RL Project Report: Reach

Learning Algorithm:

The learning algorithm implemented here is the deep deterministic policy gradient (DDPG) where neural networks are used for both actor and critic components to estimate the policy during training. The hyperparameters used include the following:

- Buffer Size: 1E5 (dictates how big the replay buffer is)
- Batch_Size: 256 (dictates how many samples to pull from replay buffer for training)
- Gamma: 0.99 (used as discount factor)
- Tau: 1E-3 (used to update neural network's parameters)
- Learning Rate: 1E-3 (for both actor and critic neural networks)
- Weight_Decay: 0 (L2 weight decay)

The learning algorithm uses two neural networks to estimate the policy and q values. The actor neural network contains:

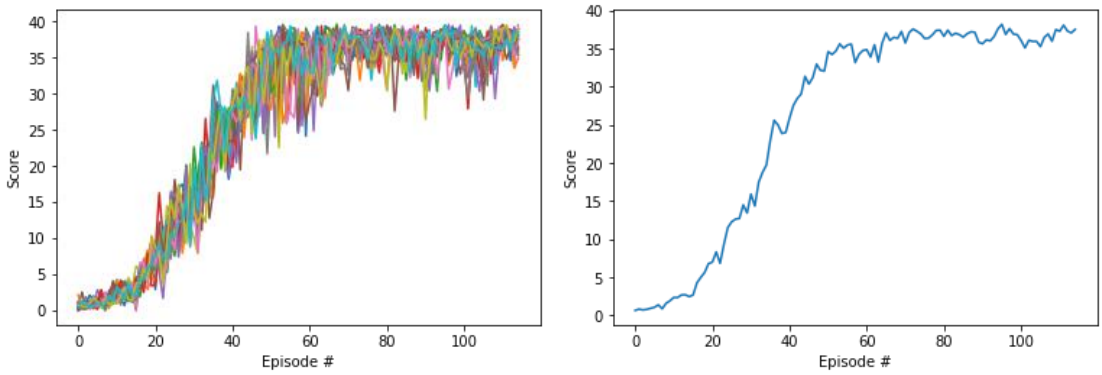
- Input Layer: state of the environment
- Hidden Layer 1: 400
- Hidden Layer 2: 300
- Output Layer: continuous value in a vector of 4

The critic neural network contains:

- Input Layer: state of the environment
- Hidden Layer 1: 400
- Extra Processing: concatenate 400 units + action_size
- Hidden Layer 2: 300
- Output Layer: 1

ReLu activation is used throughout the neural network except the last layer of the actor in which tanh is used.

Plot of Rewards:



Plot 1 on the left is the score of all 20 agents as a function of episode #. Plot 2 on the right side is the average score of 20 agents as a function of episode #.

Future Works:

1. Hyperparameter tuning that can potentially help with faster training or achieving higher scores.
2. Explore other RL algorithms such as twin-delayed DDPG or soft actor-critic method.