Policy Function

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.,

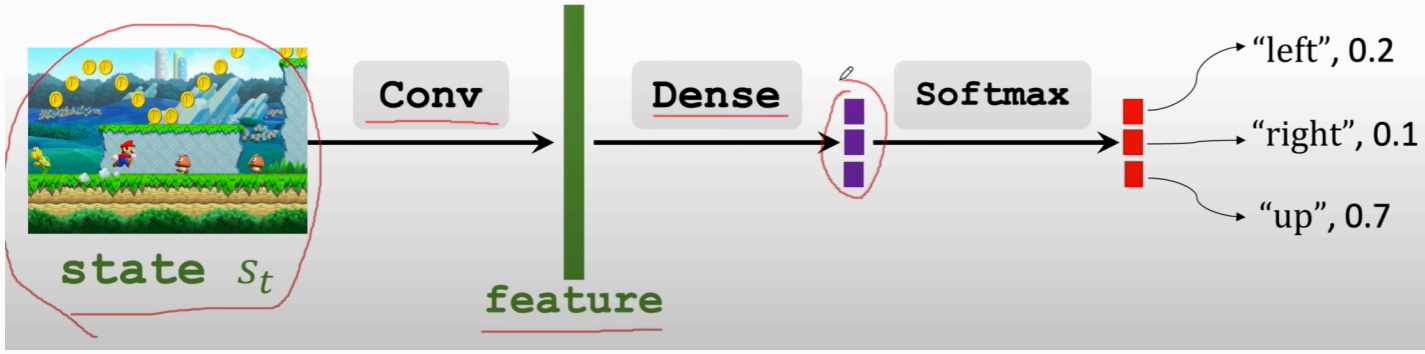
$$\pi(\text{left}|s) = 0.2,$$
$$\pi(\text{right}|s) = 0.1,$$
$$\pi(\text{up}|s) = 0.7.$$

- The agent performs an action a random drawn from the distribution.

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

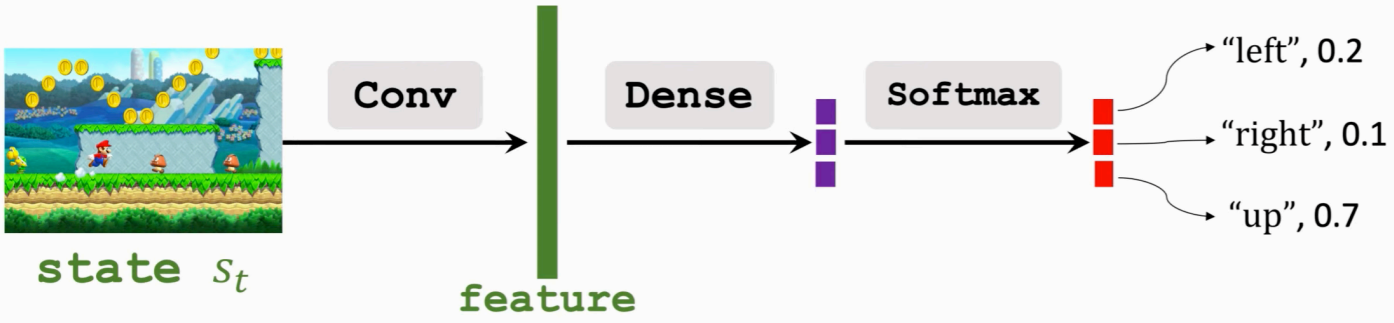
Policy network: Use a neural net to approximate $\pi(a|s)$.

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



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

- $\sum_{a \in \mathcal{A}} \pi(a|s; \boldsymbol{\theta}) = 1.$
- 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

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 $S_t, S_{t+1}, S_{t+2}, \cdots$

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.

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

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.

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

Definition: State-value function.

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

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

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!

- Observe state s.
- Update policy by: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \frac{\partial V(s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$

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. $\boldsymbol{\theta}$.

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

Pretend Q_π is independent of $\boldsymbol{\theta}$.
(It may not be true.)

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. $\boldsymbol{\theta}$.

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

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

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. $\boldsymbol{\theta}$.

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

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

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

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. $\boldsymbol{\theta}$.

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

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

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

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

Calculate Policy Gradient for Discrete Actions

If the actions are discrete, e.g., action space $\mathcal{A} = \{\text{“left”}, \text{“right”}, \text{“up”}\}$, ...

Use Form 1: $\frac{\partial V(s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_a \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot Q_\pi(s,a).$

1. Calculate $\mathbf{f}(a,\boldsymbol{\theta}) = \frac{\partial \pi(a|s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot Q_\pi(s,a)$, for every action $a \in \mathcal{A}$.
2. Policy gradient: $\frac{\partial V(s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{f}(\text{“left”},\boldsymbol{\theta}) + \mathbf{f}(\text{“right”},\boldsymbol{\theta}) + \mathbf{f}(\text{“up”},\boldsymbol{\theta})$.

Calculate Policy Gradient for Continuous Actions

If the actions are continuous, e.g., action space $\mathcal{A} = [0,1]$, ...

Use Form 2: $\frac{\partial V(s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_{A\sim\pi(\cdot|s;\boldsymbol{\theta})}\left[\frac{\partial \log \pi(A|s,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot Q_\pi(s,A)\right].$

1. Randomly sample an action $\hat{a}$ according to the PDF $\pi(\cdot\,|s;\boldsymbol{\theta})$.
2. Calculate $\mathbf{g}(\hat{a},\boldsymbol{\theta}) = \frac{\partial \log \pi(\hat{a}|s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot Q_\pi(s,\hat{a})$.
3. Use $\mathbf{g}(\hat{a},\boldsymbol{\theta})$ as an approximation to the policy gradient $\frac{\partial V(s;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$.

This approach also works for discrete actions.

Algorithm

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. (Approximate) policy gradient: $\mathbf{g}(a_t, \boldsymbol{\theta}_t) = q_t \cdot \mathbf{d}_{\theta,t}$.
6. Update policy network: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \beta \cdot \mathbf{g}(a_t, \boldsymbol{\theta}_t)$.

Algorithm

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

Option 1: REINFORCE.

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

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

Algorithm

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

Option 1: REINFORCE.

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

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

- 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

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.

Policy-Based Method

- 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.
- Policy gradient algorithm learn $\boldsymbol{\theta}$ that maximizes $\mathbb{E}_S[V(S;\boldsymbol{\theta})]$.