Lecture 7: Policy Gradient. Off-Policy Algorithms

Anton Plaksin

Markov Decision Process

Markov Property

$$\mathbb{P}[S_{t+1}|S_t, A_t] = \mathbb{P}[S_{t+1}|S_1, A_1, S_2, A_2 \dots, S_t, A_t]$$

$$\mathbb{P}[R_t|S_t, A_t] = \mathbb{P}[R_t|S_1, A_1, S_2, A_2 \dots, S_t, A_t] = 1$$

Markov Decision Process $\langle \mathcal{S}, \mathcal{S}_F, \mathcal{A}, \mathcal{P}, \mathcal{P}_0, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is an infinite state space
- $\mathcal{S}_F$ is a set of final states
- $\mathcal{A}$ is an infinite action space
- $\mathcal{P}$ is an unknown transition probability function

$$\mathcal{P}(s'|s, a) = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]$$

- $\mathcal{P}_0$ is an unknown initial state probability function
- $\mathcal{R}$ is an unknown reward function

$$\mathcal{R}(s, a) = R_t \quad \Leftrightarrow \quad \mathbb{P}[R_t|S_t = s, A_t = a] = 1$$

- $\gamma \in [0, 1]$ is a discount coefficient

DQN

Initialize q-networks Q^θ, $Q^{\theta'}$ $(\theta' = \theta)$. Let $\varepsilon = 1$.
During each episode, do

- Being in state S_t, act

$$A_t \sim \pi(\cdot|S_t), \quad \pi = \varepsilon\text{-greedy}(Q^\theta)$$

  get reward R_t, done signal D_t, and next state S_{t+1}.
  Store $(S_t, A_t, R_t, D_t, S_{t+1}) \to M$

- Get a batch $\{(s_j, a_j, r_j, d_j, s'_j)\}_{j=1}^n \leftarrow M$, determine targets

$$y_j = r_j + \gamma(1 - d_j)\max_{a'} Q^{\theta'}(s'_j, a')$$

  the loss function

$$L_1(\theta) = \frac{1}{n}\sum_{j=1}^{n}\big(y_j - Q^\theta(s_j, a_j)\big)^2,$$

  and update the parameters

$$\theta \leftarrow \theta - \beta_1 \nabla_\theta L_1(\theta), \quad \theta' \leftarrow \tau\theta' + (1-\tau)\theta$$

- Decrease ε

Greedy Policy Approximation

Deterministic Policy Approximation

$$\pi^{\eta}(s) \approx \operatorname*{argmax}_{a \in \mathcal{A}} Q^{\theta}(s, a)$$

Greedy Policy Approximation

Deterministic Policy Approximation

$$\pi^{\eta}(s) \approx \operatorname*{argmax}_{a \in \mathcal{A}} Q^{\theta}(s, a)$$

$$\Downarrow$$

Action due to ε-Greedy Policy

$$A_t = \left[\pi^{\eta}(S_t) + \varepsilon Noise\right]_{\mathcal{A}},$$

Targets for Q^{θ}

$$y_j = r_j + \gamma(1 - d_j) Q^{\theta'}(s_j', \pi^{\eta}(s_j'))$$

DQN with π^η

Initialize networks Q^θ, $Q^{\theta'}$ $(\theta' = \theta)$ and π^η. Let $\varepsilon = 1$.
During each episode, do

- Being in state S_t, act

$$A_t = \left[\pi^\eta(S_t) + \varepsilon Noise\right]_{\mathcal{A}},$$

  get reward R_t, done signal D_t, and next state S_{t+1}. Store $(S_t, A_t, R_t, D_t, S_{t+1}) \to M$
- Get a batch $\{(s_j, a_j, r_j, d_j, s'_j)\}_{j=1}^n \leftarrow M$, determine targets

$$y_j = r_j + \gamma(1 - d_j)Q^{\theta'}(s'_j, \pi^\eta(s'_j))$$

  the loss function

$$L_1(\theta) = \frac{1}{n}\sum_{j=1}^n \big(y_j - Q^\theta(s_j, a_j)\big)^2,$$

  and update the parameters

$$\theta \leftarrow \theta - \beta_1 \nabla_\theta L_1(\theta), \quad \theta' \leftarrow \tau\theta' + (1-\tau)\theta$$

- Decrease ε

Policy Gradient Theorems

Policy Gradient Theorem

Let $\exists \nabla_\eta \pi^\eta(a|s)$ and $\pi^\eta(a|s) \neq 0$ for any $s \in \mathcal{S}$, $a \in \mathcal{A}$. Then

$$\nabla_\eta J(\eta) = \mathbb{E}_{s \sim \rho_{\pi^\eta}, a \sim \pi^\eta} \left[\nabla_\eta \ln \pi^\eta(a|s) q_{\pi^\eta}(s, a) \right]$$

Policy Gradient Theorems

Policy Gradient Theorem

Let $\exists \nabla_\eta \pi^\eta(a|s)$ and $\pi^\eta(a|s) \neq 0$ for any $s \in \mathcal{S}$, $a \in \mathcal{A}$. Then

$$\nabla_\eta J(\eta) = \mathbb{E}_{s \sim \rho_{\pi^\eta}, a \sim \pi^\eta} \left[\nabla_\eta \ln \pi^\eta(a|s) q_{\pi^\eta}(s,a) \right]$$

Deterministic Policy Gradient Theorem

Let $\exists \nabla_\eta \pi^\eta(s)$ and $\exists \nabla_a q_{\pi^\eta}(s,a)$. Then

$$\nabla_\eta J(\eta) = \mathbb{E}_{s \sim \rho_{\pi^\eta}} \left[\nabla_\eta q_{\pi^\eta}(s, \pi^\eta(s)) \right] = \mathbb{E}_{s \sim \rho_{\pi^\eta}} \left[\nabla_a q_{\pi^\eta}(s, \pi^\eta(s)) \nabla_\eta \pi^\eta(s) \right]$$

DDPG: Main Points

Deterministic Policy Gradient Theorem

$$\nabla_\eta J(\eta) = \mathbb{E}_{s\sim\rho_{\pi^\eta}}[\nabla_\eta q_{\pi^\eta}(s, \pi^\eta(s))]$$

DDPG: Main Points

Deterministic Policy Gradient Theorem

$$\nabla_\eta J(\eta) = \mathbb{E}_{s \sim \rho_{\pi^\eta}} [\nabla_\eta q_{\pi^\eta}(s, \pi^\eta(s))]$$

Batch Approximation

If $\pi \approx \pi^\eta$ and $Q^\theta \approx q_{\pi^\eta}$, then

$$\nabla_\eta J(\eta) \approx \nabla_\eta \left(\frac{1}{n} \sum_{j=1}^{n} Q^\theta(s_j, \pi^\eta(s_j)) \right)$$

DDPG: Main Points

Deterministic Policy Gradient Theorem

$$\nabla_\eta J(\eta) = \mathbb{E}_{s\sim\rho_{\pi^\eta}}[\nabla_\eta q_{\pi^\eta}(s, \pi^\eta(s))]$$

Batch Approximation

If $\pi \approx \pi^\eta$ and $Q^\theta \approx q_{\pi^\eta}$, then

$$\nabla_\eta J(\eta) \approx \nabla_\eta \left(\frac{1}{n} \sum_{j=1}^{n} Q^\theta(s_j, \pi^\eta(s_j)) \right)$$

Bellman Expectation Equation for q_π

$$q_{\pi^\eta}(s, a) = \mathbb{E}[R_t + \gamma q_{\pi^\eta}(S_{t+1}, \pi^\eta(S_{t+1}))|S_t = s,\ A_t = a]$$

DDPG: Main Points

Deterministic Policy Gradient Theorem

$$\nabla_\eta J(\eta) = \mathbb{E}_{s\sim\rho_{\pi^\eta}}[\nabla_\eta q_{\pi^\eta}(s, \pi^\eta(s))]$$

Batch Approximation

If $\pi \approx \pi^\eta$ and $Q^\theta \approx q_{\pi^\eta}$, then

$$\nabla_\eta J(\eta) \approx \nabla_\eta \left(\frac{1}{n} \sum_{j=1}^{n} Q^\theta(s_j, \pi^\eta(s_j)) \right)$$

Bellman Expectation Equation for q_π

$$q_{\pi^\eta}(s,a) = \mathbb{E}[R_t + \gamma q_{\pi^\eta}(S_{t+1}, \pi^\eta(S_{t+1}))|S_t = s,\ A_t = a]$$

Batch Approximation

If

$$\frac{1}{n} \sum_{j=1}^{n} \left(r_j + \gamma Q^\theta(s'_j, \pi^\eta(s'_j)) - Q^\theta(s_j, a_j) \right) \approx 0,$$

then $Q^\theta \approx q_{\pi^\eta}$

Deep Deterministic Policy Gradient (DDPG)

Initialize networks π^η, $\pi^{\eta'}$ $(\eta' = \eta)$ and Q^θ, $Q^{\theta'}$ $(\theta' = \theta)$. Let $\varepsilon = 1$.
During each episode, do

- Being in state S_t, act

$$A_t = \left[\pi^\eta(S_t) + \varepsilon Noise\right]_{\mathcal{A}},$$

  get reward R_t, done signal D_t, and next state S_{t+1}. Store $(S_t, A_t, R_t, D_t, S_{t+1}) \to M$

- Get a batch $\{(s_j, a_j, r_j, d_j, s'_j)\}_{j=1}^n \leftarrow M$, determine targets

$$y_j = r_j + \gamma(1 - d_j) Q^{\theta'}(s'_j, \pi^{\eta'}(s'_j))$$

  the loss functions

$$L_1(\theta) = \frac{1}{n}\sum_{j=1}^n \left(y_j - Q^\theta(s_j, a_j)\right)^2, \quad L_2(\eta) = \frac{1}{n}\sum_{j=1}^n Q^\theta(s_j, \pi^\eta(s_j))$$

  and update the parameters

$$\theta \leftarrow \theta - \beta_1 \nabla_\theta L_1(\theta), \quad \eta \leftarrow \eta + \beta_2 \nabla_\eta L_2(\eta),$$

$$\theta' \leftarrow \tau\theta' + (1 - \tau)\theta, \quad \eta' \leftarrow \tau\eta' + (1 - \tau)\eta$$

- Decrease ε

DDPG Improvement

- Using fixed exploration noise $\mathcal{N}(0, \sigma)$

DDPG Improvement

- Using fixed exploration noise $\mathcal{N}(0, \sigma)$
- Using two q-function approximations Q^{θ_i}, $i = 1, 2$ in targets:

$$y_j = r_j + \gamma(1 - d_j) \min_{i=1,2} Q^{\theta'_i}(s_j, \pi^{\eta}(s_j))$$

DDPG Improvement

- Using fixed exploration noise $\mathcal{N}(0, \sigma)$
- Using two q-function approximations Q^{θ_i}, $i = 1, 2$ in targets:

$$y_j = r_j + \gamma(1 - d_j) \min_{i=1,2} Q^{\theta_i'}(s_j, \pi^\eta(s_j))$$

- Using clipped noisy actions in in targets:

$$y_j = r_j + \gamma(1 - d_j) \min_{i=1,2} Q^{\theta_i'}(s_j, a_j), \;\; a_j = \pi^\eta(s_j) + [\epsilon]_c, \;\; \epsilon \sim \mathcal{N}(0, \sigma)$$

DDPG Improvement

- Using fixed exploration noise $\mathcal{N}(0, \sigma)$
- Using two q-function approximations Q^{θ_i}, $i = 1, 2$ in targets:

$$y_j = r_j + \gamma(1 - d_j) \min_{i=1,2} Q^{\theta'_i}(s_j, \pi^\eta(s_j))$$

- Using clipped noisy actions in in targets:

$$y_j = r_j + \gamma(1 - d_j) \min_{i=1,2} Q^{\theta'_i}(s_j, a_j), \;\; a_j = \pi^\eta(s_j) + [\epsilon]_c, \;\; \epsilon \sim \mathcal{N}(0, \sigma)$$

- Updates the policy less frequently than the Q-function

DDPG Improvement

- Using fixed exploration noise $\mathcal{N}(0, \sigma)$
- Using two q-function approximations Q^{θ_i}, $i = 1, 2$ in targets:

$$y_j = r_j + \gamma(1 - d_j) \min_{i=1,2} Q^{\theta'_i}(s_j, \pi^\eta(s_j))$$

- Using clipped noisy actions in in targets:

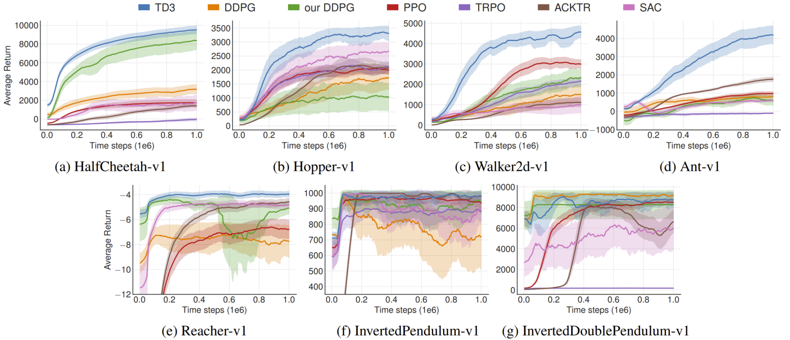
$$y_j = r_j + \gamma(1 - d_j) \min_{i=1,2} Q^{\theta'_i}(s_j, a_j), \;\; a_j = \pi^\eta(s_j) + [\epsilon]_c, \;\; \epsilon \sim \mathcal{N}(0, \sigma)$$

- Updates the policy less frequently than the Q-function

$$\Downarrow$$

Twin Delayed DDPG (TD3)

TD3 Results



Fujimoto S., Hoof H., Meger D. Addressing Function Approximation Error in Actor-Critic Methods

Markov Decision Process

Markov Property

$$\mathbb{P}[S_{t+1}|S_t, A_t] = \mathbb{P}[S_{t+1}|S_1, A_1, S_2, A_2 \dots, S_t, A_t]$$

$$\mathbb{P}[R_t|S_t, A_t] = \mathbb{P}[R_t|S_1, A_1, S_2, A_2 \dots, S_t, A_t] = 1$$

Markov Decision Process $\langle \mathcal{S}, \mathcal{S}_F, \mathcal{A}, \mathcal{P}, \mathcal{P}_0, \mathcal{R}, \gamma \rangle$

- $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ is an finite state space
- $\mathcal{S}_F \subset \mathcal{S}$ is a set of final states
- $\mathcal{A} = \{a_1, a_2, \dots, a_m\}$ is an finite action space
- $\mathcal{P}$ is an unknown transition probability function

$$\mathcal{P}(s'|s, a) = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]$$

- $\mathcal{P}_0$ is an unknown initial state probability function
- $\mathcal{R}$ is an unknown reward function

$$\mathcal{R}(s, a) = R_t \quad \Leftrightarrow \quad \mathbb{P}[R_t|S_t = s, A_t = a] = 1$$

- $\gamma \in [0, 1]$ is a discount coefficient

Soft RL Problem

RL Problem

$$\mathbb{E}\left[\sum_{i=0}^{\infty}\gamma^t\mathcal{R}(S_t, A_t)\right] \to \max_{\pi},$$

where the expectation $\mathbb{E}$ is taken by

$$S_0 \sim \mathcal{P}_0(\cdot), \quad A_t \sim \pi(\cdot|S_t), \quad S_{t+1} \sim \mathcal{P}(\cdot|S_t, A_t), \quad t = 0, 1, \ldots$$

Soft RL Problem

RL Problem

$$\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^t \mathcal{R}(S_t, A_t)\right] \to \max_{\pi},$$

where the expectation $\mathbb{E}$ is taken by

$$S_0 \sim \mathcal{P}_0(\cdot), \quad A_t \sim \pi(\cdot|S_t), \quad S_{t+1} \sim \mathcal{P}(\cdot|S_t, A_t), \quad t = 0, 1, \ldots$$

$$\Downarrow$$

Soft RL Problem

$$\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^t \Big(\mathcal{R}(S_t, A_t) + \alpha \mathcal{H}(\pi(\cdot|S_t))\Big)\right] \to \max_{\pi},$$

where the expectation $\mathbb{E}$ is taken by the same variables, $\alpha > 0$ and $\mathcal{H}(\pi(\cdot|S_t))$ is the policy entropy

$$\mathcal{H}(\pi(\cdot|s)) = -\sum_{i=1}^{m} \pi(a_i|s) \log \pi(a_i|s)$$

Soft Policy Evaluation

Soft Q-Function

$$q_\pi^\alpha(s,a) = \mathbb{E}_\pi\left[\sum_{i=0}^{\infty} \gamma^t \Big(\mathcal{R}(S_t, A_t) + \alpha\mathcal{H}(\pi(\cdot|S_t))\Big) \middle| S_0 = s, A_0 = a\right]$$

Soft Policy Evaluation

Soft Q-Function

$$q_\pi^\alpha(s,a) = \mathbb{E}_\pi\left[\sum_{i=0}^{\infty}\gamma^t\Big(\mathcal{R}(S_t,A_t) + \alpha\mathcal{H}(\pi(\cdot|S_t))\Big)\Big|S_0 = s, A_0 = a\right]$$

$$\Downarrow$$

Bellman Equation for q_π^α

$$q_\pi^\alpha(s,a) = \mathcal{R}(s,a) + \gamma\sum_{i=1}^{n}\mathcal{P}(s_i|s,a)v^\alpha(s_i),$$

$$v^\alpha(s_i) = \sum_{j=1}^{m}\pi(a_j|s_i)\Big(q_\pi^\alpha(s_i,a_j) - \alpha\log\pi(a_j|s_i)\Big)$$

Soft Policy Evaluation

Soft Q-Function

$$q_\pi^\alpha(s,a) = \mathbb{E}_\pi\left[\sum_{i=0}^{\infty}\gamma^t\Big(\mathcal{R}(S_t,A_t)+\alpha\mathcal{H}(\pi(\cdot|S_t))\Big)\Big|S_0=s,A_0=a\right]$$

$$\Downarrow$$

Bellman Equation for q_π^α

$$q_\pi^\alpha(s,a) = \mathcal{R}(s,a)+\gamma\sum_{i=1}^{n}\mathcal{P}(s_i|s,a)v^\alpha(s_i),$$

$$v^\alpha(s_i) = \sum_{j=1}^{m}\pi(a_j|s_i)\Big(q_\pi^\alpha(s_i,a_j)-\alpha\log\pi(a_j|s_i)\Big)$$

$\big($Bellman Equation for $q_\pi^\alpha\big) \to \big($ Bellman Equation for $q_\pi\big)$ as $\alpha\to 0$.

Soft Policy Improvement

KL-Divergence

$$D_{KL}\left(\pi(\cdot|s)\middle|\middle|\nu(\cdot|s)\right) = \sum_{i=1}^{m} \pi(a_i|s) \log\left(\frac{\pi(a_i|s)}{\nu(a_i|s)}\right)$$

Soft Policy Improvement

KL-Divergence

$$D_{KL}\left(\pi(\cdot|s)\middle|\middle|\nu(\cdot|s)\right) = \sum_{i=1}^{m} \pi(a_i|s) \log\left(\frac{\pi(a_i|s)}{\nu(a_i|s)}\right)$$

Soft Greedy Policy Improvement

$$\pi'(\cdot|s) = \operatorname*{argmax}_{\pi'} D_{KL}\left(\pi'(\cdot|s)\middle|\middle|\nu(\cdot|s)\right), \quad \nu(\cdot|s) = \text{Softmax}\left(\frac{1}{\alpha} q_\pi^\alpha(s,\cdot)\right)$$

Soft Policy Improvement

KL-Divergence

$$D_{KL}\Big(\pi(\cdot|s)\Big|\Big|\nu(\cdot|s)\Big) = \sum_{i=1}^{m} \pi(a_i|s) \log\left(\frac{\pi(a_i|s)}{\nu(a_i|s)}\right)$$

Soft Greedy Policy Improvement

$$\pi'(\cdot|s) = \operatorname*{argmax}_{\pi'} D_{KL}\Big(\pi'(\cdot|s)\Big|\Big|\nu(\cdot|s)\Big), \quad \nu(\cdot|s) = \text{Softmax}\left(\frac{1}{\alpha} q_\pi^\alpha(s,\cdot)\right)$$

(Soft Greedy Policy Improvement) $\rightarrow$ (Greedy Policy Improvement) as $\alpha \rightarrow 0$.

Soft Policy Improvement

KL-Divergence

$$D_{KL}\left(\pi(\cdot|s)\middle|\middle|\nu(\cdot|s)\right) = \sum_{i=1}^{m} \pi(a_i|s) \log\left(\frac{\pi(a_i|s)}{\nu(a_i|s)}\right)$$

Soft Greedy Policy Improvement

$$\pi'(\cdot|s) = \operatorname*{argmax}_{\pi'} D_{KL}\left(\pi'(\cdot|s)\middle|\middle|\nu(\cdot|s)\right), \quad \nu(\cdot|s) = \mathrm{Softmax}\left(\frac{1}{\alpha} q_\pi^\alpha(s,\cdot)\right)$$

(Soft Greedy Policy Improvement) $\rightarrow$ (Greedy Policy Improvement) as $\alpha \rightarrow 0$.

Soft Policy Improvement Theorem

Let π be a policy. If π' is defined by Soft Greedy Policy Improvement, then $q_{\pi'}^\alpha(s,a) \geq q_\pi^\alpha(s,a)$

Soft Policy Iteration

Let π_0 and $L, K \in \mathbb{N}$.
For each $k \in \overline{0, K}$, do

- (Soft Policy Evaluation)

$$q_{l+1}(s,a) = \mathcal{R}(s,a) + \gamma \sum_{i=1}^{n} \mathcal{P}(s_i|s,a) V_l(s_i),$$

$$V_l(s_i) = \sum_{j=1}^{m} \pi(a_j|s_i)\Big(q_l(s_i,a_j) - \alpha \log \pi(a_j|s_i)\Big)$$

- (Soft Policy improvement)

$$\pi_{k+1}(\cdot|s) = \operatorname*{argmax}_{\pi'} D_{KL}\Big(\pi'(\cdot,s)\Big|\Big|\mathrm{Softmax}\big(q_L(s,a)/\alpha\big)\Big)$$

Soft Policy Iteration

Let π_0 and $L, K \in \mathbb{N}$.
For each $k \in \overline{0, K}$, do

- (Soft Policy Evaluation)

$$q_{l+1}(s,a) = \mathcal{R}(s,a) + \gamma \sum_{i=1}^{n} \mathcal{P}(s_i|s,a) V_l(s_i),$$

$$V_l(s_i) = \sum_{j=1}^{m} \pi(a_j|s_i)\Big(q_l(s_i,a_j) - \alpha \log \pi(a_j|s_i)\Big)$$

- (Soft Policy improvement)

$$\pi_{k+1}(\cdot|s) = \underset{\pi'}{\operatorname{argmax}}\, D_{KL}\Big(\pi'(\cdot,s)\Big|\Big|\mathrm{Softmax}\big(q_L(s,a)/\alpha\big)\Big)$$

Theorem

$\pi_k \to \pi_*^\alpha$ as $k \to \infty$

Markov Decision Process

Markov Property

$$\mathbb{P}[S_{t+1}|S_t, A_t] = \mathbb{P}[S_{t+1}|S_1, A_1, S_2, A_2 \ldots, S_t, A_t]$$

$$\mathbb{P}[R_t|S_t, A_t] = \mathbb{P}[R_t|S_1, A_1, S_2, A_2 \ldots, S_t, A_t] = 1$$

Markov Decision Process $\langle \mathcal{S}, \mathcal{S}_F, \mathcal{A}, \mathcal{P}, \mathcal{P}_0, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is an infinite state space
- $\mathcal{S}_F$ is a set of final states
- $\mathcal{A}$ is an infinite (finite) action space
- $\mathcal{P}$ is an unknown transition probability function

$$\mathcal{P}(s'|s, a) = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]$$

- $\mathcal{P}_0$ is an unknown initial state probability function
- $\mathcal{R}$ is an unknown reward function

$$\mathcal{R}(s, a) = R_t \quad \Leftrightarrow \quad \mathbb{P}[R_t|S_t = s, A_t = a] = 1$$

- $\gamma \in [0, 1]$ is a discount coefficient

Entropy and KL-Divergence

Entropy

$$\mathcal{H}(\pi(\cdot|s)) = -\int \pi(a|s) \log \pi(a|s) da \approx \frac{1}{k} \sum_{i=1}^{k} \log \pi(a_i|s), \quad a_i \sim \pi(\cdot|s)$$

Entropy and KL-Divergence

Entropy

$$\mathcal{H}(\pi(\cdot|s)) = -\int \pi(a|s)\log \pi(a|s)da \approx \frac{1}{k}\sum_{i=1}^{k}\log \pi(a_i|s), \quad a_i \sim \pi(\cdot|s)$$

KL-Divergence

$$D_{KL}\big(\pi(\cdot|s)\big|\big|\nu(\cdot|s)\big) = \int \pi(a|s)\log\left(\frac{\pi(a|s)}{\nu(a|s)}\right)da$$

$$\approx \frac{1}{k}\sum_{i=1}^{k}\log\left(\frac{\pi(a_i|s)}{\nu(a_i|s)}\right), \quad a_i \sim \pi(\cdot|s)$$

SAC: Main Points

Policy and Value Function Approximations

$$\pi^\eta(a|s) \approx \pi_*^\alpha(a|s), \quad Q^\theta(s,a) \approx q_{\pi^\eta}^\alpha(s,a)$$

SAC: Main Points

Policy and Value Function Approximations

$$\pi^\eta(a|s) \approx \pi_*^\alpha(a|s), \quad Q^\theta(s,a) \approx q_{\pi^\eta}^\alpha(s,a)$$

Bellman equation for q_π^s

$$q_\pi^\alpha(s,a) = \mathbb{E}_\pi\Big[R_t + \gamma\big(q_\pi^\alpha(S_{t+1}, A_{t+1}) - \alpha \log \pi(A_{t+1}|S_{t+1})\big)\Big]$$

where $R_t = \mathcal{R}(s,a)$, $S_{t+1} \sim \mathcal{P}(\cdot|s,a)$, and $A_{t+1} \sim \pi(\cdot|S_{t+1})$

SAC: Main Points

Policy and Value Function Approximations

$$\pi^\eta(a|s) \approx \pi_*^\alpha(a|s), \quad Q^\theta(s,a) \approx q_{\pi^\eta}^\alpha(s,a)$$

Bellman equation for q_π^s

$$q_\pi^\alpha(s,a) = \mathbb{E}_\pi\Big[R_t + \gamma\big(q_\pi^\alpha(S_{t+1}, A_{t+1}) - \alpha \log \pi(A_{t+1}|S_{t+1})\big)\Big]$$

where $R_t = \mathcal{R}(s,a)$, $S_{t+1} \sim \mathcal{P}(\cdot|s,a)$, and $A_{t+1} \sim \pi(\cdot|S_{t+1})$

For Each Step

If

$$Q^\theta(S_t, A_t) \approx R_t + \gamma\Big(Q^\theta(S_{t+1}, A_{t+1}) - \alpha \log \pi^\eta(A_{t+1}|S_{t+1})\Big),$$

where $R_t = \mathcal{R}(S_t, A_t)$, $S_{t+1} \sim \mathcal{P}(\cdot|S_t, A_t)$, and $A_{t+1} \sim \pi^\eta(\cdot|S_{t+1})$, then

$$Q^\theta \approx q_{\pi^\eta}^\alpha$$

SAC: Main Points

Policy and Value Function Approximations

$$\pi^\eta(a|s) \approx \pi_*^\alpha(a|s), \quad Q^\theta(s,a) \approx q_{\pi^\eta}^\alpha(s,a)$$

Policy Improvement

$$\pi'(s,a) = \underset{\pi}{\operatorname{argmax}}\, D_{KL}\Big(\pi(\cdot,s)\Big|\Big|\mathrm{Softmax}\big(q_\pi^\alpha(s,\cdot)/\alpha\big)\Big)$$

SAC: Main Points

Policy and Value Function Approximations

$$\pi^\eta(a|s) \approx \pi_*^\alpha(a|s), \quad Q^\theta(s,a) \approx q_{\pi^\eta}^\alpha(s,a)$$

Policy Improvement

$$\pi'(s,a) = \operatorname*{argmax}_\pi D_{KL}\Big(\pi(\cdot,s)\Big|\Big|\text{Softmax}\big(q_\pi^\alpha(s,\cdot)/\alpha\big)\Big)$$

For Each Step

If $Q^\theta \approx q_{\pi^\eta}^\alpha$, then

$$Q^\theta(S_t, a^\eta) - \alpha \log \pi^\eta(a^\eta|S_t) \to \max_\eta$$

where $a^\eta \sim \pi^\eta(\cdot|S_t)$

Reparameterization Trick

$$L(\eta) = F(a), \quad a \sim \pi^{\eta}(\cdot|s)$$
$$\Downarrow$$
$$\nabla_{\eta} L(\eta) = ?$$

Reparameterization Trick

$$L(\eta) = F(a), \quad a \sim \pi^\eta(\cdot|s)$$
$$\Downarrow$$
$$\nabla_\eta L(\eta) = ?$$

Case of Normal Distribution

If $\pi^\eta(a|s) = \mathcal{N}(a|\mu^\eta(s), (\sigma^\eta(s))^2)$, then

$$a \sim \pi^\eta(\cdot|s) \quad \Leftrightarrow \quad a = \mu^\eta(s) + \sigma^\eta(s)\epsilon, \quad \epsilon \sim \mathcal{N}(\cdot|0, 1)$$

Reparameterization Trick

$$L(\eta) = F(a), \quad a \sim \pi^\eta(\cdot|s)$$
$$\Downarrow$$
$$\nabla_\eta L(\eta) = ?$$

Case of Normal Distribution

If $\pi^\eta(a|s) = \mathcal{N}(a|\mu^\eta(s), (\sigma^\eta(s))^2)$, then

$$a \sim \pi^\eta(\cdot|s) \quad \Leftrightarrow \quad a = \mu^\eta(s) + \sigma^\eta(s)\epsilon, \quad \epsilon \sim \mathcal{N}(\cdot|0,1)$$

$$\Downarrow$$

$$L(\eta) = F(\mu^\eta(s) + \sigma^\eta(s)\epsilon), \quad \epsilon \sim \mathcal{N}(\cdot|0,1)$$

Reparameterization Trick

$$L(\eta) = F(a), \quad a \sim \pi^\eta(\cdot|s)$$
$$\Downarrow$$
$$\nabla_\eta L(\eta) = ?$$

Case of Normal Distribution

If $\pi^\eta(a|s) = \mathcal{N}(a|\mu^\eta(s), (\sigma^\eta(s))^2)$, then

$$a \sim \pi^\eta(\cdot|s) \quad \Leftrightarrow \quad a = \mu^\eta(s) + \sigma^\eta(s)\epsilon, \quad \epsilon \sim \mathcal{N}(\cdot|0,1)$$

$$\Downarrow$$

$$L(\eta) = F(\mu^\eta(s) + \sigma^\eta(s)\epsilon), \quad \epsilon \sim \mathcal{N}(\cdot|0,1)$$

$$\Downarrow$$

$$\nabla_\eta L(\eta) = \nabla F\big(\mu^\eta(s) + \sigma^\eta(s)\epsilon\big)\Big(\nabla_\eta \mu^\eta(s) + \nabla_\eta \sigma^\eta(s)\epsilon\Big), \quad \epsilon \sim \mathcal{N}(\cdot|0,1)$$

Soft Actor-Critic (SAC)

Initialize policy network π^η and q-networks Q^{θ_i} and $Q^{\theta_i'}$ ($\theta_i' = \theta_i$), $i = 1, 2$.
During each episode, do

- Being in state S_t, act $A_t \sim \pi^\eta(\cdot|S_t)$, get reward R_t, done signal D_t, and next state S_{t+1}. Store $(S_t, A_t, R_t, D_t, S_{t+1}) \to M$
- Get a batch $\{(s_j, a_j, r_j, d_j, s_j')\}_{j=1}^n \leftarrow M$, sample $a_j' \sim \pi^\eta(\cdot|s_j')$, determine target values

$$y_j = r_j + \gamma(1 - d_j)\left(\min_{i=1,2} Q^{\theta_i'}(s_j', a_j') - \alpha \log \pi(a_j'|s_j')\right)$$

sample $a_j^\eta \sim \pi^\eta(\cdot|s_j)$, and determine the loss functions

$$L_i(\theta_i) = \frac{1}{n}\sum_{j=1}^n \left(y_j - Q^{\theta_i}(s_j, a_j)\right)^2, \quad i = 1, 2,$$

$$L_3(\eta) = \frac{1}{n}\sum_{j=1}^n \left(\min_{i=1,2} Q^{\theta_i}(s_j, a_j^\eta) - \alpha \log \pi^\eta(a_j^\eta|s_j)\right)$$

and, for $i = 1, 2$, update the parameters

$$\theta_i \leftarrow \theta_i - \beta_i \nabla_{\theta_i} L_i(\theta_i), \quad \eta \leftarrow \eta + \beta \nabla_\eta L_3(\eta), \quad \theta_i' \leftarrow \tau\theta_i' + (1-\tau)\theta_i$$

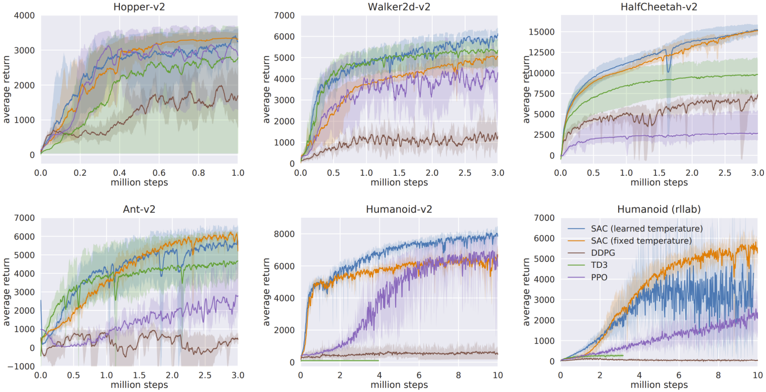
SAC Improvement

- Using $\pi^\eta(a|s) = \tanh\Big(\mathcal{N}\big(a \,\big|\, \mu^\eta(s), (\sigma^\eta(s))^2\big)\Big)$

SAC Improvement

- Using $\pi^\eta(a|s) = \tanh\Big(\mathcal{N}\big(a \,\big|\, \mu^\eta(s), (\sigma^\eta(s))^2\big)\Big)$
- It is possible to learn α (Haarnoja T. et al. Soft Actor-Critic Algorithms and Applications)

SAC Results



Haarnoja T. et al. Soft Actor-Critic Algorithms and Applications

QUESTIONS?