Deep Deterministic Policy Gradient

$V_{k+1}^{*}(s) = \max_{a} \sum_{s'} P(s'|s,a)[r + \gamma V_k^{*}(s')]$

$Q_{i+1}(s) \leftarrow \sum_{s'} P(s'|s,a,) \left[r + \gamma \cdot \max_{a'} Q_i(s',a')\right]$

Policy Evaluation

$V_{i+1}^{\pi_k}(s) = \sum_{s'} P(s'|s,\pi_k(s))[r + \gamma V_i^{\pi_k}(s')]$

Policy Improvement

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s'|s,a)[r + \gamma V^{\pi_k}(s')]$

Q – Learning:

$Q_{i+1}(s,a) \leftarrow Q_i(s,a) + \alpha\left(r + \gamma \cdot \max_{a'} Q_i(s',a') - Q_i(s,a)\right)$

DQN Update:

$\theta_{t+1} = \theta_t + \alpha(y_t - Q(s_t,a_t;\theta))\nabla_{\theta_t} Q(s_t,a_t;\theta_t)$

DQN Q Update:

$Q(s_t,a_t;\theta_{t+1}) \approx Q(s_t,a_t;\theta) + \alpha(y_t - Q(s_t,a_t;\theta_t)) \left|\left|\nabla_{\theta_t} Q(s_t,a_t;\theta_t)\right|\right|_2^2$

n – step return:

$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n \hat{v}(s_{t+n})$

- Stochastic target v/s Deterministic target:
  - $Q^{\pi}(s_t,a_t) = \mathbb{E}_{r_t,s_{t+1}}\left[r(s_t,a_t) + \gamma \mathbb{E}_{a_{t+1}\sim\pi}[Q^{\pi}(s_{t+1},a_{t+1})]\right]$
  - $Q^{\pi'}(s_t,a_t) = \mathbb{E}_{r_t,s_{t+1}}\left[r(s_t,a_t) + \gamma Q^{\pi'}(s_{t+1},\pi'(s_{t+1}))\right]$
- Critic loss function:
  - $L(w) = \mathbb{E}_{s_t\sim\rho^{\beta},a_t\sim\beta,r_t\sim E}[(Q(s_t,a_t|w) - y_t)^2]$
- Actor Performance function:
  - $\nabla_{\theta'} J \approx \mathbb{E}[\nabla_a Q(s_t,\pi'(s_t)|w)\nabla_{\theta'}(\pi'(s_t|\theta'))]$
- Exploration in continuous action space
  - $\mathcal{N}_t$ is a temporally correlated noise for exploration: Obtained using Ornstein – Uhlenbeck process.
  - $\mu(s_t) = \pi'(s_t|\theta_t) + \mathcal{N}_t$

- State visitation distribution for a policy π is ρ^{π}
- Behaviour policy is $\beta \leftarrow$ off policy.
- Replay Buffer $\mathcal{D}$
- Separate Target Network
  - $y_t = r(s_t,a_t) + \gamma Q(s_{t+1},\pi'(s_{t+1})|w)$

Algorithm 1: DDPG Algorithm

Initialize critic network $Q(s,a|w)$ and actor $\pi'(s|\theta)$ with weights w and θ
Initialize target network Q' and μ with weights: $w' \leftarrow w$, $\theta' \leftarrow \theta$
Initialize replay buffer $\mathcal{D}$
For episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial state s_1
  For t = 1, T **do**
    Select action $a_t = \pi'(s_t|\theta) + \mathcal{N}_t$ according to the current policy & noise
    Execute action a_t and observe reward r_t and next state s_{t+1}
    Store transition (s_t,a_t,r_t,s_{t+1}) in $\mathcal{D}$
    Sample random minibatch of transitions (s_i,a_i,r_i,s_{i+1}) from $\mathcal{D}$
    Set $y_i = r_i + \gamma Q'(s_{i+1},\pi'(s_{i+1}|\theta');w')$
    Update critic by minimizing loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i,a_i|w))^2$
    Update the actor policy: $\nabla_{\theta} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|w)\big|_{s=s_i,a=\pi'(s_i)} \nabla_{\theta}\pi'(s|\theta)\big|_{s_i}$
    Update the target networks:
    $$w' \leftarrow \tau w + (1-\tau)w';\ \theta' \leftarrow \tau\theta + (1-\tau)\theta'$$
  end For
end For

https://www.pavankumar.co.in/

Lillicrap, Timothy P. et al. "Continuous control with deep reinforcement learning." arXiv preprint arXiv: 1509.02971 (2015)