Deep Deterministic Policy Gradient Discussion

2019.03.15

王政飞

目录

- Deterministic Policy Gradient. ICML 2014. DeepMind.
- Deep Deterministic Policy Gradient. ICLR 2016. DeepMind.
- Applications (*fake*).

Deterministic Policy Gradient

- Stochastic policy gradient （SPG）用一个参数化的概率分布$\pi_\theta(a|s) = \mathbb{P}[a|s;\theta]$来表示策略
- Deterministic policy gradient[1]（DPG）希望用确定性策略$a = \mu_\theta(s)$来表示
- 针对同样的状态s：
  - 参数确定的SPG会采取不同的动作;
  - 参数确定的DPG会产生同样的动作。
  - 论文中提到SPG的训练会需要更多的sample，WHY?
- 为了保持对所有状态和动作的探索，随机策略仍旧是十分必要的
- 采用了actor-critic算法
- 论文中提到DPG的应用场景比SPG更广泛，WHY? （Introduction最后一段）

[1] Deterministic policy gradient algorithms. ICML 2014. DeepMind.

Gradients of Deterministic Policies

- 使用$Q^{\mu}(s,a)$表示动作价值函数，θ为策略函数μ的参数
- 一种贪心的参数更新思路：在每一个状态s，都尽可能的优化μ使得当前状态的Q尽可能大

$$\theta^{k+1} = \theta^k + \alpha \mathbb{E}_{s\sim\rho^{\mu^k}}\left[\nabla_\theta Q^{\mu^k}(s,\mu_\theta(s))\right]$$

- 应用链式法则：

$$\theta^{k+1} = \theta^k + \alpha \mathbb{E}_{s\sim\rho^{\mu^k}}\left[\nabla_\theta \mu_\theta(s)\ \nabla_a Q^{\mu^k}(s,a)\Big|_{a=\mu_\theta(s)}\right]$$

Deep Deterministic Policy Gradient

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

[2] Continuous control with deep reinforcement learning. ICLR 2016. DeepMind.

Get your Hands Dirty

1. Replay Buffer (for all off-policy RL algorithms)
   a) st, at, rt, st+1 – algorithms needed
   b) done – for calculate Q
2. Actor-Critic: share architecture (initialization)
3. DDPG agent
   a) select actions: random / greedy
   b) remember trajectories
   c) update policy
4. interaction with environment
5. if needed, normalize your environment

Watching Code Time

- Spinning Up: https://github.com/openai/spinningup/blob/master/spinup/algos/ddpg/ddpg.py

Applications

- NIPS/NeurIPS challenges, Learning to Run/AI for Prosthetics
- What else?