Implementation Report: Project 3 Collaboration and Competition

Deep Reinforcement Learning Nanodegree

Thomas Gallien

October 28, 2019

Problem Description

The objective is to solve a multi-agent environment where two agents control a racket in order to bounce a ball over a net in a Tennis-like fashion.

Environment

The environment consists of 2 instances of a racket being controlled in order to bounce a randomly appearing ball over a rack. Figure 1 illustrates the configuration.

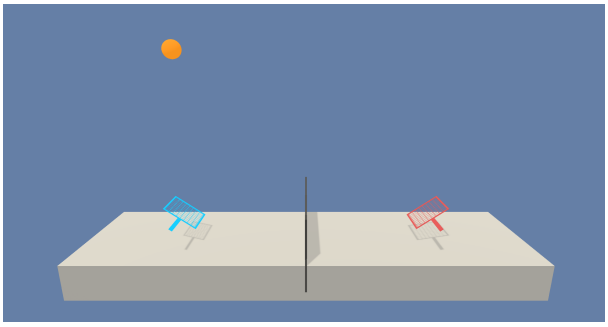

Figure 1: Unity's tennis environment utilizing 2 rackets.

Observation Space

For each racket a 24-dimensional observation vector can be observed corresponding to the positions and velocities of the ball and rackets.

Action Space

The action space is 2-dimensional and continuous (numbers between -1.0 and 1.0) for each racket corresponding to the movement toward (or away from) the net, and jumping.

Reward

If an agent hits the ball over the net, it receives a reward of 0.1. If an agent lets a ball hit the ground or hits the ball out of bounds, it receives a reward of -0.01.

Goal

The environment is considered solved, when the average (over 100 episodes) of the maximum of the individual scores is at least 0.5.

Training Algorithm

The task was used by instantiating two Deep Deterministic Policy Gradient (DDPG) agents (see Lillicrap et.al. [1] for details) independently. Surprisingly, the chosen approach showed to be superior in terms of convergence to multi-agent approaches using fingerprinting (see Foerster et.al. [2]) or Multi-Agent Deterministic Policy Gradient(MADDPG) (see Lowe et.al. [3])

Architecture

The architecture of the actor and critic networks was chosen similar to the supplementary information shown in the DDPG paper [1].

Actor Network

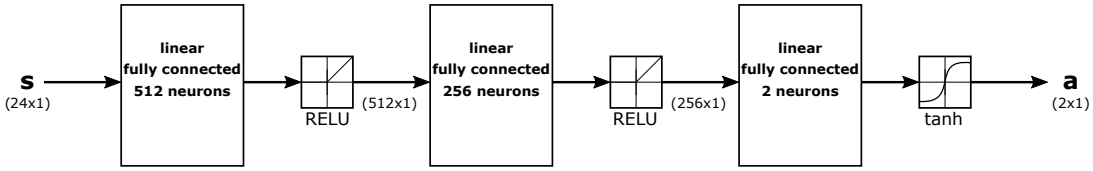

Figure 2: Forward path of the agent's fully connected actor network(s).

Critic Network

The architecture of the forward path is shown below. For training, a dropout layer applying dropout with a probability of 0.2 is used.

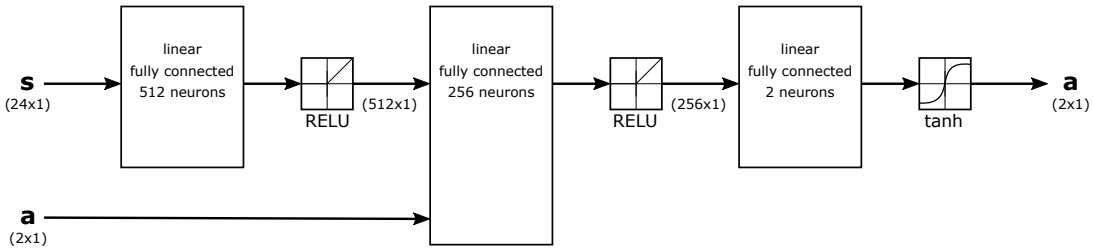

Figure 3: Forward path of the agent's fully connected critic network(s).

Hyperparameter

The table below shows the set of hyperparameter used to solve the Tennis environment. L2 regularization mentioned in Lillicrap et.al. [1] for the optimization of the critic did not reveal in an increased performance. Hence, the weight decay parameter was set to 0.

Hyperparameter	Abbreviation	Value
Buffer size replay memory	M	10^5
Batch size	N	512
Discount factor	γ	0.99
Learn rate actor network	α_{actor}	10^{-4}
Learn rate critic network	α_{critic}	$3 \cdot 10^{-4}$
Interpolation parameter soft update	τ	$2 \cdot 10^{-1}$
Update rate	l	5
Iterations per update	m	10
Decay rate UO-noise	θ	0.15
Standard deviation UO-noise	σ	0.2
Dropout probability critic	p_{drop}	0.2

Table 1: Used set of hyperparameter for each DDPG agent solving the Tennis environment.

Results

The figure below shows the scores calculated by averaging the maximum of both individual immediate rewards during training (blue line) and the corresponding moving average with respect to 100 filter taps. As can be easily seen, the agent approaches the desired number 0.5 of the average score after 438 episodes.

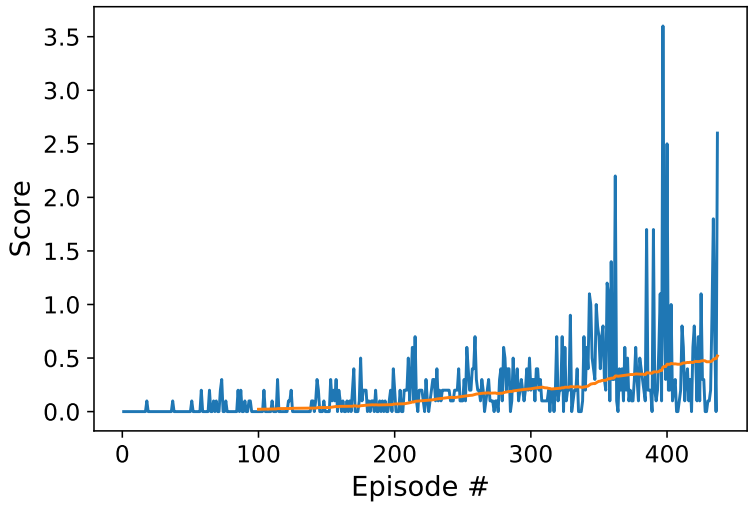

Figure 4: Performance of the double DDPG agent approach in terms of episodic cumulative reward (maximum of immediate scores, blue line) for 438 episodes and corresponding moving average filtered signal (red line).

Conclusion & Outlook

The approach of using two independent DDPG agents to tackle the multi-agent Tennis environment revealed in surprisingly well performing policy networks. However, dedicated multi-agent approaches such as fingerprinting and MADDPG proved to perform worse which raises the question why is this the case. Hence, for future work this question will be addressed. Moreover, algorithms in the context of Proximal Policy Optimization (PPO) [4] and Trust Region Policy Optimization (TRPO)[5] will be investigated in terms of their capability to solve multi-agent environments.

References

[1] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel et. al. *Continuous Control with Deep Reinforcement Learning*. Proceedings of the 2016 International Conference on Learning Representation (ICLR), 2016.

[2] J. Foerster, N. Nardelli, G. Farquhar et. al. *Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learnings*. Proceedings of the 34$^{\text{th}}$ International Conference on Machine Learning (ICML) 2017.

[3] R. Lowe, Y. Wu, A. Tamr et. al. *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. Proceedings of the 31$^{\text{st}}$ Conference on Neural Information Processing Systems (NIPS), 2017.

[4] J. Schulman, F. Wolski, P. Dhariwal et. al. *Proximal Policy Optimization Algorithms*. http://arxiv.org/abs/1707.06347, 2017.

[5] J. Schulman, S. Levine, P. Moritz et. al. *Trust Region Policy Optimization*. Proceedings of the 2015 International Conference on Machine Learning (ICML), 2015.