CS 4803 / 7643: Deep Learning

Topics:

- Policy Gradients
- Actor Critic

Ashwin Kalyan

Georgia Tech

Topics we'll cover

- Overview of RL
  - RL vs other forms of learning
  - RL "API"
  - Applications
- Framework: Markov Decision Processes (MDP's)
  - Definitions and notations
  - Policies and Value Functions
  - Solving MDP's
    - Value Iteration (recap)
    - Q-Value Iteration (new)
    - Policy Iteration
- Reinforcement learning
  - Value-based RL (Q-learning, Deep-Q Learning)
  - Policy-based RL (Policy gradients)
  - Actor-Critic

Recap: MDPs

- Markov Decision Processes (MDP):
  - States: $\mathcal{S}$
  - Actions: $\mathcal{A}$
  - Rewards: $\mathcal{R}(s, a, s')$
  - Transition Function: $\mathbb{T}(s, a, s') = p(s'|s, a)$
  - Discount Factor: γ

Recap: Optimal Value Function

The **optimal Q-value function** at state s and action a, is the expected cumulative reward from taking action a in state s and acting optimally thereafter

$$Q^*(s,a) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi^*\right]$$

Recap: Optimal Value Function

The **optimal Q-value function** at state s and action a, is the expected cumulative reward from taking action a in state s and acting optimally thereafter

$$Q^*(s,a) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi^*\right]$$

Optimal policy:

$$\boxed{\pi^*(s) = \arg\max_a Q^*(s,a)}$$

Recap: Learning Based Methods

- Typically, we don't know the environment
  - $\mathbb{T}(s, a, s')$ unknown, how actions affect the environment.
  - $\mathcal{R}(s, a, s')$ unknown, what/when are the good actions?

Recap: Learning Based Methods

- Typically, we don't know the environment
  - $\mathbb{T}(s, a, s')$ unknown, how actions affect the environment.
  - $\mathcal{R}(s, a, s')$ unknown, what/when are the good actions?
- But, we can learn by trial and error.
  - Gather experience (data) by performing actions.

$$\{s, a, s', r\}_{i=1}^{N}$$

  - Approximate unknown quantities from data.

Recap: Deep Q-Learning

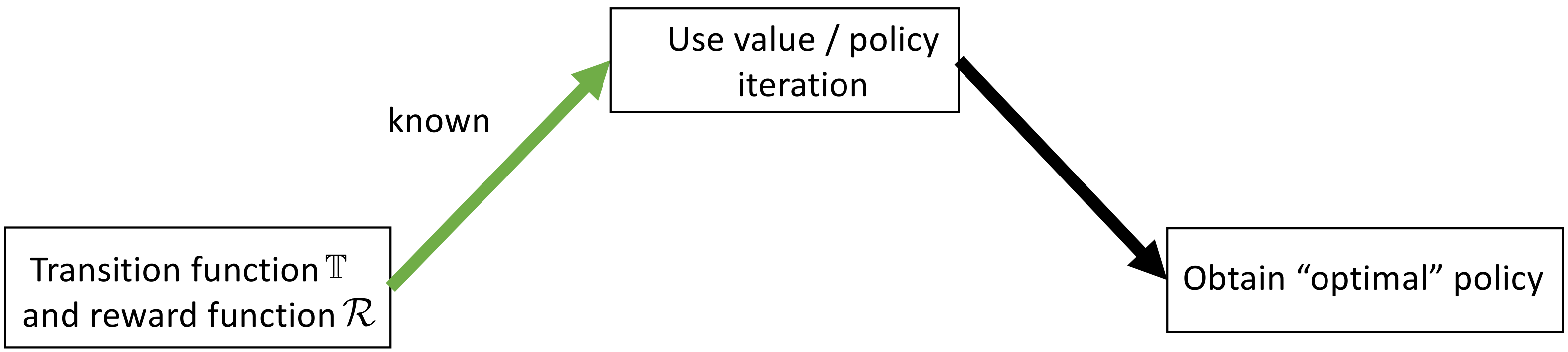
- Collect a dataset $\{(s, a, s', r)_i\}_{i=1}^{N}$
- Loss for a single data point:

$$\text{MSE Loss} := \Big(\underbrace{Q_{new}(s, a)}_{\text{Predicted Q-Value}} - \underbrace{(r + \max_a Q_{old}(s', a))}_{\text{Target Q-Value}}\Big)^2$$

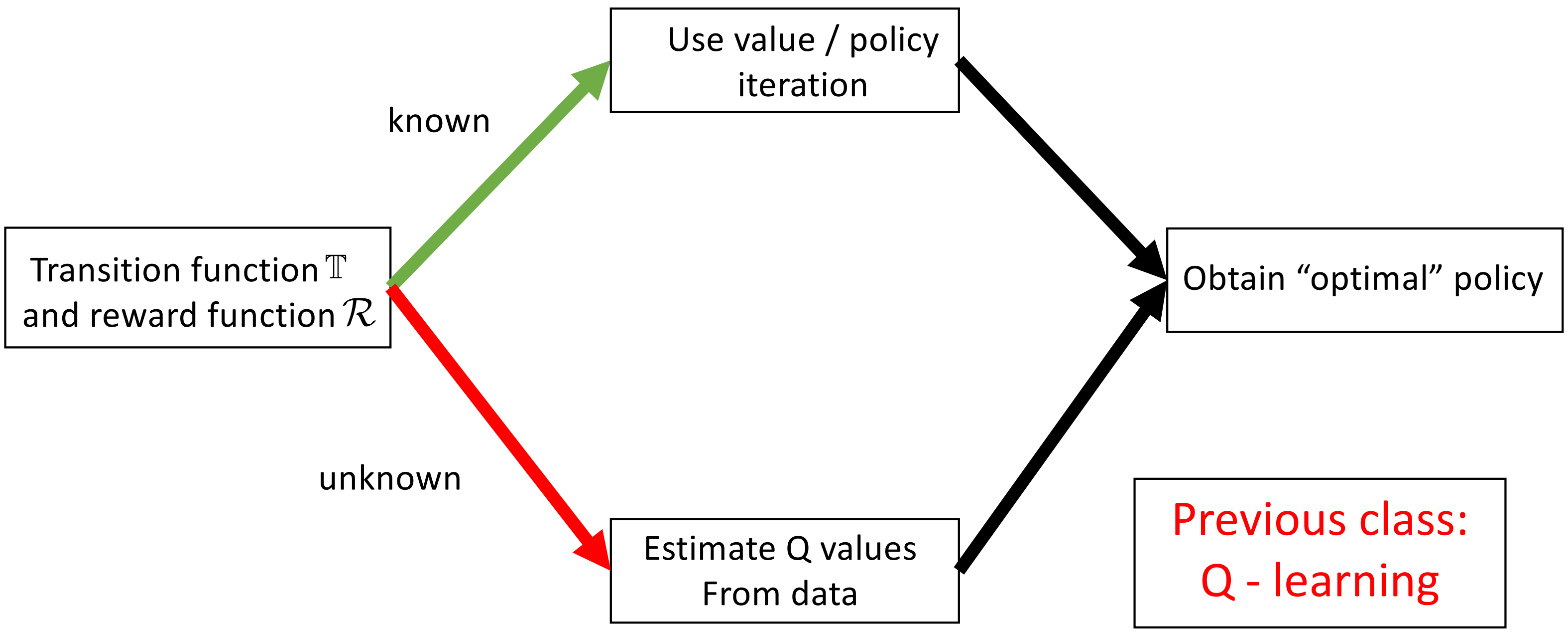
- Act according optimally according to the learnt Q function:

$$\pi(s) = \underbrace{\arg\max_{a \in \mathcal{A}} Q(s, a)}_{\text{Pick action with best Q value}}$$

Getting to the optimal policy

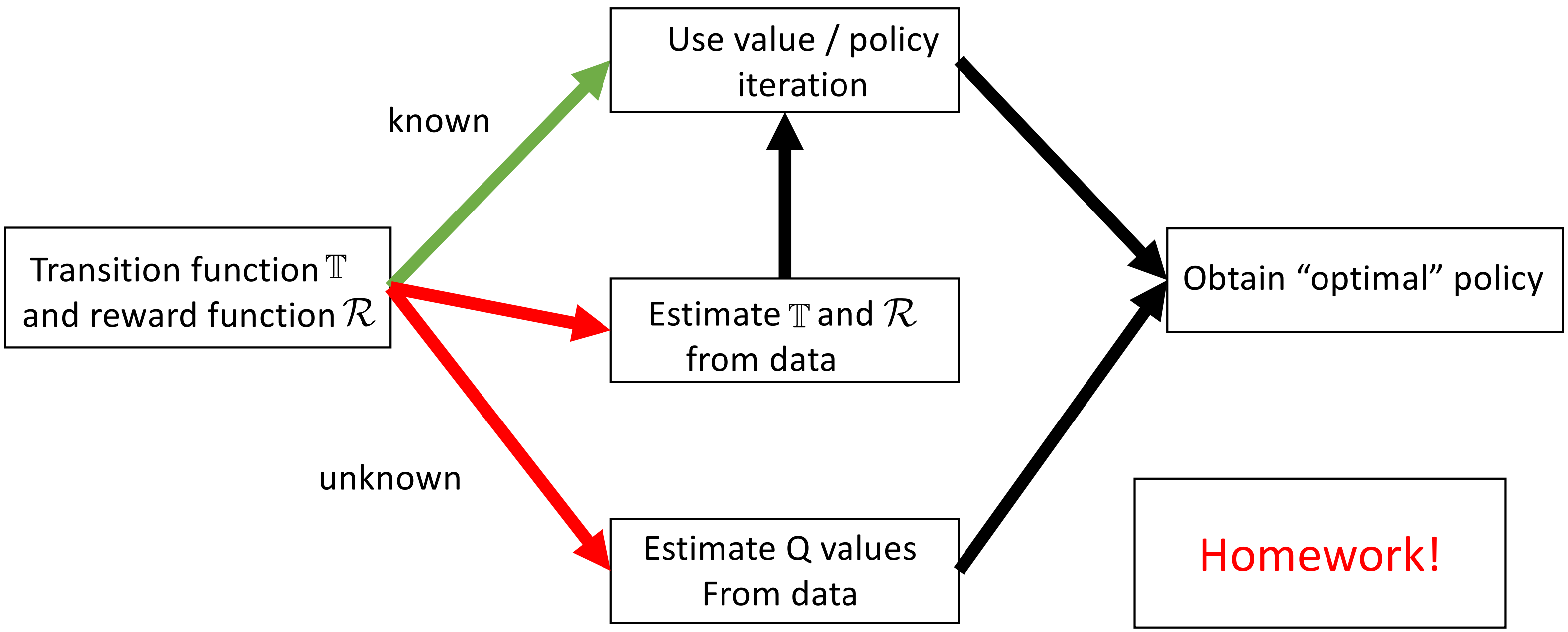


Getting to the optimal policy

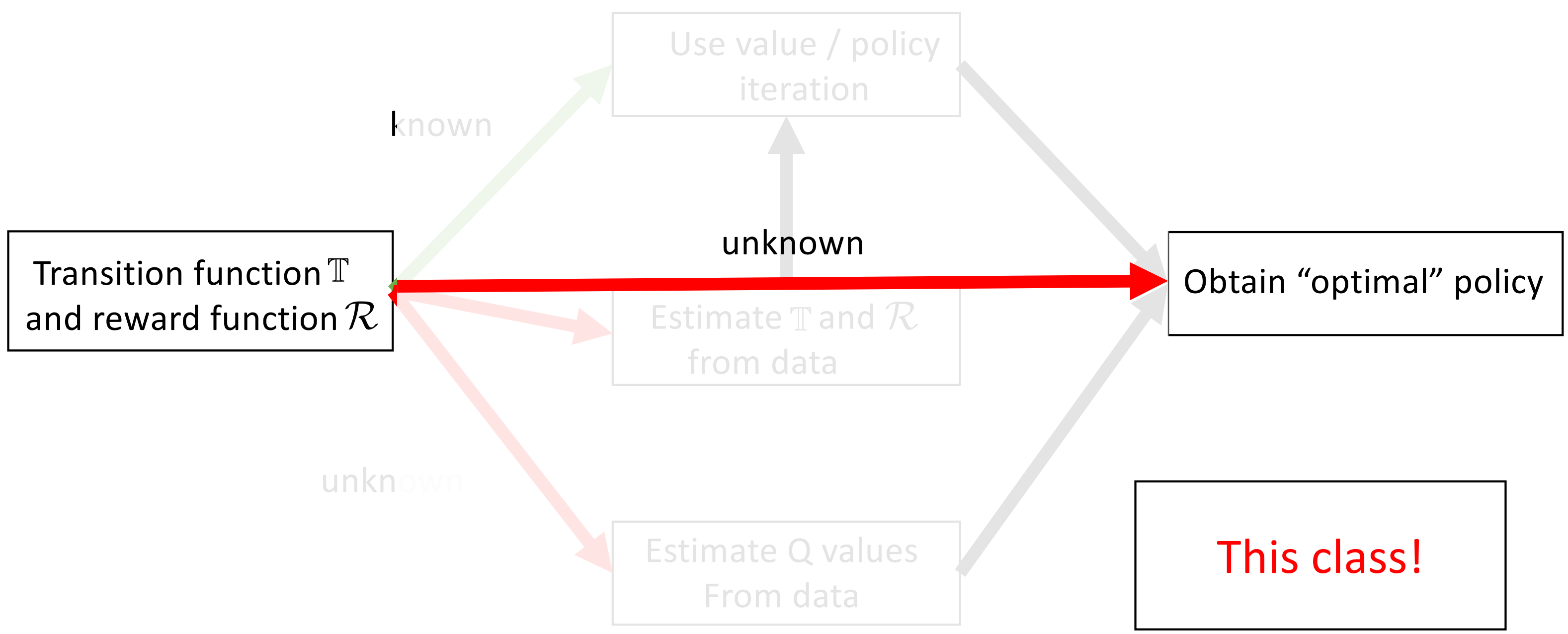


Previous class:
Q - learning

Getting to the optimal policy



Getting to the optimal policy



Learning the optimal policy

- Class of policies defined by parameters θ

$$\pi_\theta(a|s) : \mathcal{S} \rightarrow \mathcal{A}$$

  - Eg: θ can be parameters of linear transformation, deep network, etc.

Learning the optimal policy

- Class of policies defined by parameters θ

$$\pi_\theta(a|s) : \mathcal{S} \to \mathcal{A}$$

  - Eg: θ can be parameters of linear transformation, deep network, etc.
- Want to maximize:

$$J(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(s_t, a_t)\right]$$

- In other words,

$$\pi^* = \arg\max_{\pi:\mathcal{S}\to\mathcal{A}} \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(s_t, a_t)\right] \Rightarrow \theta^* = \arg\max_{\theta} \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(s_t, a_t)\right]$$

Learning the optimal policy

- Class of policies defined by parameters θ

$$\pi_\theta(a|s) : \mathcal{S} \rightarrow \mathcal{A}$$

  - Eg: θ can be parameters of linear transformation, deep network, etc.
- Want to maximize:

$$J(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(s_t, a_t)\right]$$

- In other words,

$$\pi^* = \arg\max_{\pi:\mathcal{S}\rightarrow\mathcal{A}} \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(s_t, a_t)\right] \Longrightarrow \theta^* = \arg\max_{\theta} \mathbb{E}\left[\sum_{t=1}^{T} \mathcal{R}(s_t, a_t)\right]$$

Learning the optimal policy

- Slightly rewriting the notation:
  - Let $\tau = (s_0, a_0, \ldots s_T, a_T)$, the trajectory

$$p_\theta(\tau) = p_\theta(s_0, a_0, \ldots s_T, a_T)$$
$$= \prod_{t=0}^{T} p_\theta(a_t \mid s_t) \cdot p(s_{t+1} \mid s_t, a_t)$$

$$\arg\max_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)} [\mathcal{R}(\tau)]$$

Learning the optimal policy

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\mathcal{R}(\tau) \right]$$

$$= \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)} \left[\sum_{t=0}^{T} \mathcal{R}(s_t, a_t) \right]$$

Sample a few trajectories $\{\tau_i\}_{i=1}^{N}$ by acting according to π_θ

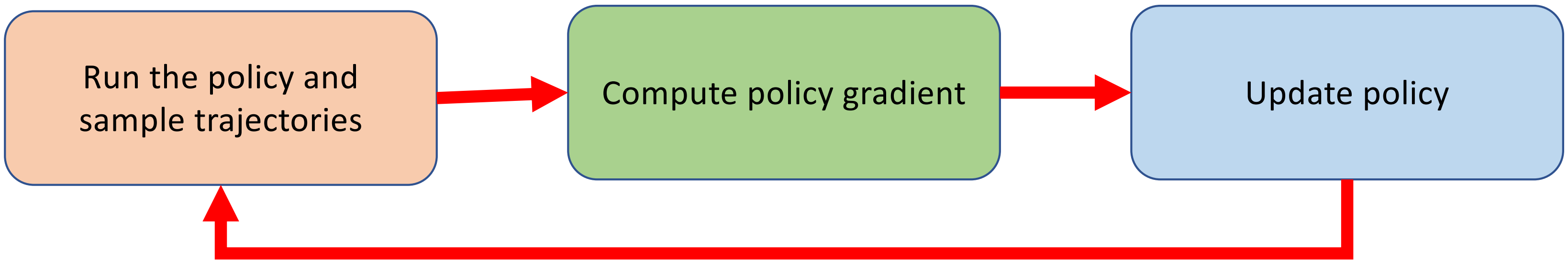
$$\approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} r(s_t^i, a_t^i)$$

REINFORCE

1. Sample trajectories $\tau_i = \{s_1, a_1, \dots s_T, a_T\}_i$ by acting according to π_θ
2. Compute policy gradient as

$$\nabla_\theta J(\theta) \approx \sum_i \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t^i) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t^i \mid a_t^i) \right]$$

3. Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$



Policy Gradients

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[\mathcal{R}(\tau)]$$

$$= \nabla_\theta \int \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Expand expectation

Policy Gradients

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[\mathcal{R}(\tau)]$$

$$= \nabla_\theta \int \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$ Expand expectation

$$= \int \nabla_\theta \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$ Exchange integration and expectation

Policy Gradients

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[\mathcal{R}(\tau)]$$

$$= \nabla_\theta \int \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Expand expectation

$$= \int \nabla_\theta \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Exchange integration and expectation

$$= \int \nabla_\theta \pi_\theta(\tau) \cdot \frac{\pi_\theta(\tau)}{\pi_\theta(\tau)} \cdot \mathcal{R}(\tau)d\tau$$

Policy Gradients

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[\mathcal{R}(\tau)]$$

$$= \nabla_\theta \int \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Expand expectation

$$= \int \nabla_\theta \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Exchange integration and expectation

$$= \int \nabla_\theta \pi_\theta(\tau) \cdot \frac{\pi_\theta(\tau)}{\pi_\theta(\tau)} \cdot \mathcal{R}(\tau)d\tau$$

$$\nabla_\theta \log \pi(\tau) = \frac{\nabla_\theta \pi(\tau)}{\pi(\tau)}$$

Policy Gradients

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[\mathcal{R}(\tau)]$$

$$= \nabla_\theta \int \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Expand expectation

$$= \int \nabla_\theta \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Exchange integration and expectation

$$= \int \nabla_\theta \pi_\theta(\tau) \cdot \frac{\pi_\theta(\tau)}{\pi_\theta(\tau)} \cdot \mathcal{R}(\tau)d\tau$$

$$= \int \pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

$$\nabla_\theta \log \pi(\tau) = \frac{\nabla_\theta \pi(\tau)}{\pi(\tau)}$$

Policy Gradients

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[\mathcal{R}(\tau)]$$

$$= \nabla_\theta \int \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Expand expectation

$$= \int \nabla_\theta \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Exchange integration and expectation

$$= \int \nabla_\theta \pi_\theta(\tau) \cdot \frac{\pi_\theta(\tau)}{\pi_\theta(\tau)} \cdot \mathcal{R}(\tau)d\tau$$

$$= \int \pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

$$\nabla_\theta \log \pi(\tau) = \frac{\nabla_\theta \pi(\tau)}{\pi(\tau)}$$

$$= \mathbb{E}_{\tau \sim p_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau)\mathcal{R}(\tau)]$$

Policy Gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) \mathcal{R}(\tau)]$$

$$\nabla_\theta \left[\log p(s_0) + \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t) + \sum_{t=1}^{T} \log p(s_{t+1} \mid s_t, a_t) \right]$$

$$\begin{aligned} p_\theta(\tau) &= p_\theta(s_0, a_0, \ldots s_T, a_T) \\ &= \prod_{t=0}^{T} p_\theta(a_t \mid s_t) \cdot p(s_{t+1} \mid s_t, a_t) \end{aligned}$$

Policy Gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[\underbrace{\nabla_\theta \log \pi_\theta(\tau)}\mathcal{R}(\tau)]$$

$$\nabla_\theta \left[\cancel{\log p(s_0)} + \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t) + \sum_{t=1}^{T} \cancel{\log p(s_{t+1} \mid s_t, a_t)} \right]$$

Doesn't depend on
Transition probabilities!

Policy Gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[\underbrace{\nabla_\theta \log \pi_\theta(\tau)}\mathcal{R}(\tau)]$$

$$\nabla_\theta \left[\log p(s_0) + \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t) + \sum_{t=1}^{T} \log p(s_{t+1} \mid s_t, a_t) \right]$$

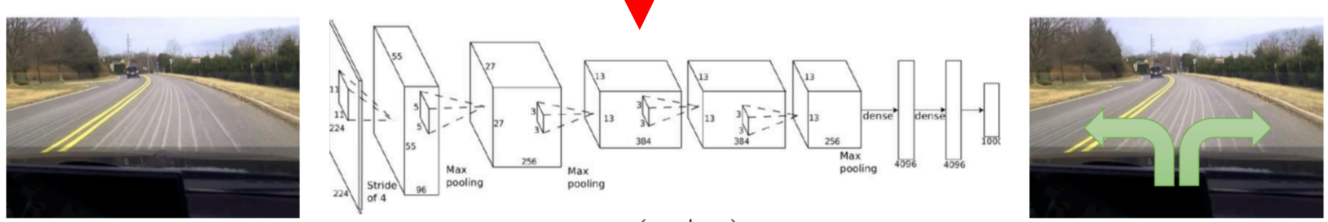
$$= \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t, a_t) \right]$$

Policy Gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[\underbrace{\nabla_\theta \log \pi_\theta(\tau)}\mathcal{R}(\tau)]$$

$$\nabla_\theta \left[\cancel{\log p(s_0)} + \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t) + \sum_{t=1}^{T} \cancel{\log p(s_{t+1} \mid s_t, a_t)} \right]$$

$$= \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t, a_t) \right]$$



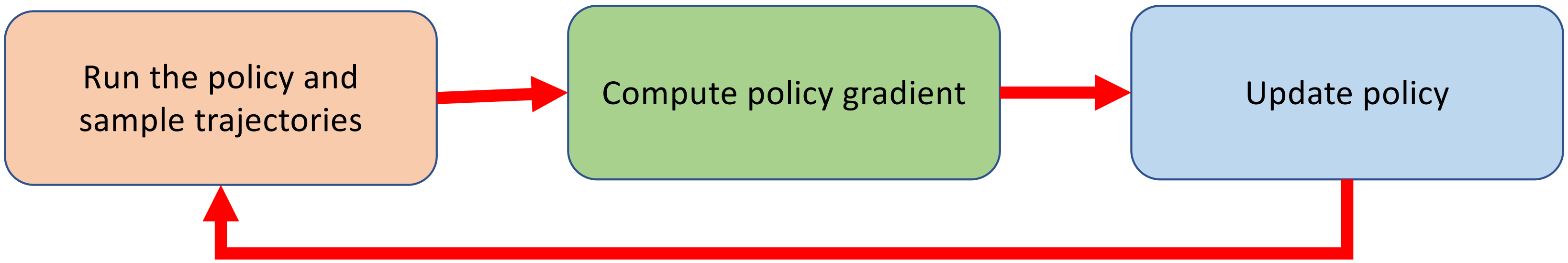
$\mathbf{s}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ $\mathbf{a}_t$

REINFORCE

1. Sample trajectories $\tau_i = \{s_1, a_1, \ldots s_T, a_T\}_i$ by acting according to π_θ
2. Compute policy gradient as

$$\nabla_\theta J(\theta) \approx \sum_i \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t^i) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t^i \mid a_t^i) \right]$$

3. Update policy $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$



Slide credit: Sergey Levine

Pong from pixels

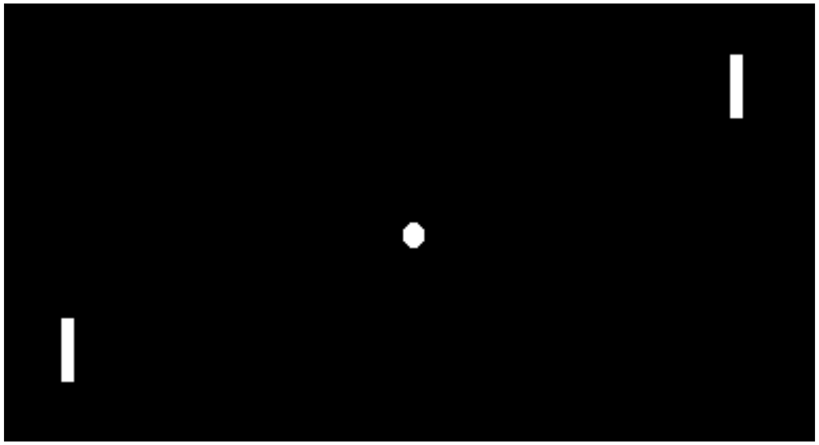


Image Credit: http://karpathy.github.io/2016/05/31/rl/

Pong from pixels

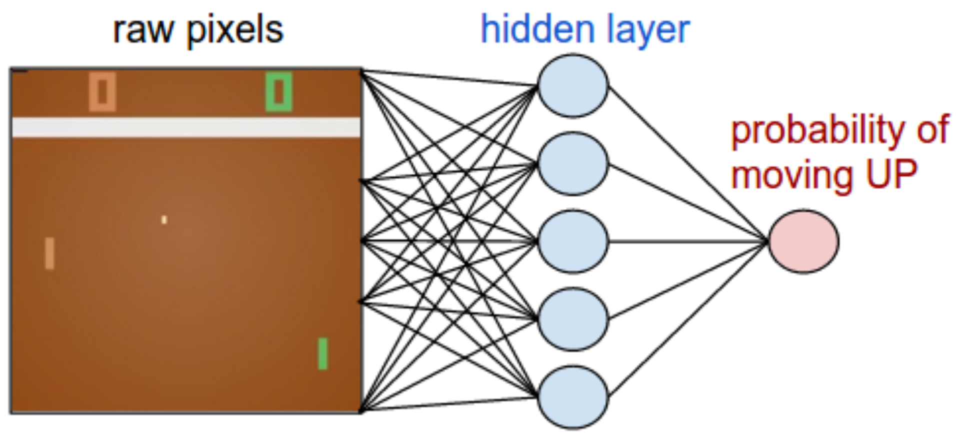


Image Credit: http://karpathy.github.io/2016/05/31/rl/

Pong from pixels

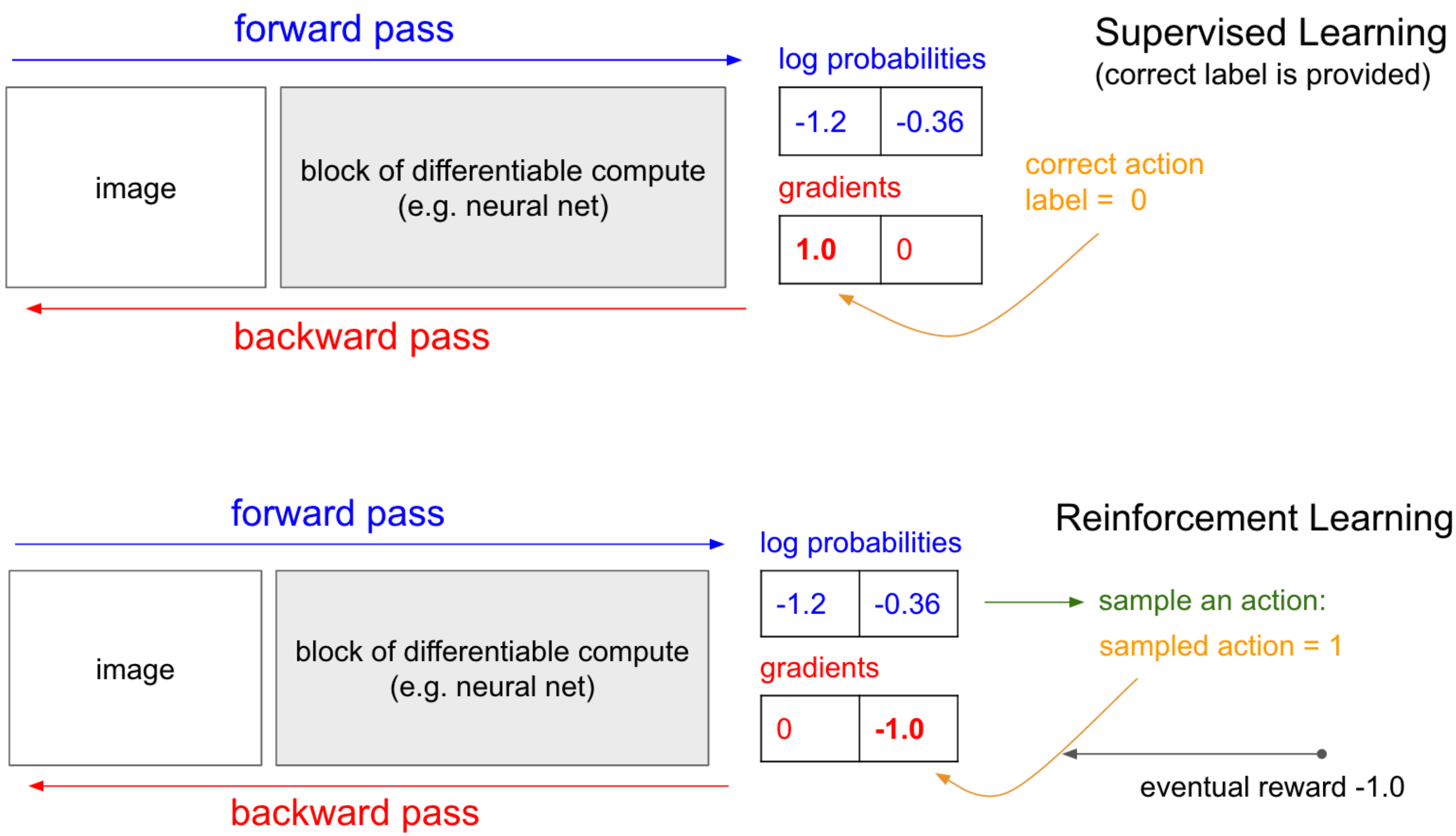
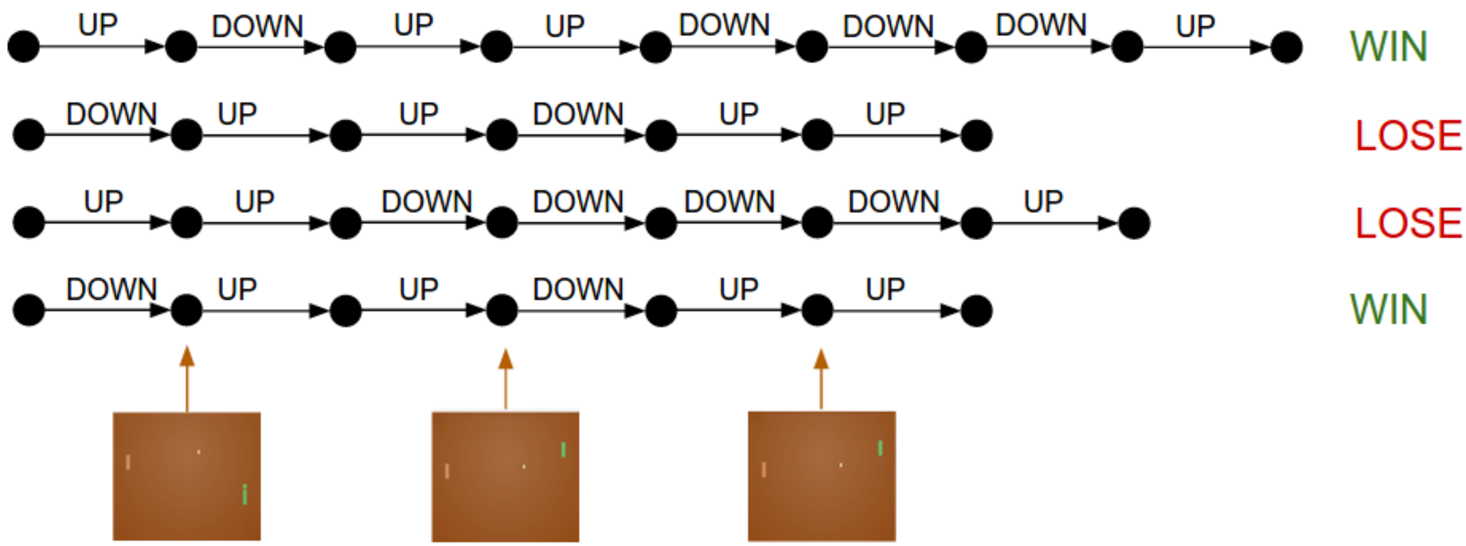


Image Credit: http://karpathy.github.io/2016/05/31/rl/

Intuition


UP
DOWN
UP
UP
DOWN
DOWN
DOWN
UP
WIN
DOWN
UP
UP
DOWN
UP
UP
LOSE
UP
UP
DOWN
DOWN
DOWN
DOWN
UP
LOSE
DOWN
UP
UP
DOWN
UP
UP
WIN

Policy Gradients

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[\underbrace{\nabla_\theta \log \pi_\theta(\tau)}\mathcal{R}(\tau)]$$

$$\nabla_\theta \left[\cancel{\log p(s_0)} + \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t) + \sum_{t=1}^{T} \cancel{\log p(s_{t+1} \mid s_t, a_t)} \right]$$

$$= \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t, a_t) \right]$$

Formalizes notion of "trial and error":

- If reward is high, probability of actions seen is increased
- If reward is low, probability of actions seen is reduced

Issues with Policy Gradients

- Credit assignment is hard!
  - Which specific action led to increase in reward
  - Suffers from high variance → leading to unstable training

Issues with Policy Gradients

- Credit assignment is hard!
  - Which specific action led to increase in reward
  - Suffers from high variance → leading to unstable training
- How to reduce the variance?
  - Subtract a constant from the reward!

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t, a_t) - b \right]$$

Issues with Policy Gradients

- Credit assignment is hard!
  - Which specific action led to increase in reward
  - Suffers from high variance → leading to unstable training
- How to reduce the variance?
  - Subtract a constant from the reward!

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t, a_t) - b \right]$$

  - Why does it work?
  - What is the best choice of b?

} Homework!

Taking a step back

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)} \left[\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot \sum_{t=1}^{T} \mathcal{R}(s_t, a_t) \right]$$

Policy Evaluation
(Recall Policy iteration)

- REINFORCE: Evaluate and update policy based on Monte-Carlo estimates of the total reward – very noisy!
- Other ways of policy evaluation?
  - If we had the Q function, we could have used it!

Actor-Critic

- Learn both policy and Q function
  - Use the “actor” to sample trajectories
  - Use the Q function to “evaluate” or “critic” the policy

Actor-Critic

- Learn both policy and Q function
  - Use the “actor” to sample trajectories
  - Use the Q function to “evaluate” or “critic” the policy

- REINFORCE: $\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) \mathcal{R}(s, a) \right]$

- Actor-critic: $\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \right]$

Actor-Critic

- Learn both policy and Q function
  - Use the "actor" to sample trajectories
  - Use the Q function to "evaluate" or "critic" the policy
- REINFORCE: $\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a \sim \pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)\mathcal{R}(s,a)\right]$
- Actor-critic: $\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a \sim \pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)Q^{\pi_\theta}(s,a)\right]$
- Q function is unknown too! Update using $\mathcal{R}(s,a)$

Actor-Critic

- Initialize s, θ (policy network) and β (Q network)

Actor-Critic

- Initialize s, θ (policy network) and β (Q network)
- sample action $a \sim \pi_\theta(\cdot|s)$

Actor-Critic

- Initialize s, θ (policy network) and β (Q network)
- sample action $a \sim \pi_\theta(\cdot|s)$
- For each step:
  - Sample reward $\mathcal{R}(s, a)$ and next state $s' \sim p(s'|s, a)$

Actor-Critic

- Initialize s, θ (policy network) and β (Q network)
- sample action $a \sim \pi_\theta(\cdot|s)$
- For each step:
  - Sample reward $\mathcal{R}(s,a)$ and next state $s' \sim p(s'|s,a)$
  - evaluate "actor" using "critic" $Q_\beta(s,a)$

Actor-Critic

- Initialize s, θ (policy network) and β (Q network)
- sample action $a \sim \pi_\theta(\cdot|s)$
- For each step:
  - Sample reward $\mathcal{R}(s,a)$ and next state $s' \sim p(s'|s,a)$
  - evaluate "actor" using "critic" $Q_\beta(s,a)$ and update policy:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a \mid s) Q_\beta(s,a)$$

Actor-Critic

- Initialize s, θ (policy network) and β (Q network)
- sample action $a \sim \pi_\theta(\cdot|s)$
- For each step:
  - Sample reward $\mathcal{R}(s, a)$ and next state $s' \sim p(s'|s, a)$
  - evaluate "actor" using "critic" $Q_\beta(s, a)$ and update policy:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a \mid s) Q_\beta(s, a)$$

  - Update "critic":
    - Recall Q-learning

$$\text{MSE Loss} := \left(Q_{new}(s, a) - (r + \max_a Q_{old}(s', a)) \right)^2$$

Actor-Critic

- Initialize s, θ (policy network) and β (Q network)
- sample action $a \sim \pi_\theta(\cdot|s)$
- For each step:
  - Sample reward $\mathcal{R}(s, a)$ and next state $s' \sim p(s'|s, a)$
  - evaluate "actor" using "critic" $Q_\beta(s, a)$ and update policy:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a \mid s) Q_\beta(s, a)$$

  - Update "critic":
    - Recall Q-learning

$$\text{MSE Loss} := \left(Q_{new}(s, a) - (r + \max_a Q_{old}(s', a))\right)^2$$

    - Update β Accordingly
- $a \leftarrow a', s \leftarrow s'$

Actor-critic

- In general, replacing the policy evaluation or the "critic" leads to different flavors of the actor-critic
  - REINFORCE:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) \mathcal{R}(s, a) \right]$$

  - Q – Actor Critic

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s, a) \right]$$

Actor-critic

- In general, replacing the policy evaluation or the “critic” leads to different flavors of the actor-critic
  - REINFORCE:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)\mathcal{R}(s,a)\right]$$

  - Q – Actor Critic

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)Q^{\pi_\theta}(s,a)\right]$$

  - Advantage Actor Critic:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)A^{\pi_\theta}(s,a)\right]$$

$$= Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$$

“how much better is an action than expected?

Summary

- Policy Learning:
  - Policy gradients
  - REINFORCE
  - Reducing Variance (Homework!)
- Actor-Critic:
  - Other ways of performing "policy evaluation"
  - Variants of Actor-critic