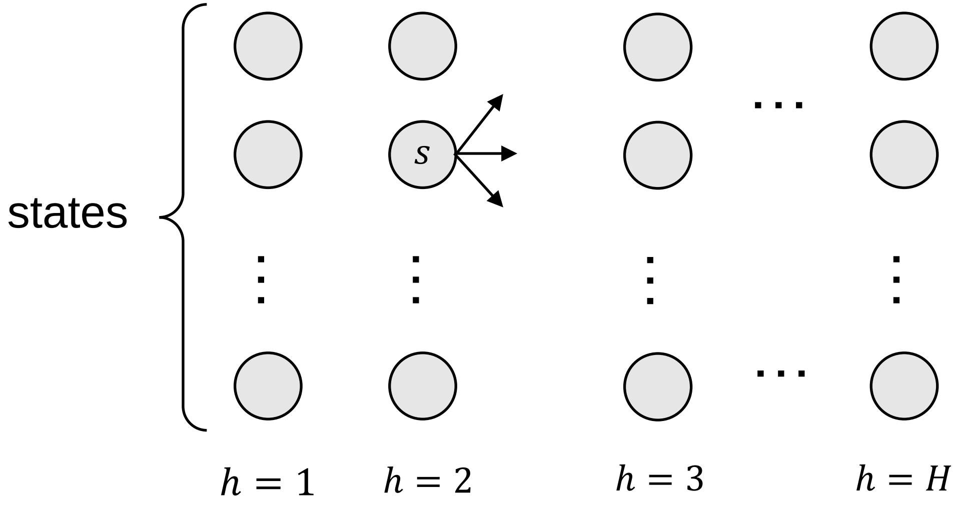
- Approaches
  - Exploration bonus or perturbation on **values** + greedy, e.g., UCB, Thompson sampling
  - Policies randomization, e.g., ϵ-greedy, Boltzmann exploration
  - Baseline, e.g., $\frac{r_t(a)-\mathbf{1}}{p_t(a)}\,\mathbb{I}\{a_t = a\}$ in EXP3

- The degree of exploration may be
  - Agnostic about uncertainty, e.g., $p(a) \propto \exp\left(\lambda \hat{R}_t(a)\right)$
  - Uncertainty-aware, e.g., $\operatorname*{argmax}_a \left(\hat{R}_t(a) + \frac{c}{\sqrt{\boldsymbol{N_t(a)}}}\right)$

Credit Assignment

Model the problem as Markov decision process, and try to find the optimal action on every state

Markov Decision Processes



Bellman Optimality Equation

$$Q^\star(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^\star(s',a')$$

Related algorithms: (approximate) value iteration

Bellman Equation

$$Q^\pi(s,a) = R(s,a) + \gamma \sum_{s',a'} P(s'|s,a)\, \pi(a'|s') Q^\pi(s',a')$$

Related algorithms: (approximate) policy evaluation

Performance Difference Lemma

$$V^{\pi'}(\rho) - V^\pi(\rho) = \sum_{s,a} d_\rho^{\pi'}(s)\left(\pi'(a|s) - \pi(a|s)\right) Q^\pi(s,a)$$

Related algorithms: (approximate) policy iteration, policy gradient

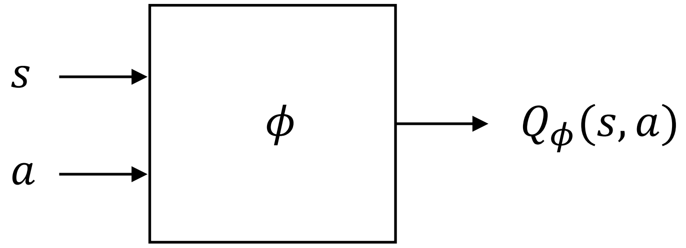
Value-Based Approach (for $Q^\star$)

Try to make

$$Q_\phi(s,a) \approx R(s,a) + \gamma\, \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[\max_{a'} Q_\phi(s',a')\right]$$

Value Iteration + Regression in each iteration

$$\phi_{k+1} \leftarrow \operatorname*{argmin}_{\phi} \sum_i \left(Q_\phi(s_i,a_i) - r_i - \gamma \max_{a'} Q_{\phi_k}(s_i', a')\right)^2$$



Source of instability: function approximation error, insufficient samples, non-i.i.d., max operator

Accompanied techniques: replay buffer, target network, double network

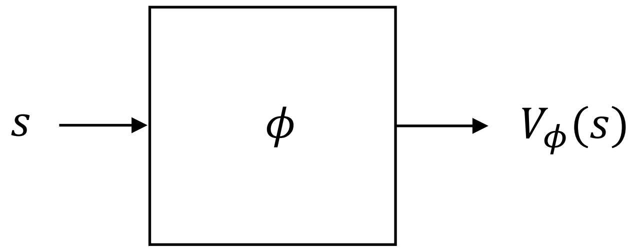
LSVI, DQN, DDQN

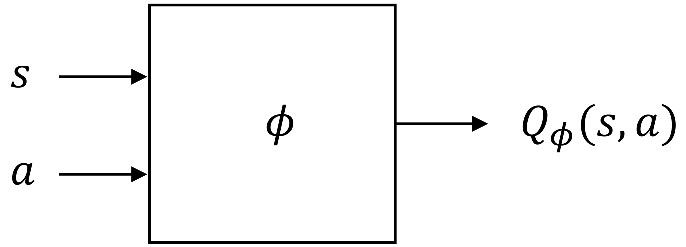
Policy Evaluation (for V^π, Q^π)

Try to make

$$V_\phi(s) \approx \mathbb{E}_{a\sim\pi(\cdot|s)}\left[R(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V_\phi(s')\right]\right]$$

$$Q_\phi(s,a) \approx R(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)} \mathbb{E}_{a'\sim\pi(\cdot|s')}\left[Q_\phi(s',a')\right]$$





Temporal difference learning (with on-policy samples)

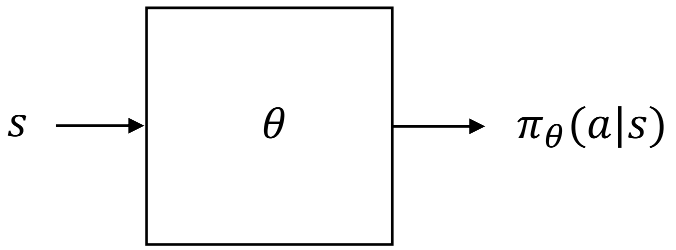
$$\phi_{k+1} \leftarrow \phi_k - \alpha \sum_i \nabla_\phi \left(V_\phi(s_i) - r_i - \gamma V_{\phi_k}(s_i')\right)^2$$

$$\phi_{k+1} \leftarrow \phi_k - \alpha \sum_i \nabla_\phi \left(Q_\phi(s_i,a_i) - r_i - \gamma Q_{\phi_k}(s_i',a_i')\right)^2$$

Can combine with Monte Carlo estimation to balance bias and variance

TD(0), TD(λ), Monte Carlo Estimation

Policy-Based Approach



NPG, PG

Natural Policy Gradient or Policy Gradient

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \left(V^{\pi_\theta} - V^{\pi_{\theta_k}} - \frac{1}{\eta} D(\theta, \theta_k) \right)$$

or $\theta_{k+1} \leftarrow \theta_k + \eta \nabla_\theta V^{\pi_{\theta_k}}$

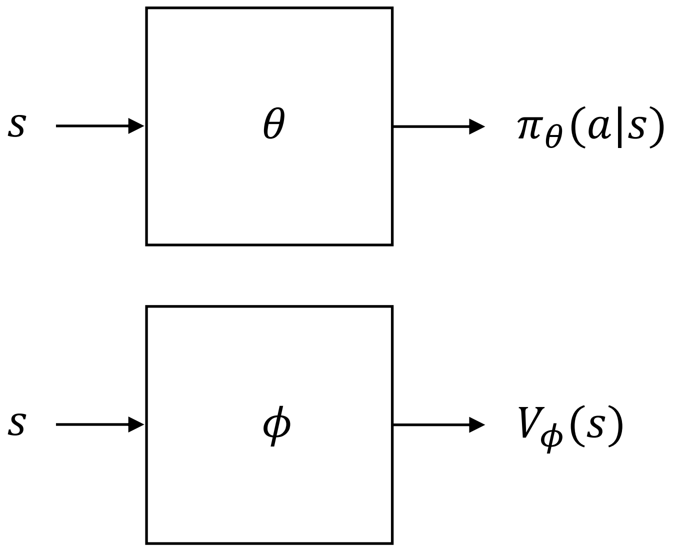
Estimate from samples using **Monte Carlo estimators**

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_i \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \boldsymbol{R_i} - \frac{1}{\eta} D\big(\pi_\theta(\cdot\,|s_i), \pi_{\theta_k}(\cdot\,|s_i)\big) \right)$$

$$\theta_{k+1} \leftarrow \theta_k + \eta \sum_i \nabla_\theta \log \pi_{\theta_k}(a_i|s_i)\ \boldsymbol{R_i}$$

$\boldsymbol{R_i} \coloneqq$ sum of trajectory reward from (s_i, a_i)

Actor-Critic Approach



A2C, PPO

Natural Policy Gradient or Policy Gradient

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \left(V^{\pi_\theta} - V^{\pi_{\theta_k}} - \frac{1}{\eta} D(\theta, \theta_k) \right)$$

or $\theta_{k+1} \leftarrow \theta_k + \eta \nabla_\theta V^{\pi_{\theta_k}}$

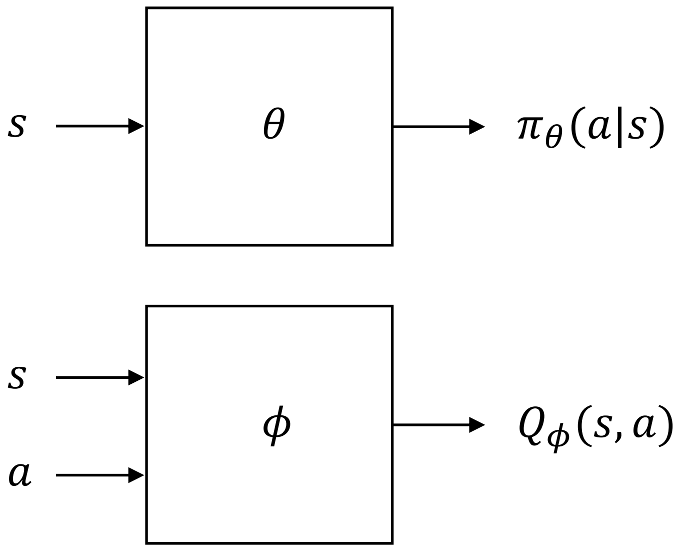
Estimate from samples using **On-Policy Policy Evaluation**

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_i \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \boldsymbol{A_i} - \frac{1}{\eta} D\big(\pi_\theta(\cdot\,|s_i), \pi_{\theta_k}(\cdot\,|s_i)\big) \right)$$

$$\theta_{k+1} \leftarrow \theta_k + \eta \sum_i \nabla_\theta \log \pi_{\theta_k}(a_i|s_i)\ \boldsymbol{A_i}$$

$\boldsymbol{A_i}$ ≔ advantage estimator, e.g., $r_i + \gamma V_\phi(s_i') - V_\phi(s_i)$

Actor-Critic Approach



+ target network, replay buffer, double Q-network

DDPG, TD3, SAC

Natural Policy Gradient or Policy Gradient

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \left(V^{\pi_\theta} - V^{\pi_{\theta_k}} - \frac{1}{\eta} D(\theta, \theta_k) \right)$$

or $\theta_{k+1} \leftarrow \theta_k + \eta \nabla_\theta V^{\pi_{\theta_k}}$

$\downarrow$ Estimate from samples using **Off-Policy Policy Evaluation**

$$\theta_{k+1} \leftarrow \underset{\theta}{\operatorname{argmax}} \sum_i \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \boldsymbol{Q_i} - \frac{1}{\eta} D\big(\pi_\theta(\cdot\,|s_i), \pi_{\theta_k}(\cdot\,|s_i)\big) \right)$$

$$\theta_{k+1} \leftarrow \theta_k + \eta \sum_i \nabla_\theta \log \pi_{\theta_k}(a_i|s_i)\ \boldsymbol{Q_i}$$

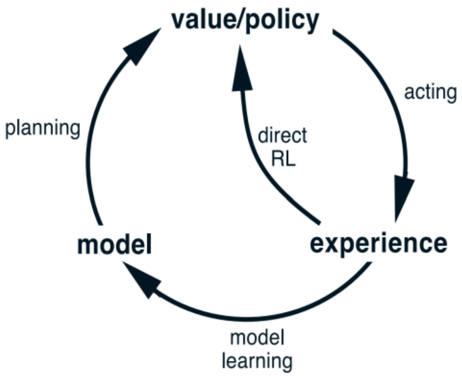
$$\boldsymbol{Q_i} \coloneqq Q_\phi(s_i, a_i)$$

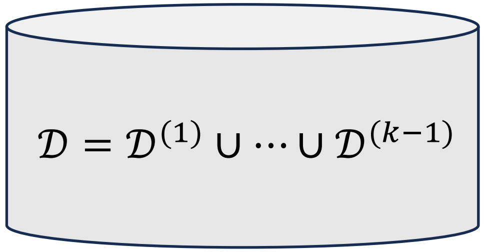
Some Topics Not Covered

Topics Not Covered

- Model-Based Approach
- Offline RL
- Imitation Learning
- Inverse RL
- Distributional RL
- Hierarchical RL

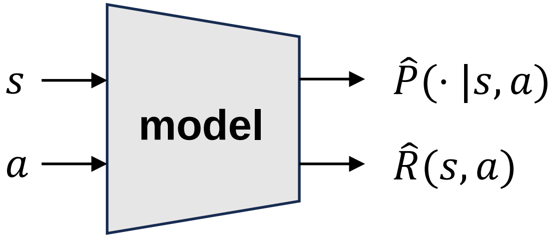
Model-Based Reinforcement Learning





$$\phi_k \leftarrow \underset{\theta}{\operatorname{argmin}}\ \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - r - \gamma \max_{a'} Q_{\phi_{k-1}}(s',a')\right)^2\right]$$

Model-free



Trained with $\mathcal{D}$

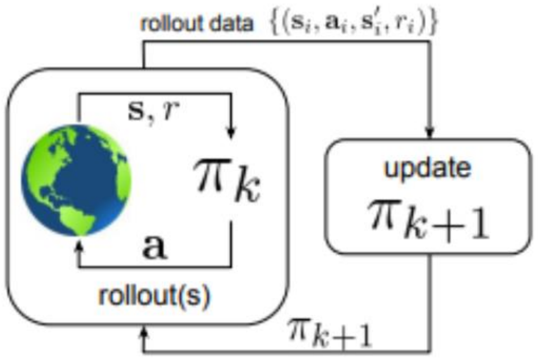
Loop: Interact with environment → model training → planning

Planning: Find a good policy using the trained model

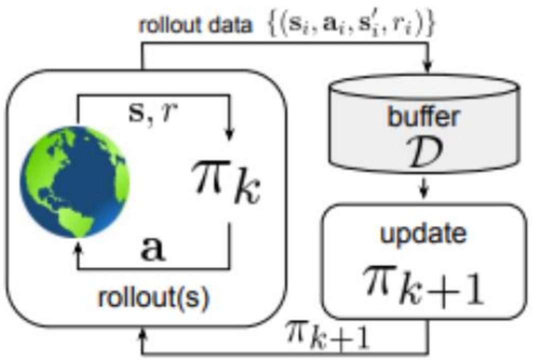
Model-based

Offline Reinforcement Learning

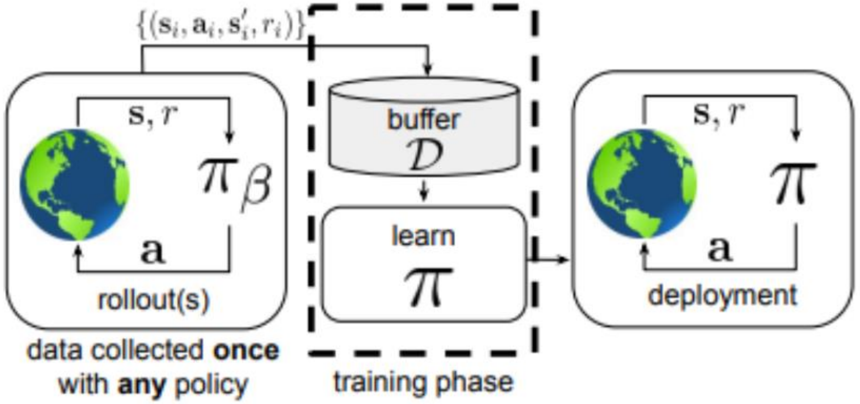
on-policy RL



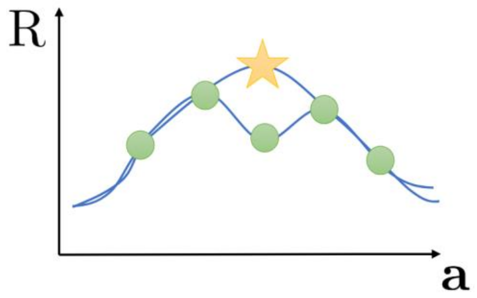
off-policy RL



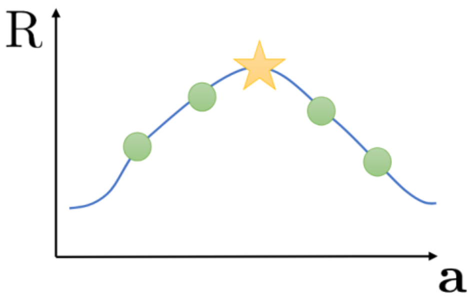
offline reinforcement learning



online RL setting



offline RL setting



Additional challenge compared to online RL: errors are not corrected

Offline RL: Be Conservative and Pessimistic

Conservative Q-learning:

For $k = 1,2,\ldots$

Obtain ϕ_k by minimizing $L(\phi_k)$

Let $\pi = \mathrm{Greedy}(Q_{\phi_k})$

$$L(\phi) = \sum_i \left(Q_\phi(s,a) - r - \mathbb{E}_{a' \sim \pi(\cdot|s')}\left[Q_{\phi_{k-1}}(s',a')\right]\right)^2 + \alpha\left(\max_\mu \mathbb{E}_{\tilde{a} \sim \mu(\cdot|s)}\left[Q_\phi(s,\tilde{a})\right] - Q_\phi(s,a)\right)$$

Kumar, Zhou, Tucker, Levine. Conservative Q-Learning for Offline Reinforcement Learning. 2020.

Goal of This Course (from the first lecture)

We will

- Provide a **systematic overview** of basic techniques in RL
- Provide **reasonings** for the design of RL algorithms
- Provide **mathematical tools** to analyze RL algorithms

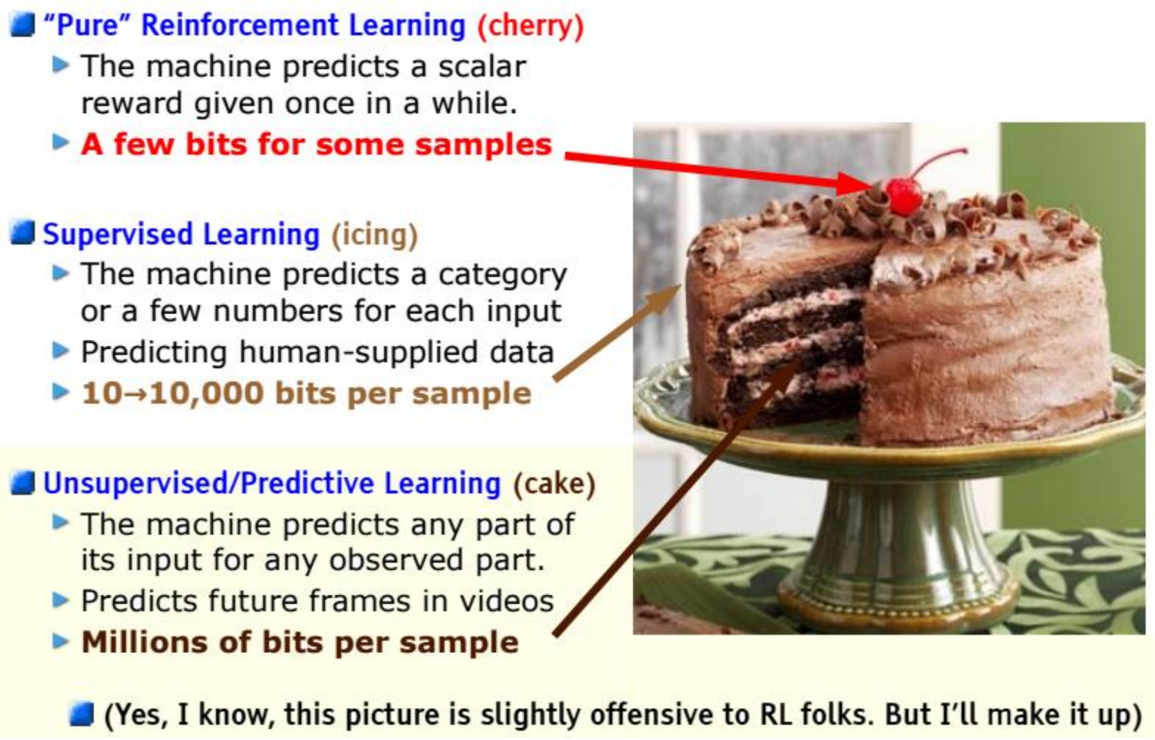
After taking this course, you should be able to

- Feel grounded when reading other RL materials
- Implement basic RL algorithms
- Know **design principles** of RL algorithms

Final Remark: RL with reward has sparse signal

SL feedback: “what to do in each step”

RL feedback: “how you’re doing overall”



SL and RL differ because the supervision signals are different.

Our goal is to learn decision-making. There can be many **supervision signals**:

- Demonstration
- Language
- Preference feedback

There is also **offline data** not directly related to the task, but useful in building a world model.

Try to combine RL with other ML techniques to accomplish your task.