

Deep Deterministic Policy Gradients - Components and Extensions

Yannik Frisch

Tabea Wilke

Maximilian Gehrke

Group 19 Oleg Arenz



TECHNISCHE
UNIVERSITÄT
DARMSTADT



▶ Actor-Critic Method

- ▶ Approximated critic $Q(s, a|\theta^Q)$
- ▶ Approximated actor $\pi(s|\theta^\pi)$

▶ Deep Q-Learning

- ▶ Q-Learning with approximated critic by nn
- ▶ $\nabla_{\theta^Q} L(\theta^Q) = \mathbf{E} \left[(r + \gamma \max_{a'} Q(s', a'|\theta^Q) - Q(s, a|\theta^Q)) \nabla_{\theta^Q} Q(s, a|\theta^Q) \right]$
- ▶ Experience Replay Buffer
- ▶ Target Network(s)

▶ Deterministic Policy Gradient

- ▶ $\nabla_{\theta^\pi} J(\theta^\pi) \approx \mathbf{E} \left[\nabla_a Q(s, a)|_{a=\pi(s|\theta^\pi)} \nabla_{\theta^\pi} \pi(s|\theta^\pi) \right]$
- ▶ Enables to learn a deterministic policy while following a stochastic exploratory policy

DDPG Pseudocode



Algorithm 2 Deep Deterministic Policy Gradient (DDPG)

Initialize: Replay buffer D with high capacity

Initialize: Critic network $Q(s, a|\theta^Q)$ and actor network $\pi(s|\theta^\pi)$ with random weights θ^Q and θ^π

Initialize: Initialize target networks Q' and π' with weights $\theta^{Q'} \leftarrow \theta^Q$ and $\theta^{\pi'} \leftarrow \theta^\pi$
for episode 1 **to** M **do**

 Initialize random process N for action exploration

 Reset environment to state s_1

for $t = 1$ **to** T **do**

 Select action $a_t = \pi(s_t|\theta^\pi) + N_t$ from local actor

 Execute action a_t and observe reward r_t and next state s_{t+1}

 Save (s_t, a_t, r_t, s_{t+1}) in replay buffer D

 Sample mini-batch $(s_i, a_i, r_i, s_{i+1})_k$ with size k from D

 Set TD-target from target networks:

$$y_i = r_i + \gamma Q'(s_{i+1}, \pi'(s_{i+1}|\theta^{\pi'}))|\theta^{Q'}$$

 Update the critic by minimizing the loss:

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

 Update the actor using the sampled policy gradient:

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

 Update the target networks:

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

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

end for

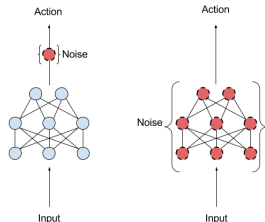
end for

Improvements for DDPG

► D4PG

- Importance Weighted Experience Sampling
- Utilizing N-Step return
- Parallelized Actors

► Parameter Noise for Exploration



► Deep Changes



- ▶ For publication references please see our paper "Deep Deterministic Policy Gradients: Components and Extensions"
- ▶ GoogLeNet: <https://towardsdatascience.com/an-intuitive-guide-to-deep-network-architectures-65fdc477db41>
- ▶ Parameter noise for exploration:
<https://openai.com/blog/better-exploration-with-parameter-noise/>