Project 2: Continuous Control

In this environment, a double-jointed arm can move to target locations. A reward of +0.1 is provided for each step that the agent's hand is in the goal location. Thus, the goal of the agent is to maintain its position at the target location for as many time steps as possible.

The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm. Each action is a vector with four numbers, corresponding to torque applicable to two joints. Every entry in the action vector should be a number between -1 and 1.

The task is episodic, and in order to solve the environment, the agent must get an average score of +30 over 100 consecutive episodes.

Implementation

For this environment, a DDPG (Deep Deterministic Policy Gradient) was used, which is an amalgamation of value-based and policy-based methods using the best of both worlds. DDPG lies under the class of Actor Critic Methods but is a bit different than the vanilla Actor-Critic algorithm. The actor produces a deterministic policy instead of the usual stochastic policy and the critic evaluates the deterministic policy. The critic is updated using the TD-error and the actor is trained using the deterministic policy gradient algorithm.

Few techniques implemented which improved the overall performance were:

1. Batch Normalization – The neural network weights are normalized according to the batch size, which speeds up training and helps in better convergence
2. Soft Updates - Instead of updating after every N step like in DQN, we update a small percentage from the target network
3. Fixed Targets - 2 targets for actor and critic model similar to DQN
4. Experience Replay - We maintain a replay buffer from which we randomly sample experiences to train our model in order to avoid unwanted correlation between the sequential observations.

Network Architecture:

Actor Model	**Critic Model**
Input Layer = 33 units	Input Layer = 33 units
Hidden Layer = 256 units with RELU and Batch Norm	Hidden Layer = 256 units with RELU and Batch Norm
Hidden Layer = 256 units with RELU and Batch Norm	Action input = 4 units
Action Output with tanh activation (4 units)	Hidden Layer with input from layer 2 + action input with RELU and Batch Norm
	Q-Value output = 1 unit

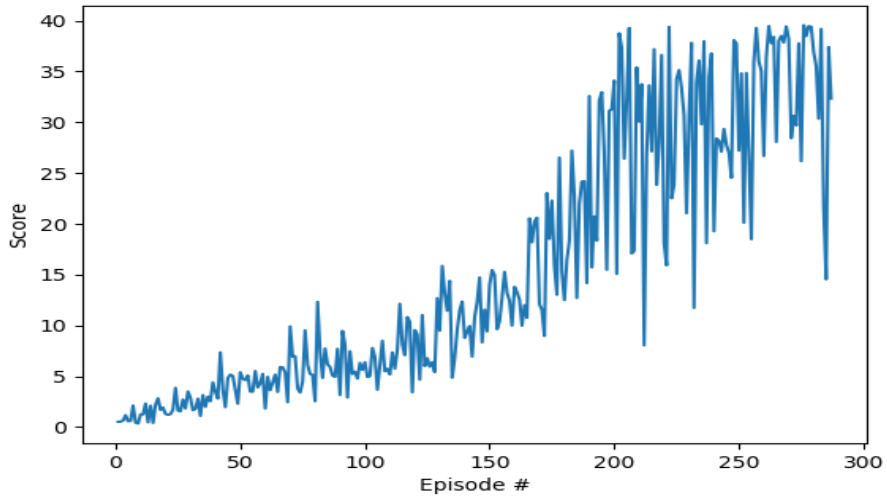
Hyper Parameters

The following hyper parameters were used:

Hyperparameter	Value
Replay buffer size	1e6
Batch size	128
Discount factor/Gamma	0.99
Target update mix	1e-3
Actor Learning rate	3e-4
Critic Learning rate	3e-4
Number of episodes	1000
Max number of timesteps per episode	1e6

Results

The best performance was achieved by DDPG where the reward of +30 was achieved in **287** episodes.



Ideas for improvement

1. Algorithms like PPO and DP4G look very promising in the literature and are expected to improve the performance.
2. Exhaustive grid search on different hyper parameters as tweaking them have improved performance in different runs
3. Using Prioritized experience replay might improve the performance by taking more advantage off the experiences that lead to greater reward.