Solving Reacher Environment using DDPG

Abstract: DDPG is an algorithm that helps map from pixels to actions. The algorithm implements an Actor Critic network in which the actor takes in a state as input and outputs distribution of actions as output. The critic inputs a state and outputs the state value function of policy pi.

Hyperparameters :

BUFFER_SIZE = int(1e5)

BATCH_SIZE = 128

GAMMA = 0.99

TAU = 1e-3

LR_ACTOR = 2e-4

LR_CRITIC = 2e-4

WEIGHT_DECAY = 0

n_episodes=1000

max_t=10000

print_every=100

Random Seed = 10

Application :

The main goal is to train a robotic arm.

Environment:

Number of Visual Observations (per agent): 0

Vector Observation space type: continuous

Vector Observation space size (per agent): 33

Number of stacked Vector Observation: 1

Vector Action space type: continuous

Vector Action space size (per agent): 4

Sample States:

0.00000000e+00 -4.00000000e+00 0.00000000e+00 1.00000000e+00

-0.00000000e+00 -0.00000000e+00 -4.37113883e-08 0.00000000e+00

0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00

0.00000000e+00 0.00000000e+00 -1.00000000e+01 0.00000000e+00

1.00000000e+00 -0.00000000e+00 -0.00000000e+00 -4.37113883e-08

0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00

0.00000000e+00 0.00000000e+00 5.75471878e+00 -1.00000000e+00

5.55726671e+00 0.00000000e+00 1.00000000e+00 0.00000000e+00

-1.68164849e-01

This solution is not a multi-agent model. Here we implement it for a single agent and the target reward is +30/100 consecutive episodes.

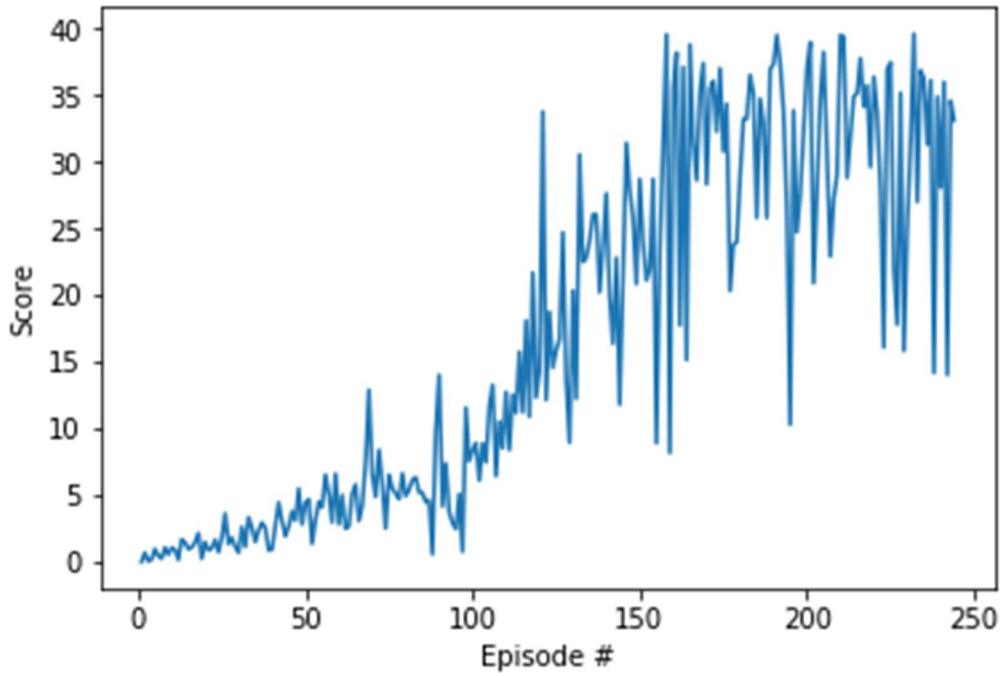
Observations:

1) The weight_decay , max_t & random seed play an important role in the training.
2) Also, there was a need for adding batch norm & dropout.
3) Changing the OUNoise parameters theta and sigma also can accelerate learning. But if we do not select the right value the results could become random.

Scope for improvement:

1) Try another algorithm like A2C , A3C and PPO.
2) Also experiment with different noise processes and different theta , sigma values for the OUNoise process.
3) Try out different architectures for both Actor and Critic models.
4) Experiment with a different set of hyperparameters.

Reward Plot:




```
Episode 100    Average Score: 3.55
Episode 200    Average Score: 23.62
Episode 244    Average Score: 30.01
Environment solved in 244 episodes!  Average Score: 30.01
```

Conclusion: Hence the Reacher environment could be solved.