DDPG

: Deep Deterministic Policy Gradient

24.02.29
정상혁

Deterministic Policy Gradient (DPG)

Stochastic policy gradient

- Learns a **stochastic** policy

  $$\pi_\theta(a\,|\,s)$$

- (**Stochastic**) Policy gradient theorem

  Objective function $J(\theta) = E_{s\sim\rho_\pi,\, a\sim\pi_\theta}[Q^\pi(s,a)]$

  $$\nabla_\theta J(\theta) = E_{s\sim\rho_\pi,\, a\sim\pi_\theta}[Q^\pi(s,a)\nabla_\theta log\pi_\theta(a\,|\,s)]$$

- Weight update

  $$\theta := \theta + \alpha\nabla_\theta J(\theta)$$

Deterministic policy gradient

- Learns a **deterministic** policy

  $$a = \mu(s)$$

- **Deterministic** policy gradient theorem

  Objective function $J(\theta) = E_{s\sim\rho_\mu}[Q^\mu(s,a)]$

  $$\nabla_\theta J(\theta) = E_{s\sim\rho_\mu}[\nabla_a Q^\mu(s,a)\,|_{a=\mu_\theta(s)}\nabla_\theta\mu_\theta(s)]$$

- Weight update

  $$\theta := \theta + \alpha\nabla_\theta J(\theta)$$

- DPG requires less samples to approximate the gradient than stochastic PG

Deterministic Policy Gradient Theorem

(Stochastic) Policy Gradient Theorem

Objective function

$$J(\theta) = E_{s\sim\rho_\pi,\, a\sim\pi_\theta}[Q^\pi(s,a)] = \int_S \rho_\pi(s) \int_A \pi_\theta(a\,|\,s)\, Q^\pi(s,a)\ da\, ds$$

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int_S \rho_\pi(s) \int_A \nabla_\theta \pi_\theta(a\,|\,s)\, Q^\pi(s,a)\ da\, ds \\
&= \int_S \rho_\pi(s) \int_A \pi_\theta(a\,|\,s)\, Q^\pi(s,a) \frac{\nabla_\theta \pi_\theta(a\,|\,s)}{\pi_\theta(a\,|\,s)}\ da\, ds \\
&= \int_S \rho_\pi(s) \int_A \pi_\theta(a\,|\,s)\, Q^\pi(s,a)\, \nabla_\theta log\, \pi_\theta(a\,|\,s)\ da\, ds \\
&= E_{s\sim\rho_\pi,\, a\sim\pi_\theta}[Q^\pi(s,a)\, \nabla_\theta\, log \pi_\theta(a\,|\,s)]
\end{aligned}$$

Deterministic Policy Gradient Theorem

Objective function

$$J(\theta) = E_{s\sim\rho_\mu}[Q^\mu(s,a)] = \int_S \rho_\mu(s) Q^\mu(s,a)\, ds \quad \text{where } a = \mu_\theta(s)$$

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int_S \rho_\mu(s)\, \nabla_\theta Q^\mu(s,a)\ ds \\
&= \int_S \rho_\mu(s)\, \nabla_a Q^\mu(s,a)\,|_{a=\mu_\theta(s)}\, \nabla_\theta \mu_\theta(s)\ ds \\
&= E_{s\sim\rho_\pi,\, a\sim\pi_\theta}[\nabla_a Q^\mu(s,a)\,|_{a=\mu_\theta(s)}\, \nabla_\theta \mu_\theta(s)]
\end{aligned}$$

- **State visitation frequency :** discounted sum of probabilities of visiting a given state s under policy π

$$\rho_\pi(s) = \Sigma_{t=0}^{\infty}\, \gamma^t P(s_t = s\,|\,\pi) = \int_S \Sigma_{t=0}^{\infty}\, \gamma^t p_0(s') p(s' \to s\,|\,t, \pi)\, ds'$$

$p_0(s')$: initial probability

$p(s' \to s\,|\,t, \pi)$: visitation probability from s' to s in t steps following π

$\rightarrow \quad \Sigma_{s\in S}\ \rho_\pi(s) = \Sigma_{t=0}^{\infty}\, \gamma^t\ \Sigma_{s\in S} P(s_t = s\,|\,\pi) = \Sigma_{t=0}^{\infty}\, \gamma^t = \dfrac{1}{1-\gamma}$

$\rightarrow$ Therefore, $(1-\gamma)\rho_\pi(s)$ considered as a probability distribution over the state space

Deep Deterministic Policy Gradient (DDPG)

- **DDPG**
  - Actor-Critic algorithm
  - DPG (deterministic policy) + DQN (experience replay / target network)

Actor

Objective function (maximize)

$$J(\theta) = E_{s \sim \rho_\mu}[Q_\phi(s,a)] \quad \text{where } a = \mu_\theta(s)$$

$$\nabla_\theta J(\theta) = E_{s \sim \rho_\mu}[\nabla_a Q_\phi(s,a)|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s)]$$

Critic

Loss function (minimize)

$$L(\phi) = E_{s \sim \rho_\mu}[(r + \gamma \hat{Q}_{\hat{\phi}}(s', \hat{\mu}_{\hat{\theta}}(s')) - Q_\phi(s,a))^2] \quad \text{where } a = \mu_\theta(s)$$

$$\nabla_\phi L(\phi) = -E_{s \sim \rho_\mu}[(r + \gamma \hat{Q}_{\hat{\phi}}(s', \hat{\mu}_{\hat{\theta}}(s')) - Q_\phi(s,a)) \nabla_\phi Q_\phi(s,a)]$$

- For action exploration, adding a noise : $a_t = \mu_\theta(s_t) + N_t$
- Soft update : $\hat{\phi} \leftarrow \tau \phi + (1-\tau)\hat{\phi}$, $\hat{\theta} \leftarrow \tau \theta + (1-\tau)\hat{\theta}$
- Difficulty : performance does not improve monotonically

DDPG pseudo code

Initialize critic network $Q(s, a; \phi)$ and actor network $\mu(s; \theta)$ randomly
Initialize target networks $\hat{Q}, \hat{\mu}$ with weights $\hat{\phi} = \phi$, $\hat{\theta} = \theta$
Initialize replay buffer $\mathcal{R}$
for episode $= 1, M$ **do**
  Initialize a random noise process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for $t = 1, T$ **do**
    Select action $a_t = \mu(s_t; \theta) + \mathcal{N}_t$ exploration
    Execute a_t and observe r_{t+1}, s_{t+1}
    Store transition $(s_t, a_t, r_{t+1}, s_{t+1})$ in $\mathcal{R}$ replay buffer
    Sample minibatch of B transitions $(s_i, a_i, r_{i+1}, s_{i+1})$ from $\mathcal{R}$
    Set $y_i = r_{i+1} + \gamma \hat{Q}(s_{i+1}, \hat{\mu}(s_{i+1}; \hat{\theta}); \hat{\phi})$
    Update critic network by minimizing the loss $L = \frac{1}{B} \sum_i (y_i - Q(s_i, a_i; \phi))^2$
    Update actor network using the deterministic policy gradient:
      $\nabla_\theta J \approx \frac{1}{B} \sum_i \nabla_a Q(s_i, a; \phi)\Big|_{a=\mu(s_i;\theta)} \nabla_\theta \mu(s_i; \theta)$
    Update target networks: $\hat{\phi} \leftarrow \tau\phi + (1-\tau)\hat{\phi}$, $\hat{\theta} \leftarrow \tau\theta + (1-\tau)\hat{\theta}$ soft update
  end
end