< Project 2 – Continuous Control>

1. Learning algorithm - DDPG

DDPG is an algorithm that achieves high performance in determining actions in a continuous action space. DDPG uses two neural networks: actor and critic. The actor is used to approximate the optimal policy that outputs the best action in any state, and the critic is learned to evaluate the value function of the optimal policy using the best action output from the actor. DDPG uses a replay buffer to update neural networks, using a soft update method that mixes regular and target networks little by little without updating them at once.

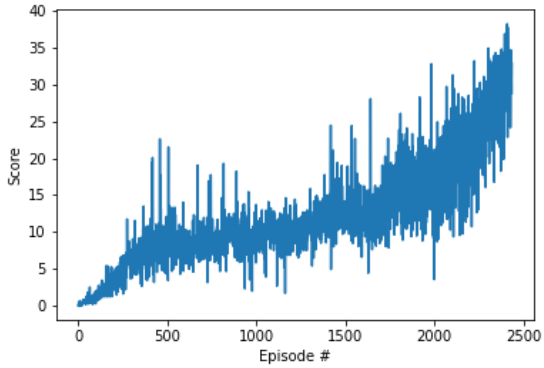
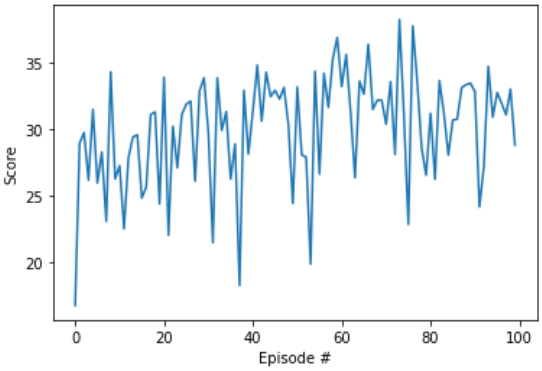
2. Chosen parameters

Hyperparameters	Value
Replay buffer size	1,000,000
Batch size	64
Discount factor	0.95
Soft update of target parameter	0.001
Learning rate of the actor	0.0001
Learning rate of the critic	0.0001
L2 weight decay	0

3. Model architecture for neural network

Actor	Critic
(fc1): Linear(in_features=33, out_features=128, bias=True) (fc2): Linear(in_features=128, out_features=64, bias=True) (fc3): Linear(in_features=64, out_features=4, bias=True)	(fcs1): Linear(in_features=33, out_features=128, bias=True) (fc2): Linear(in_features=132, out_features=64, bias=True) (fc3): Linear(in_features=64, out_features=1, bias=True)

4. Plot of reward per episode

A plot of each Episode	A plot of last 100 episode
The  Environment solved in 2335 episodes! Average Score: 30.06	

5. Idea for Future Work

Currently, I have created an algorithm that controls only one robot arm, but later I will create an algorithm that controls 20 robot arms of version 2. We will modify our algorithms to enable faster learning by adjusting the structure and hyperparameters of neural networks.

- Create Applicable Algorithms in Version 2
- Adjust Hyperparameters and Neural Network
- Using other algorithms: PPO, A2C, A3C