Environment description

Actor-critic models

TD3 algorithm

Results

Conclusions

Application of TD3 algorithm to the BipedalWalker-v3 environment

Luka Utješinović, Luka Boljević

May 25, 2022

Environment description

BipedalWalker-v3 - environment provided by the famous Gym OpenAI Python library.

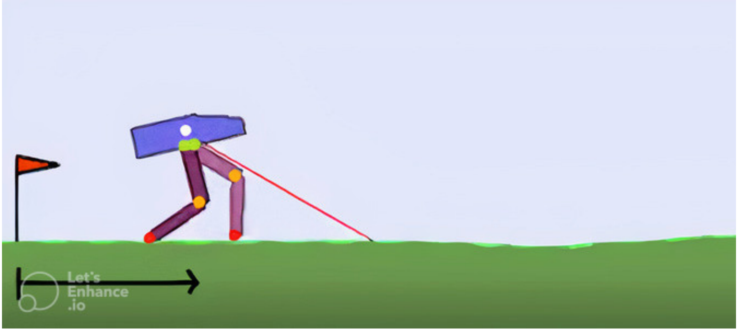


Figure: A single frame from the environment

Drawbacks of Q-learning

The first algorithm that comes to mind is Q-learning.
Q-learning update rule:

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\bigg(\underbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}_{\text{new value (temporal difference target)}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \bigg)}^{\text{temporal difference}}$$

Infeasible for continuous action space environments because of the $\max_a Q(s_{t+1}, a)$ term.

Alternative: actor-critic paradigm

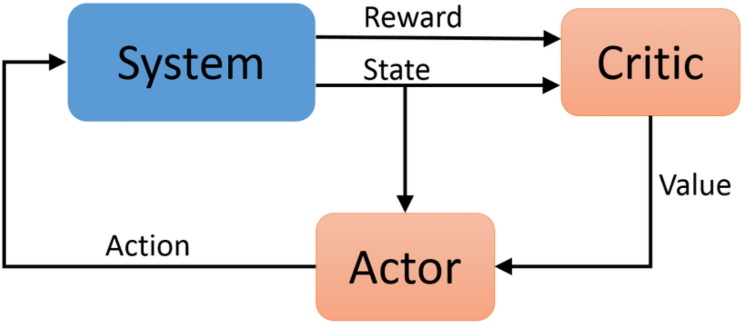
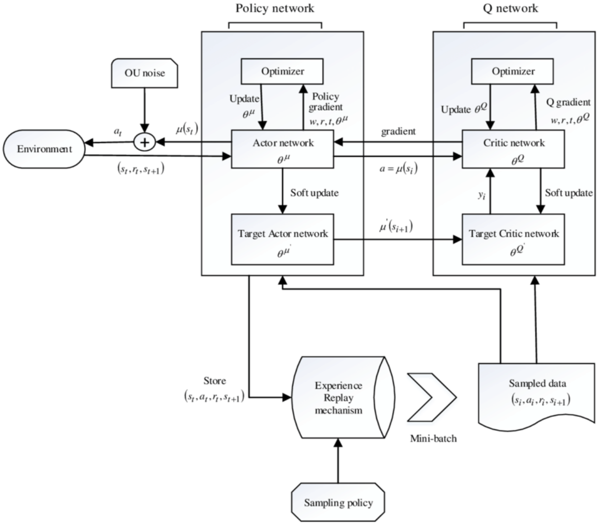


Figure: High level overview of the actor-critic paradigm

DDPG - most compact explanation possible

DDPG - Deep deterministic policy gradient

$y_t \leftarrow r_t + \gamma Q_w(s_{t+1}, \pi_\theta(s_{t+1}))$ (w - "critic", θ - "actor")



TD3

DDPG has exhibited poor performance for this environment

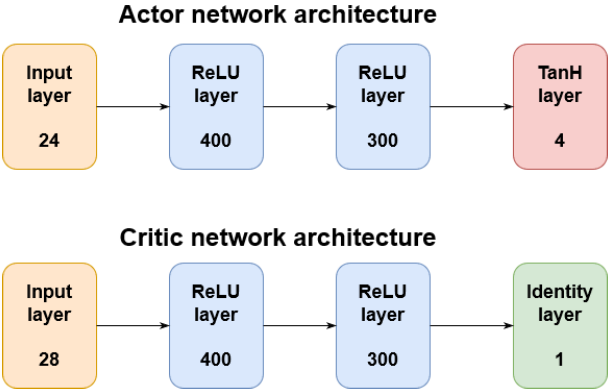
An extended version of DDPG - **Twin delayed DDPG** (TD3)

- Normal distribution instead of complicated OU (Ornstein–Uhlenbeck) random process for exploration
- Two critics Q_{w_1}, Q_{w_2} with target networks $Q_{w_1'}, Q_{w_2'}$
  Modified ground truth:
  $y_t \leftarrow r_t + \gamma \min_{i=1,2} Q_{w_i'}(s_{t+1}, \pi_{\theta'}(s_{t+1}))$

There are a few more slight differences, but they are not crucial for this presentation. Details can be found in the paper.

Results 1

The following architecture for actor and critic networks was used:



tanh - activation of choice for output layer of actor networks, as agent's joint movement $\Leftrightarrow$ motor speed values $\in [-1, 1]$.

Results 2

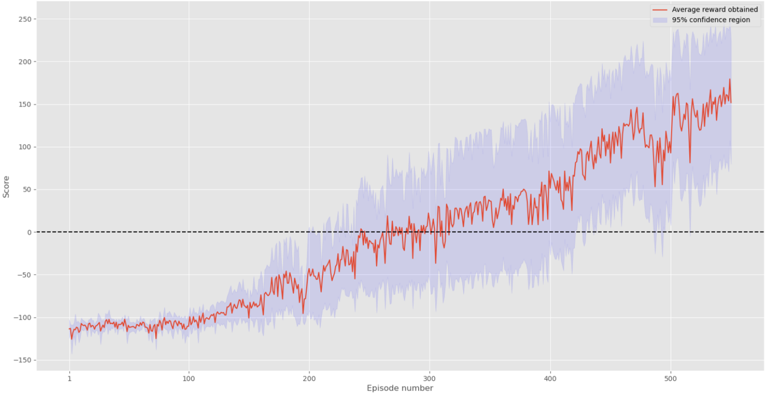


Figure: 15 agents trained for 550 episodes

Conclusions

- TD3 is a very powerful algorithm, but the obtained agent is highly variable.
- A vague idea to improve stability would be to average trained agents