Objective:

To train an agent to maintain its position at the target location for as many time steps as possible. A reward of +0.1 is provided for each step that the agent's hand is in the goal location.

Environment:

The observation space consists of 33 variables corresponding to the 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 must be a number between -1 and 1.

Implementation:

1. The algorithm being used is from [this paper](https://arxiv.org/pdf/1509.02971.pdf), _Continuous Control with Deep Reinforcement Learning_, by researchers at Google Deepmind.
2. It is a model-free, off-policy actor-critic algorithm for continuous action space.
3. 20 agents are being used together to train the algorithm.
4. Replay buffer is being used to train the network.
5. Batch Normalisation has helped to achieve the results faster.
6. Used the Ornstein-Uhlenbeck process, as suggested by [Google DeepMind](https://arxiv.org/pdf/1509.02971.pdf)

Model:

We are using Actor Critic Algortihm with these specifications:

Actor:

1. Input Layer size: 33
2. Hidden Layers: [128, 256]
3. Batch Normalization
4. Hidden Layer Activation Function: Relu
5. Output Layer size: 4
6. Output Layer Activation Function: tanh

Critic:

1. Input Layer size: 33
2. First Hidden Layer size: 128
3. Batch Normalization
4. Activation function: Relu
5. Concatenation of Actor output and first hidden layer output

6. Second Hidden Layer size: 256
7. Output size: 1

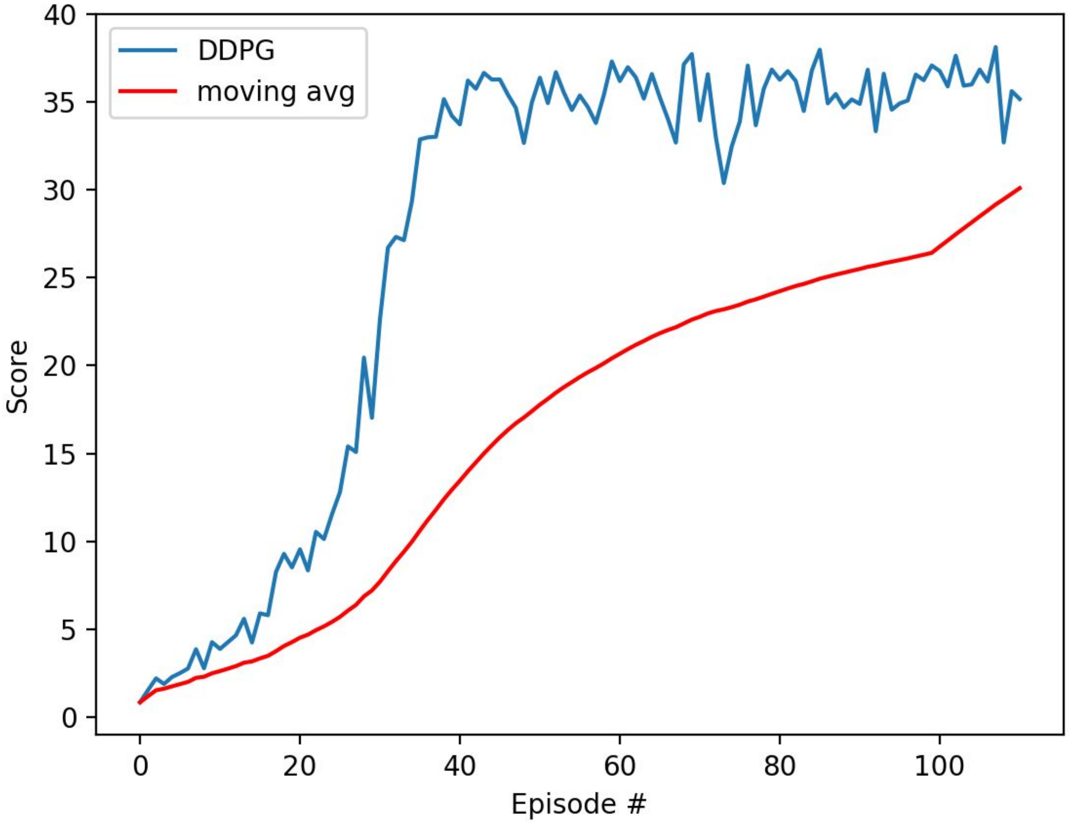
Hyperparameters:

1. Buffer Size:
    a. Description: size of replay memory
    b. Value: int(1e5)
2. Batch size:
    a. Description: number of samples being used in one iteration
    b. Value: 64
3. Gamma:
    a. Description: discount factor
    b. Value: 0.99
4. Tau:
    a. Description: factor for the soft update of the target model
    b. Value: 1e-3
5. LR:
    a. Description: learning rate
    b. Value: 5e-4
6. Update_every:
    a. Description: After how many samples we need to learn
    b. Value: 4
7. Learn-every:
    a. Description: After how many timesteps we need to learn
    b. Value: 20
8. Learn-num:
    a. Description: how many times do we have have to learn
    b. Value: 10

Output:

1. The agent took 111 episodes to reach an average reward of 30 over 100 consecutive episodes.
2. Weights are being saved at actor_ckpt.pth and critic_ckpt.pth

The graph below shows the final results:



Ideas for Future Work:

1. Use of different actor-critic algorithms.
2. Use of gradient clipping.