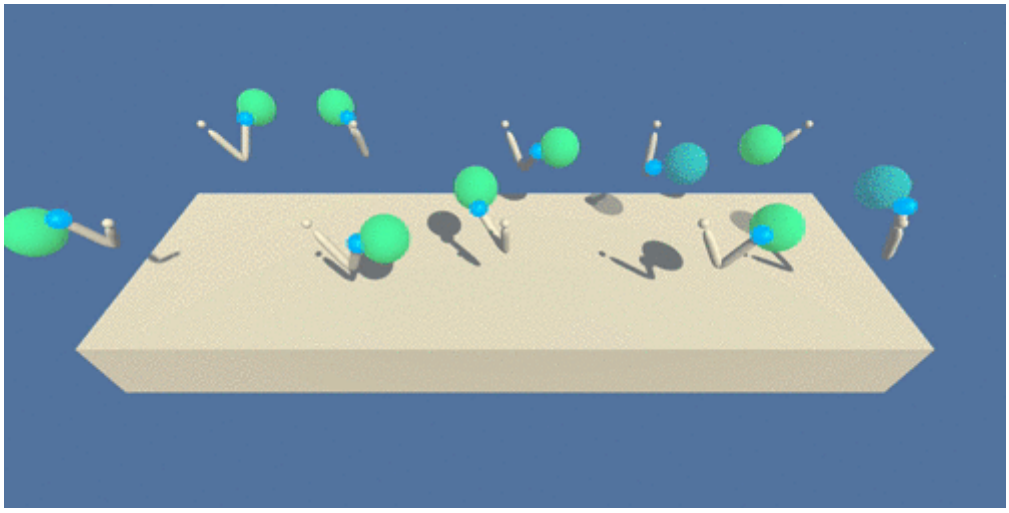
REPORT.md

Report

The Enviroment



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 your 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, your agent must get an average score of `+30` over `100` consecutive episodes.

Solution

The learning algorithm is implemented in the files `Continuous_Control.ipynb` and `ddpg_agent.py`. Once the agent solves the environment with an average score of `+30` over `100` consecutive episodes, the actor network and critic network are saved in the files `checkpoint_actor.pth` and `checkpoint_critic.pth`, respectively.

The solution is based on the Udacity implementation DDPG Pendulum.

Deep Deterministic Policy Gradients (DDPG)

The environment was solved using the amended DDPG Agent. The Actor-Network consists of two fully connected hidden layers `(33, 400)` and `(400, 300)` with ReLU activation. The final linear output layer produces a vector containing the action values for each possible action.

The Critic-Network consists of two fully connected hidden layers `(33, 400)` and `(404, 300)` with Leaky-ReLU activation. The final linear output layer maps (state, action) pairs to their corresponding Q-values.

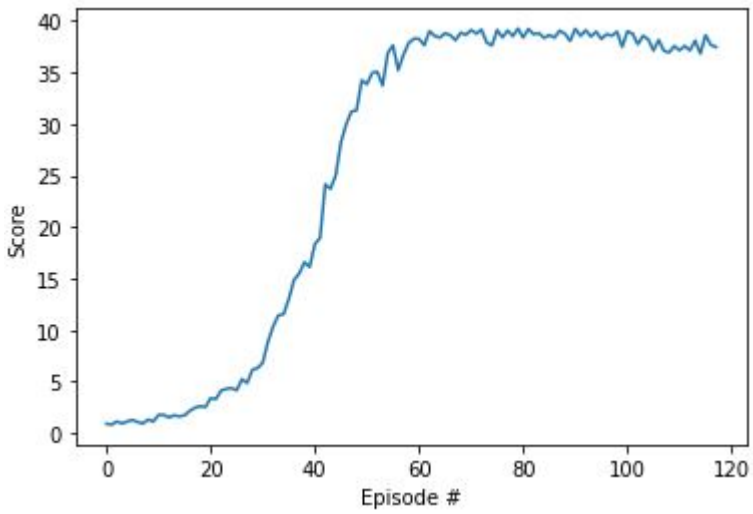
At every timestep the experiences (S, A, R, S') for each agent are stored in a replay buffer, while both the actor and critic networks are updated **`10`** times after every **`20`** timesteps.

Final Hyperparameters

```
BUFFER_SIZE = int(1e6)
BATCH_SIZE = 128
GAMMA = 0.99
TAU = 1e-3
LR_ACTOR = 1e-4
LR_CRITIC = 1e-4
WEIGHT_DECAY = 0
```

Results

Training finished in **`125`** episodes with an average score of **`30.18`** over the recent **`100`** consecutive episodes.



Future Improvements

In the future, we could implement Trust Region Policy Optimization (TRPO), Truncated Natural Policy Gradient (TNPG), or Proximal Policy Optimization (PPO) which should achieve better performance in control tasks as discussed in this paper

Additionally, by manually tweaking hyperparameters such as LR_ACTOR and LR_CRITIC finding an optimal value may be very time-consuming. Instead of manually manipulating these parameters, we could create a pipeline that automatically optimizes hyperparameters.