Collaboration and Competition

My solution is based on two DQN agents where they share memory from replay buffer during training period.

DQN can solve problems with high-dimensional observation spaces; it can only handle discrete and low-dimensional action spaces. Many tasks of interest, most notably physical control tasks, have continuous (real valued) and high dimensional action spaces. DQN cannot be straightforwardly applied to continuous domains since it relies on a finding the action that maximizes the action-value function, which in the continuous valued case requires an iterative optimization process at every step.

In order to solve our problem statement we will be using DDPG(Deep Deterministic Policy Gradient) which is an actor-critic algorithm that extends **DQN** to work in continuous spaces. Here, we use two deep neural networks, one as actor and the other as critic. Similar network architectures are used for both actor and critic.

During initial trials, my average episodes used to take to reach average reward of 0.2 used to very high(5000)and by the time I think I am near to the solution, rewards used to decrease again. After some research and suggestions in slack channel and after hyper tuning parameters I was able to solve the problem.

After trying with different units for hidden layers like 64/128 and 128/256 for Actor and Critic networks, I have used 2 fully-connected hidden layers of 128 nodes as my final setup, both trained with a learning rate of 1e-3, using batches of 512, a replay buffer of 1e6, and a discount of 0.99. Batch normalization has been added to both actor and critic networks similar to the way I have implemented in second architecture for better rewards.

Hyper Parameters:

BUFFER_SIZE = int(1e6) # replay buffer size

BATCH_SIZE = 512 # minibatch size

GAMMA = 0.99 # discount factor

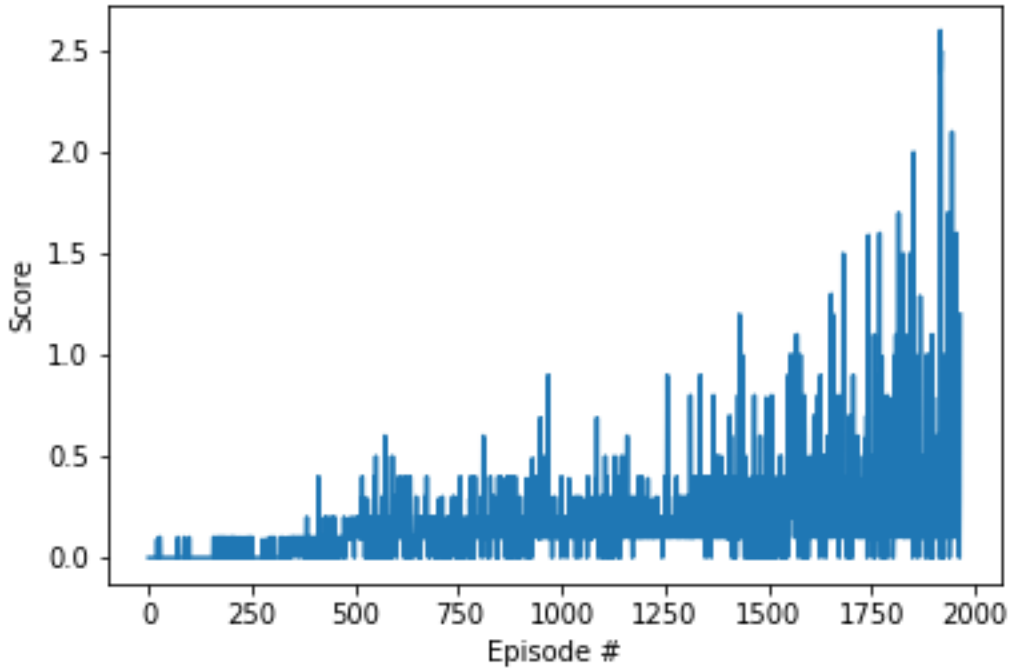
TAU = 1e-2 # for soft update of target parameters

LR_ACTOR = 1e-3 # learning rate of the actor

LR_CRITIC = 1e-3 # learning rate of the critic

WEIGHT_DECAY = 0 # L2 weight decay

Plot of Rewards:



Ideas for future work:

I want to try hyper tuning parameters more and try to solve using PPO algorithms instead of DDPG. I want to solve soccer task as well may be using similar architecture as I used here.