Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient, is a modelfree off-policy actor-critic algorithm, combining DPG with DQN (Deep QNetwork) stabilizes the learning of the Q-function by using experience replay and a frozen target network.

The original DQN works in discrete space, and DDPG extends it to continuous space with the actor-critic framework while learning a deterministic policy.
In DDPG, we use 2 deep neural networks : one is the actor and the other is the critic:

The actor is used to approximate the optimal policy deterministically.

That means we always want to output the best believed action for any given state.

This is unlike a stochastic policy which learn a probability distribution over the actions.
In DDPG, we want the believed best action every single time we query the actor network, that is a deterministic policy. The actor is basically learning *argmaxa* $Q(s; a)$ which is the best action

The critic learns to evaluate the optimal action-value function by using the actor's best believed action.

In DDPG, not only we have a regular network for the actor as well for the critic, we do also have a copy of these regular networks.

We have a target actor and target critic, $\mathbf{Q0(s; a|\theta Q0)}$ and $\mathbf{\mu 0(s|\theta \mu 0)}$ respectively that are used for calculating the target values.

We update those target networks using a soft-update strategy. A soft-update strategy consists of slowly blending your regular network weights with your target network weights

A major challenge of learning in continuous action spaces is exploration. An advantage of off-policies algorithms such as DDPG is that we can treat the problem of exploration independently from the learning algorithm.
As such, an exploration policy $\mu 0$ is constructed by adding noise N

The project done by using policy-based methods to learn the optimal policy The observation space consists of 33 variables the action vector is a number between -1 and 1.

- Policy-based methods have(critic) high variance
- value-based methods(actor) have low variance

combine both actor and critic to get a Actor-critic methods where Actor represented as a neural network that update the policy by choice best action to maxmamize the reward and the critic another network that evaluate the policy being learned that been used to train the actor

Hyperparameters

There were many hyperparameters involved in the experiment.

Value Replay buffer size 1e6

Batch size 800

Actor Learning rate 1e-4

Critic Learning rate 3e-4

Max timesteps per episode 1500

γ (discount factor) 0.99

τ 1e-3

Actor network is FNN with input of 33 and output of 4 , the rest layers (heddin layers are : 512 units , 256 units , 128 units (we use tanh for the output layer to represent values -1,1)

Critic network is FNN with input of 33 and output of 1 , the rest layers (heddin layers are : 516 units , 256 units , 128 units.

idea for improvment

try to use The Q-prop algorithm, which combines both off-policy and on-policy learning.