Deep Deterministic Policy Gradient

Suppose we have a perfect Q-value estimator $Q_\theta(s,a)$, then can we use it to learn a policy with off-policy alogorithm. This is because for an optimal policy $\eta : S \to A$, we have

$$Q(s,\eta(s)) \geq Q(s,a) \forall a \in A$$

Hence, we can update the policy parameter by doing gradient ascend on

$$\mathbb{E}[Q(s,\eta(s))]$$

through a variable.

The idea of DDPG is to learn a Q-function via Q-learning styled algorithm and learn a deterministic policy η by maximizing $Q(s,\eta(s))$.

Why η needs to be a deterministic policy?

Algorithm 1: Deep Deterministic Policy Gradient

Input: initial policy parameter θ, Q-function parameters ϕ, emtpy replay buffer D

Set target parameters equal to main parameters $\theta_{targ} \leftarrow \theta$, $\phi_{targ} \leftarrow \phi$

repeat

Observe state s and select an action (with Gaussian noise) $a = clip(\eta_\theta(s) + \epsilon, a_{low}, a_{high}), \epsilon \sim N$

Execute a in the environment

Observe next state s', reward r and done signal d signal d to indicate whether s' is terminal.

Store (s,a,r,s',d) in the replay buffer D

If s' is terminal, reset environment state

if it's time to update **then**

for however many updates **do**

Randomly sample a batch of transitions, $B = \{(s,a,r,s',d)\}$ from D

Compute targets

$$y(r,s',d) = r + \gamma(1-d)Q_{\phi_{targ}}(s', \mu_{\phi_{targ}}(s'))$$

Update policy by one sttep of gradient ascent using

$$\nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s',d)\in B} (Q_\phi(s,a) - y(r,s',d))^2$$

Update policy by one step of gradient using

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\theta(s, \mu_\theta(s))$$

Update target network with

$$\phi_{targ} \leftarrow \rho\phi_{targ} + (1-\rho)\phi \quad (1)$$

$$\theta_{targ} \leftarrow \rho\theta_{targ} + (1-\rho)\theta \quad (2)$$

end for
end if
until convergence