Policy-Based Reinforcement Learning

Shusen Wang

Policy Function Approximation

Action-Value Function

Definition: Discounted return (aka cumulative discounted future reward).

- $R_t = r_t + \gamma\, r_{t+1} + \gamma^2\, r_{t+2} + \gamma^3\, r_{t+3} + \cdots$ (to infinity.)

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\,[R_t | s_t, a_t, \pi]$.

- Taken w.r.t. actions $a_{t+1}, a_{t+2}, a_{t+3}, \cdots$ and states $s_{t+1}, s_{t+2}, s_{t+3}, \cdots$
- Actions are randomly sampled: $a_t \sim \pi(\cdot\,| s_t)$. (Policy function.)
- States are randomly sampled: $s_{t+1} \sim p(\cdot\,| s_t, a_t)$. (State transition.)

State-Value Function

Definition: Discounted return (aka cumulative discounted future reward).

- $R_t = r_t + \gamma\, r_{t+1} + \gamma^2\, r_{t+2} + \gamma^3\, r_{t+3} + \cdots$ (to infinity.)

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[R_t | s_t, a_t, \pi\right]$.

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_{a\sim\pi(\cdot|s_t)}\left[Q_\pi(s_t, a)\right] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$.

  - Integrate out action a.
  - Given s_t and π, state-value function can tell the expected return.

State-Value Function

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_{a \sim \pi(\cdot|s_t)}\left[Q_\pi(s_t, a)\right] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$.

Policy Function Approximation

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_{a \sim \pi(\cdot|s_t)}\left[Q_\pi(s_t, a)\right] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a).$

Policy-based learning: Learn a policy π that maximizes $\mathbb{E}_s[V_\pi(s)]$.

Policy Function Approximation

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_{a\sim\pi(\cdot|s_t)}[Q_\pi(s_t, a)] = \sum_a \pi(a|s_t) \cdot Q_\pi(s_t, a)$.

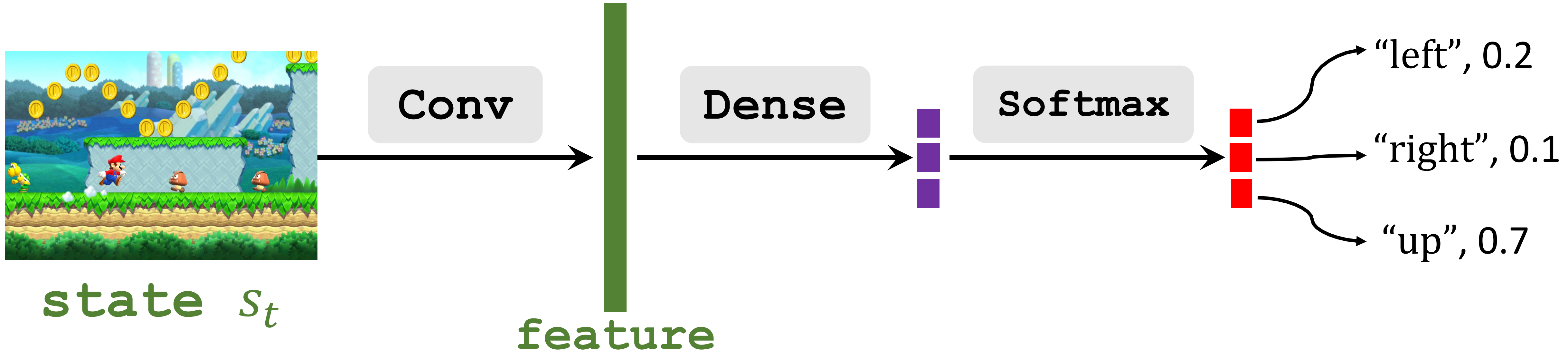
Policy-based learning: Learn a policy π that maximizes $\mathbb{E}_s[V_\pi(s)]$.

Policy network: Use a neural net to approximate $\pi(a|s)$.

- Use neural net $\pi(a|s; \boldsymbol{\theta})$ to approximate $\pi(a|s)$.
- $\boldsymbol{\theta}$: trainable parameters of the neural net.

Policy Network $\pi(a|s,\boldsymbol{\theta})$

- $\pi(a|s;\boldsymbol{\theta}) = 0.2$ means that observing s, the agent shall take action a with probability 0.2.
- Let $\mathcal{A}$ be the set all actions, e.g., $\mathcal{A} = \{\text{"left"}, \text{"right"}, \text{"up"}\}$.
- $\sum_{a\in\mathcal{A}} \pi(a|s,\boldsymbol{\theta}) = 1$. (That is why we use softmax activation.)



Policy Gradient

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_{a} \pi(a|s;\boldsymbol{\theta}) \cdot Q_{\pi}(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_{a} \frac{\partial \pi(a|s;\boldsymbol{\theta}) \cdot Q_{\pi}(s,a)}{\partial\, \boldsymbol{\theta}}$

Push the differentiation into the summation.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$

Q_π is independent of $\boldsymbol{\theta}$.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}} = \sum_a \boxed{\frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \cdot Q_\pi(s,a)$$
$$= \sum_a \boxed{\pi(a|s;\boldsymbol{\theta}) \cdot \frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \cdot Q_\pi(s,a)$$

- Chain rule: $\frac{\partial \log[f(x)]}{\partial\, x} = \frac{\partial\, f(x)}{\partial\, x} \cdot \frac{1}{f(x)}$.
- Thus $\frac{\partial\, f(x)}{\partial\, x} = f(x) \cdot \frac{\partial \log[f(x)]}{\partial\, x}$.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $$\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)}{\partial\, \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$
$$= \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot \frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)$$
$$= \mathbb{E}_a \left[\frac{\partial \log \pi(a|s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a) \right].$$

The expectation is taken w.r.t. the random variable $a \sim \pi(\cdot\,|s;\boldsymbol{\theta})$.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{a\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(a|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)\right]$.

Policy Gradient

Definition: Approximate state-value function.

- $V(s;\boldsymbol{\theta}) = \sum_a \pi(a|s;\boldsymbol{\theta}) \cdot Q_\pi(s,a)$.

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{a\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(a|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)\right]$.

- Policy gradient ascent:

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \beta \cdot \frac{\partial\, V(s_t;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}.$$

- Increasing the state-value means improving the policy.

Policy Gradient

Policy gradient: Derivative of $V(s; \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{a\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(a|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, a)\right].$

Question: How to compute the policy gradient $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$?

Policy Gradient

Policy gradient: Derivative of $V(s;\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$.

- $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_{a\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(a|s,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,a)\right]$.

Question: How to compute the policy gradient $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$?

- Sample actions: $a^{(1)}, a^{(2)}, \cdots, a^{(k)} \sim \pi(\cdot\,|s;\boldsymbol{\theta})$.
  - (The agent does not actually perform the actions.)
  - Sample only one action (i.e., $k = 1$) also works.
- Compute $\tilde{\mathbf{g}}(\boldsymbol{\theta}) = \frac{1}{k}\sum_{i=1}^{k} \frac{\partial \log \pi(a^{(1)}|s_t,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi\left(s_t, a^{(1)}\right)$.
- $\tilde{\mathbf{g}}(\boldsymbol{\theta})$ is unbiased estimate of $\frac{\partial\, V(s;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

Update policy network using policy gradient

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot \mid s_t; \boldsymbol{\theta}_t)$.

The agent may not actually perform action a_t.

Update policy network using policy gradient

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot\,|s_t;\boldsymbol{\theta}_t)$.
3. Compute $q_t \approx Q_\pi(s_t, a_t)$ (some estimate).
4. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t|s_t,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}\,\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
5. (Stochastic) policy gradient: $\tilde{\mathbf{g}}(\boldsymbol{\theta}_t) \approx q_t \cdot \mathbf{d}_{\theta,t}$.
6. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \tilde{\mathbf{g}}(\boldsymbol{\theta}_t)$.

Update policy network using policy gradient

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot\,|s_t;\boldsymbol{\theta}_t)$.
3. $\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?
4. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t|s_t,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
5. (Stochastic) policy gradient: $\tilde{\mathbf{g}}(\boldsymbol{\theta}_t) \approx q_t \cdot \mathbf{d}_{\theta,t}$.
6. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \tilde{\mathbf{g}}(\boldsymbol{\theta}_t)$.

Update policy network using policy gradient

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot\,|\,s_t;\boldsymbol{\theta}_t)$.
3. $\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?

Option 1: Monte Carlo.

- Play the game to the end and generate the trajectory:

$$s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, \quad \cdots \quad, s_T, a_T, r_T.$$

- Compute the discounted return $R_t = \sum_{k=t}^{T} \gamma^k r_k$.
- Since $Q_\pi(s_t, a_t) = \mathbb{E}[R_t]$, we can use R_t to approximate $Q_\pi(s_t, a_t)$.

Update policy network using policy gradient

1. Observe the state s_t.
2. Randomly sample action a_t according to $\pi(\cdot\,|s_t;\boldsymbol{\theta}_t)$.
3. Compute $q_t \approx Q_\pi(s_t, a_t)$ (some estimate). How?

Option 2: Approximate Q_π using a neural network.

- This leads to the actor-critic method.

Summary

Policy-Based Method

- If a good policy function $\pi(a|s)$ is known, the agent can follow the policy: $a_t \sim \pi(\cdot\,|s_t)$.
- Approximate policy function $\pi(a|s)$ by policy network $\pi(a|s;\boldsymbol{\theta})$.
- Learn the policy network by policy gradient.
- Policy gradient algorithm learn $\boldsymbol{\theta}$ that maximizes $\mathbb{E}_s[V(s;\boldsymbol{\theta})]$.