Project Report

In this project, two agents are trained to play tennis. It is an example of a multi-agent setup that requires collaboration in a competitive environment.

Environment

The environment is a Unity Machine Learning Agents environment with a pre-compiled executable. It contains a platform with a net and two controllable tennis racks. The continuous observation space per agent is 24-dimensional and contains observables like distance from the net. Each agent's continuous action space is 2-dimensional, where one action corresponds to movement towards and away from the net and the other action corresponds to jumping.

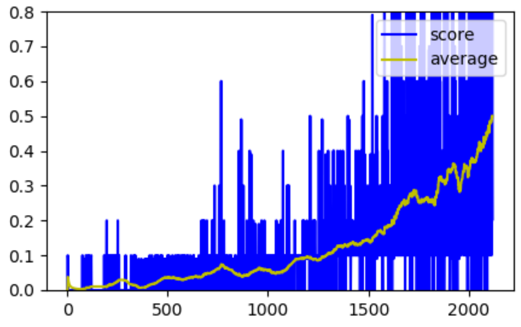
Learning Algorithm

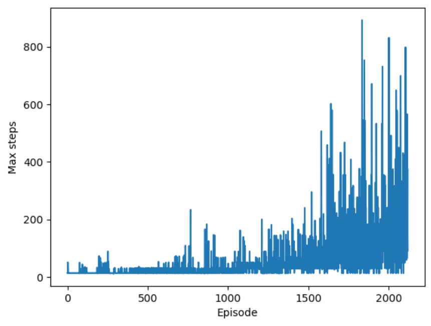
The agents are trained with a multi-agent model using Deep Deterministic Policy Gradients (DDPG). The idea of DDPG is to take the standard Deep-Q Learning approach to the continuous domain by using an actor-critic model, where the actor deterministically produces actions for a given state, while the critic tries to evaluate those actions in combination with the state. During training, the actor is optimized towards generating actions that result in high critic values. Both actor and critic are realized as fully connected feed forward artificial neural networks, with two hidden layers at 128 units each, with ReLU activation. The output layer of the actor uses hyperbolic tangent activation, while the critic output is a simple linear layer. Choosing fewer than 128 neurons in the hidden layers seems to decrease training performance. As suggested by the slack community (and contrary to the approach showed in the online lessons), the critic takes the action and state vectors together as inputs in the first layer, instead of pushing the state vector through one fully connected layer before concatenating with the action vector.

The algorithm achieves exploration by randomizing actions with an Ornstein-Uhlenbeck process. The hyperparameters used for training can be found in the Readme.me file, together with links to relevant papers. The code is mostly adapted from the DRLND github repository DDPG examples and the MADDPG lab exercise. Getting the hyperparameters right took serious effort and training success only came after following other people's suggestions from slack. Especially setting the L2 weight decay to zero seems necessary. Furthermore, a moderately increased value for tau and small values for actor and critic learning rates seem necessary for stable training.

Rewards

The agent was able to reach the required 100-episode-average maximum reward of 0.500 in 2119 episodes. The following plots show that the reward and the number of time steps played increase steadily, but with significant variance. Even after 2000 episodes of training,





the agents occasionally drop the ball before playing it over the net even once. This shows that this multi-agent environment poses a significant challenge for current algorithms.

Future Work

Since the agent learns very slowly in the beginning, it might be beneficial to increase the randomness of actions for better exploration. Furthermore, the environment seems to be well suited for imitation learning, for example by collecting experience from human-played episodes.