Policy-Based Reinforcement Learning

Adapted from slides by Shusen Wang at Stevens Institute of Technology

http://wangshusen.github.io/

Policy Function Approximation

Policy Function $\pi(a|s)$

- Policy function $\pi(a|s)$ is a probability density function (PDF).
- It takes state s as input.
- It output the probabilities for all the actions, e.g.,

$$\pi(\mathrm{left}|s) = 0.2,$$
$$\pi(\mathrm{right}|s) = 0.1,$$
$$\pi(\mathrm{up}|s) = 0.7.$$

- Randomly sample action a random drawn from the distribution.

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

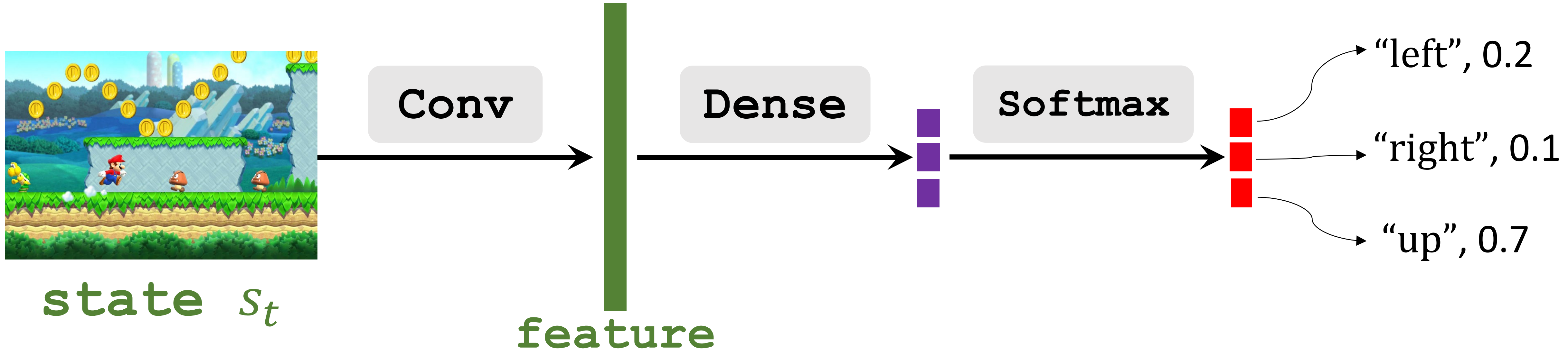
Policy network: Use a neural net to approximate $\pi(a|s)$.

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

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

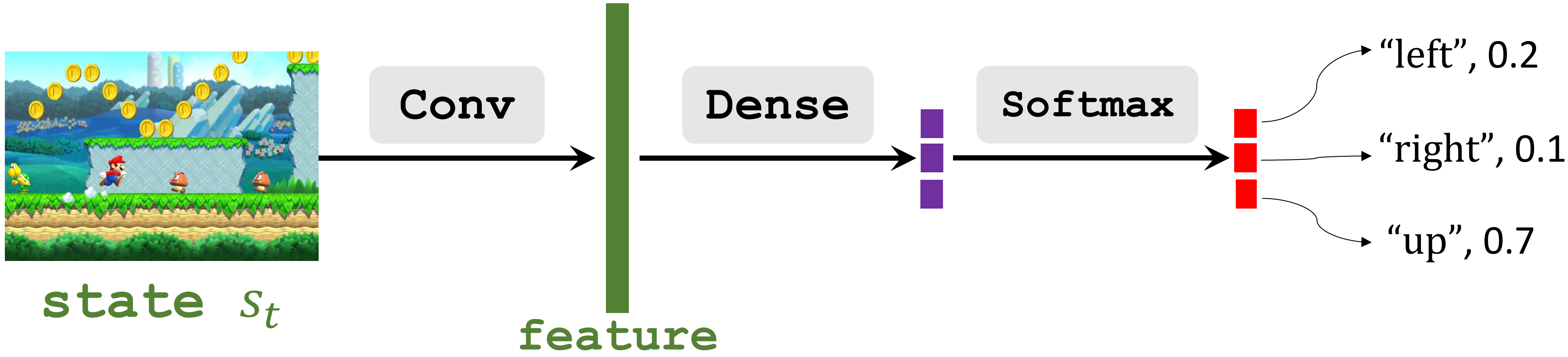
Policy network: Use a neural net to approximate $\pi(a|s)$.

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



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

- $\sum_{a \in \mathcal{A}} \pi(a|s;\boldsymbol{\theta}) = 1.$
- Here, $\mathcal{A} = \{\text{“left”}, \text{“right”}, \text{“up”}\}$ is the set all actions.
- That is why we use softmax activation.



State-Value Function Approximation

Action-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

- The return depends on actions $A_t, A_{t+1}, A_{t+2}, \cdots$ and states $S_t, S_{t+1}, S_{t+2}, \cdots$
- Actions are random: $\mathbb{P}[A = a \mid S = s] = \pi(a|s)$. (Policy function.)
- States are random: $\mathbb{P}[S' = s' | S = s, A = a] = p(s'|s, a)$. (State transition.)

Action-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

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

The expectation is taken w.r.t.
actions $A_{t+1}, A_{t+2}, A_{t+3}, \cdots$
and states $S_{t+1}, S_{t+2}, S_{t+3}, \cdots$

State-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t \mid S_t = s_t, A_t = a_t\right].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\left[Q_\pi(s_t, A)\right]$

Integrate out action $A \sim \pi(\cdot \mid s_t)$.

State-Value Function

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

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

Definition: State-value function.

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

Integrate out action $A \sim \pi(\cdot\,|s_t)$.

Policy-Based Reinforcement Learning

Definition: State-value function.

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

Approximate state-value function.

- Approximate policy function $\pi(a|s_t)$ by policy network $\pi(a|s_t; \boldsymbol{\theta})$.
- Approximate value function $V_\pi(s_t)$ by:

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

Policy-Based Reinforcement Learning

Definition: Approximate state-value function.

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

Policy-based learning: Learn $\boldsymbol{\theta}$ that maximizes $J(\boldsymbol{\theta}) = \mathbb{E}_{S}[V(S; \boldsymbol{\theta})]$.

Policy-Based Reinforcement Learning

Definition: Approximate state-value function.

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

Policy-based learning: Learn $\boldsymbol{\theta}$ that maximizes $J(\boldsymbol{\theta}) = \mathbb{E}_S[V(S; \boldsymbol{\theta})]$.

How to improve $\boldsymbol{\theta}$? Policy gradient ascent!

- Update policy by: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \cdot \boxed{\nabla J(\boldsymbol{\theta})}$.

Policy gradient

Policy Gradient

Reference

- Sutton and others: Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 2000.

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. $\boldsymbol{\theta}$.

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

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. $\boldsymbol{\theta}$.

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

Policy Gradient

Note: This derivation is over-simplified and not rigorous.

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. $\boldsymbol{\theta}$.

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

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. $\boldsymbol{\theta}$.

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

- Chain rule: $\frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \frac{1}{\pi(\boldsymbol{\theta})} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.
- ➔ $\pi(\boldsymbol{\theta}) \cdot \frac{\partial \log[\pi(\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \cancel{\pi(\boldsymbol{\theta})} \cdot \cancel{\frac{1}{\pi(\boldsymbol{\theta})}} \cdot \frac{\partial\, \pi(\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

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. $\boldsymbol{\theta}$.

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

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

Policy Gradient

Policy gradient:

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

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{(s,a)\sim\pi(\cdot;\boldsymbol{\theta})}[\nabla \log \pi(a|s,\boldsymbol{\theta}) \cdot Q_\pi(s,a)]$$

Calculate Policy Gradient

Policy Gradient: $\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{(s,a)\sim\pi(\cdot;\boldsymbol{\theta})}[\nabla \log \pi(a|s, \boldsymbol{\theta}) \cdot Q_\pi(s, a)]$.

Calculate Policy Gradient

Policy Gradient: $\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{(s,a)\sim\pi(\cdot;\boldsymbol{\theta})}\left[\nabla \log \pi(a|s,\boldsymbol{\theta}) \cdot Q_\pi(s,a)\right]$.

1. Randomly sample trajectories (s_t, a_t) according to $\pi(\cdot;\boldsymbol{\theta})$.
2. Calculate $\nabla \bar{J}(\boldsymbol{\theta})$ using (s, a) pairs from the trajectories.
3. Update parameters by $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \nabla \bar{J}(\boldsymbol{\theta})$.

Update policy network using policy gradient

Algorithm

1. Randomly sample trajectories (s_t, a_t) according to $\pi(\cdot; \boldsymbol{\theta})$.
2. Compute $q_t \approx Q_\pi(s_t, a_t)$ (some estimate).
3. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t | s_t, \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
4. (Approximate) policy gradient: $\nabla \bar{J}(\boldsymbol{\theta})$.
5. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \cdot \nabla \bar{J}(\boldsymbol{\theta})$.

Algorithm

1. Randomly sample trajectories (s_t, a_t) according to $\pi(\cdot; \boldsymbol{\theta})$.
2. $\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?
3. Differentiate policy network: $\mathbf{d}_{\theta,t} = \frac{\partial \log \pi(a_t|s_t,\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$.
4. (Approximate) policy gradient: $\nabla \bar{J}(\boldsymbol{\theta})$.
5. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \cdot \nabla \bar{J}(\boldsymbol{\theta})$.

Algorithm

$\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?

Option 1: REINFORCE.

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

$$s_1, a_1, r_1, \; s_2, a_2, r_2, \quad \cdots \quad , s_T, a_T, r_T.$$

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

Algorithm

$\boxed{\text{Compute } q_t \approx Q_\pi(s_t, a_t) \text{ (some estimate).}}$ How?

Option 2: Approximate Q_π using a neural network.

- This leads to the actor-critic method.

Summary

Policy-Based Learning

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