Syed Bilal Mehdi
Project: Collaboration and Competition
Deep Reinforcement Learning

Learning Algorithm (DDPG)

The implementation of Deep Deterministic Policy Gradient (DDPG) algorithm used for the "Continuous Control" Project was reused with some variations. The details of the final implementation are as follows:

Given the information on the "benchmark implementation" page of the project, and for keeping the problem interesting, only one agent was trained that would output actions for both the tennis players.

DDPG parameters:

A few little variations were made to the vanilla DDPG algorithm:

1. The training wasn't started until 2000 samples were already collected.
2. Although data from 2 different agents was received after every step of the environment, the local actor and critic networks were updated only once after every step of the environment.

The hyperparameters used are as follows:

Adam optimizer with a learning rate of 1e-4 and gradient clipping was used for both the actor and critic network. Batch size of 128 and a total buffer size of 1e6 was used.

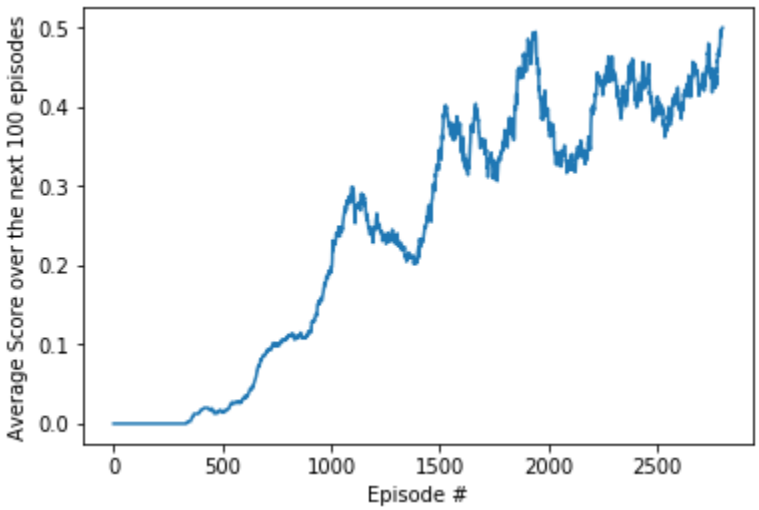
Actor Network:

Two hidden layers with 400 and 300 units each with ReLU are used. The first layer also had batch normalization. The output layer used tanh non-linearity.

Critic Network:

Two hidden layers with 400 and 300 units each with ReLU are used. The states fed the first layer, whereas, the first hidden layer and the actions fed the second layer. The output layer also used ReLU non-linearity.

Results



As can be seen, the agent trained well. However, the training is not as stable as it could be.

Future Work:

Although this version of the environment could be solved quickly with decent ranges of parameters, a more difficult variation such as soccer could be tried. A more difficult environment would also justify use of more sophisticated methods such as MADDPG.