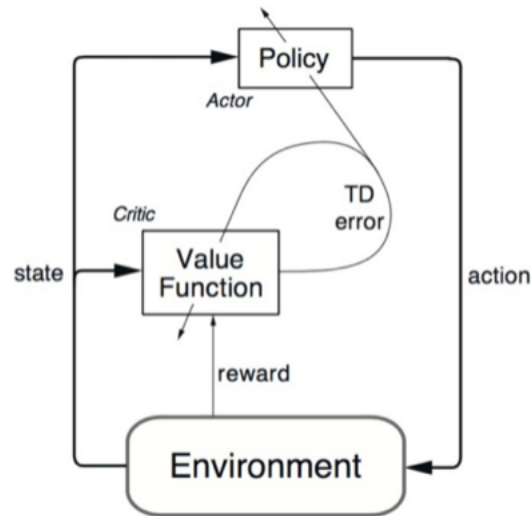


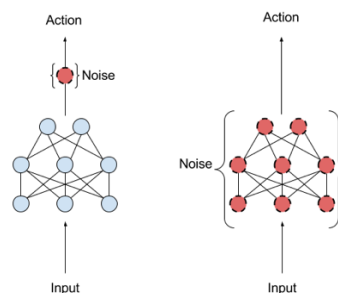
Learning Algorithm

Deep Deterministic Policy Gradient (DDPG)

DDPG algorithm continuously improves the policy while exploring the environment and converges on large action space by using the actor-critic architecture. The actor specifies action in a current state while critic criticizes the actions made by the actor by using Temporal Difference Error.



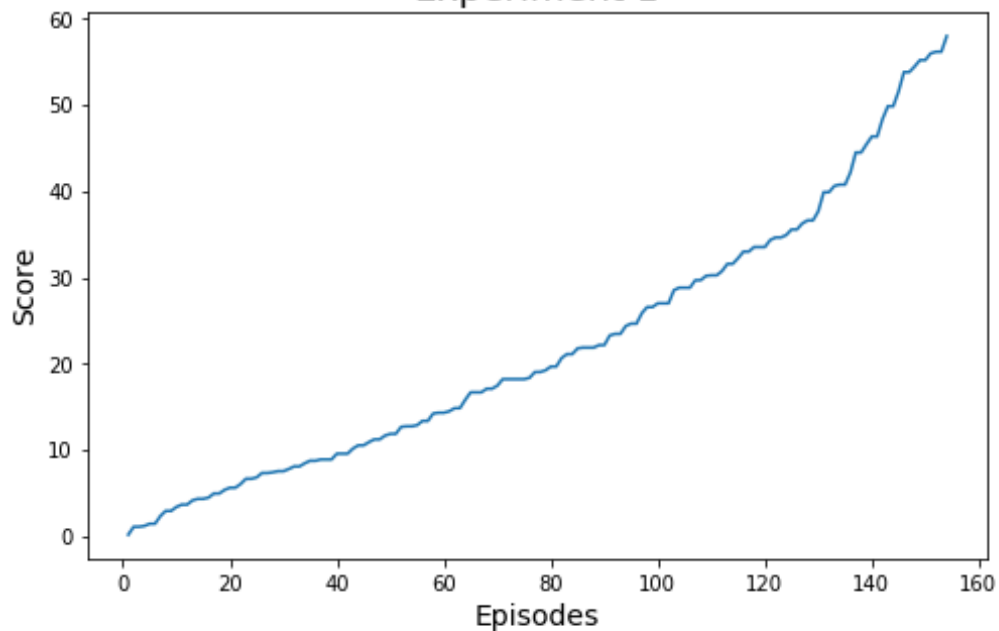
It maximizes the action-value function to compute the direction to change the current action to increase overall discounted reward. However, it does not take into consideration how exploration is done. In the implemented DDPG code an agent adds its experience to the replay buffer and local actor and critic are updated 10 times in a row using different samples from the replay buffer. The OUNoise parameters are also experimented to add noise to the action space of the policy.



Parameters

steps = 10
num_update = 10
SEED = 2
BUFFER_SIZE = int(1e6) - (replay buffer size)
BATCH_SIZE = 32 - (minibatch size)
GAMMA = 0.99 - (discount factor)
TAU = 1e-3 - (for soft update of target parameters)
LR_ACTOR = 1e-4 - (learning rate of the actor)
LR_CRITIC = 3e-4 - (learning rate of the critic)
WEIGHT_DECAY = 0.0001- (L2 weight decay)
OUNoise Parameters: mu =0.,thetha =0.45,sigma=0.50

Experiment 1

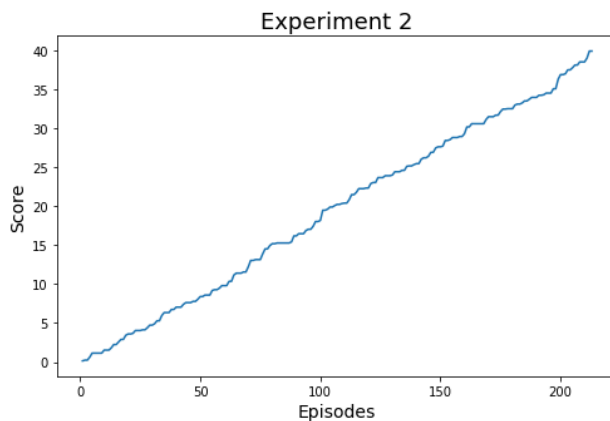


Environment 1 solved in 54 episodes! Average Score: 30.34

Other Experiments to build intuition

Parameters

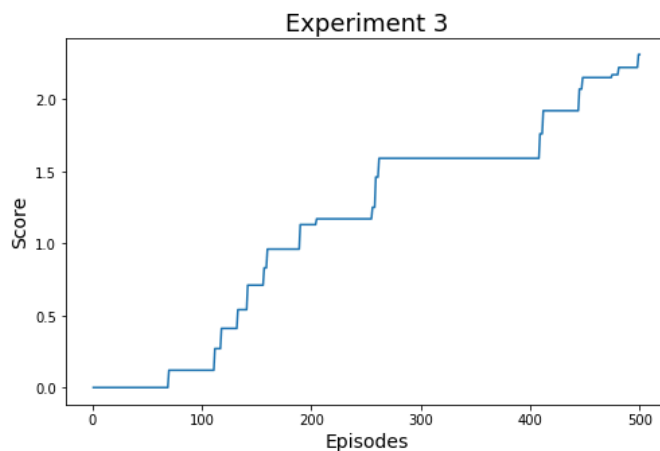
```
steps = 10
num_update = 10
SEED = 2
BUFFER_SIZE = int(1e6) - (replay buffer size)
BATCH_SIZE = 128 - (minibatch size)
GAMMA = 0.99 - (discount factor)
TAU = 1e-3 - (for soft update of target parameters)
LR_ACTOR = 1e-4 - (learning rate of the actor )
LR_CRITIC = 3e-4 - (learning rate of the critic)
WEIGHT_DECAY = 0.0001- (L2 weight decay)
OUNoise Parameters: mu =0.,thetha =0.15,sigma=0.2
```



Environment 2 solved in 113 episodes! Average Score: 30.01

Parameters

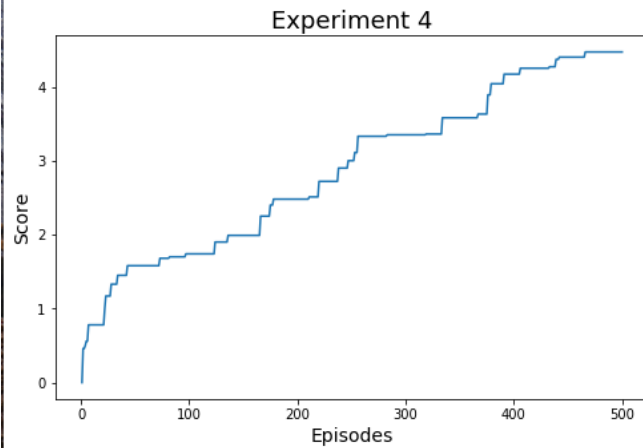
```
steps = 10
num_update = 10
SEED = 2
BUFFER_SIZE = int(1e2) - (replay buffer size)
BATCH_SIZE = 256 - (minibatch size)
GAMMA = 0.50 - (discount factor)
TAU = 1e-2 - (for soft update of target parameters)
LR_ACTOR = 1e-2 - (learning rate of the actor )
LR_CRITIC = 3e-2 - (learning rate of the critic)
WEIGHT_DECAY = 0.0001- (L2 weight decay)
OUNoise Parameters: mu =0.,thetha =0.15,sigma=0.2
```



Environment 3 did not solved in 500 episodes! Average Score: 2.03

Parameters

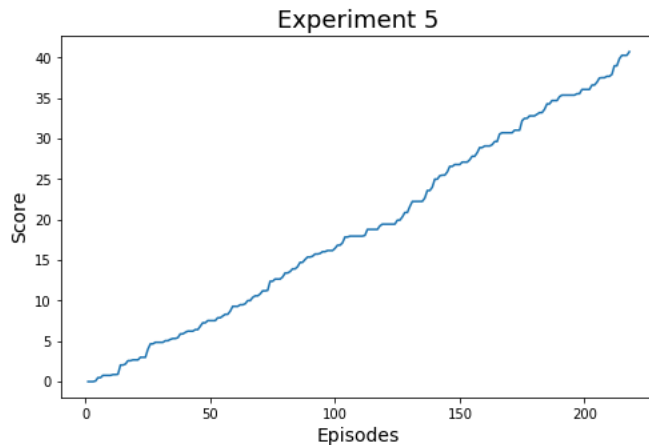
steps = 50
num_update = 50
SEED = 5
BUFFER_SIZE = int(1e6) - (replay buffer size)
BATCH_SIZE = 32 - (minibatch size)
GAMMA = 0.99 - (discount factor)
TAU = 1e-3 - (for soft update of target parameters)
LR_ACTOR = 1e-4 - (learning rate of the actor)
LR_CRITIC = 3e-4 - (learning rate of the critic)
WEIGHT_DECAY = 0.0001 - (L2 weight decay)
OUNoise Parameters: mu =0.,thetha =0.15,sigma=0.2



Environment 4 did not solve in 500 episodes! Average Score: 4.36 in 500 episodes

Parameters

steps = 30
num_update = 30
SEED = 2
BUFFER_SIZE = int(1e6) - (replay buffer size)
BATCH_SIZE = 128 - (minibatch size)
GAMMA = 0.99 - (discount factor)
TAU = 1e-3 - (for soft update of target parameters)
LR_ACTOR = 1e-3 - (learning rate of the actor)
LR_CRITIC = 3e-4 - (learning rate of the critic)
WEIGHT_DECAY = 0.0001 - (L2 weight decay)
OUNoise Parameters: mu =0.,thetha =0.15,sigma=0.2



Environment 5 solved in 118 episodes! Average Score: 30.14

Ideas for Future Work

- Optimize the parameters
- Experiment with other off-policy reinforcement learning algorithms and develop a strong understanding
- Solve Crawl challenge which is a more difficult control continuous environment

References:

<http://www.cs.sjsu.edu/faculty/pollett/masters/Semesters/Spring18/ujjawal/DDPG-Algorithm.pdf>

<https://www.cs.ubc.ca/~gberseeth/blog/demystifying-the-many-deep-reinforcement-learning-algorithms.html>

<https://arxiv.org/pdf/1509.02971.pdf>

<https://arxiv.org/abs/1604.06778>

<https://github.com/udacity/deep-reinforcement-learning/tree/master/ddpg-bipedal>

<https://github.com/udacity/deep-reinforcement-learning/blob/master/ddpg-pendulum/DDPG.ipynb>

<https://github.com/ShangtongZhang/DeepRL>

<https://github.com/vy007vikas/PyTorch-ActorCriticRL>

<https://github.com/ikostrikov/pytorch-ddpg-naf>