Collaboration and Competition Project Report

Udacity Nano Degree Program

Student:

Nikhil Masinete

Course:

Deep Reinforcement Learning

Introduction

Reinforcement Learning is a branch of machine learning where a software agent is taught to how to take actions in an environment in order to maximize its cumulative reward. This theory is fundamentally based on Markov Decision Process. This concept resembles the way living beings live. The following are the terms which are frequently used in this report.

Agent: An agent is the smart being which is trained to achieve tasks given to it.

Environment: An environment is a situation where the agent lives.

State: A state is a quantified description of a situation.

Action: An action is the way how the agent should respond in each state.

Reward: A reward is the outcome given to the agent for taking a particular action in each state. A reward is positive or negative based on the action taken was desirable or undesirable.

State-Action value function (Q function): A state action value function is a matrix that stores the expected reward when a particular action is taken in each state.

Policy: The policy is the strategy that the agent employs to determine the next action based on the current state.

Episode: All states that come in between an initial-state and a terminal state.

Challenge

In this environment, two agents control rackets to bounce a ball over a net. 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. Thus, the goal of each agent is to keep the ball in play.

Algorithm

To solve this challenge Multi Agent Deep Deterministic Policy Gradients algorithm[1] was used. DQL algorithm is best suited for environments where the environment has a low discrete action space. In this scenario the action space is a continuous one. So DDPG was required. Generally, DDPG algorithm consists of an actor and a critic. An actor decides the which action to be taken for a given state. A critic is good at estimating the worth of action taken and able to predict the future rewards. Policy gradient methods is used as an actor and Monte-carlo estimates are used to determine the future rewards. The combination of these two models is DDPG algorithm. In solving the current environment, the following hyper parameters were used.

The main difference between DDPG and MADDPG is that, in MADDPG the actions taken by all agents during training. This makes learning faster and leads to a robust policy.

The hyperparameters used in this algorithm are

Buffer_size = 1,000,000

Gamma = 0,99

Tau = 1e-2

LR_CRITIC = 1e-3

LR_CRITIC = 1e-4

Weight_Decay = 0

Num_agents = 2

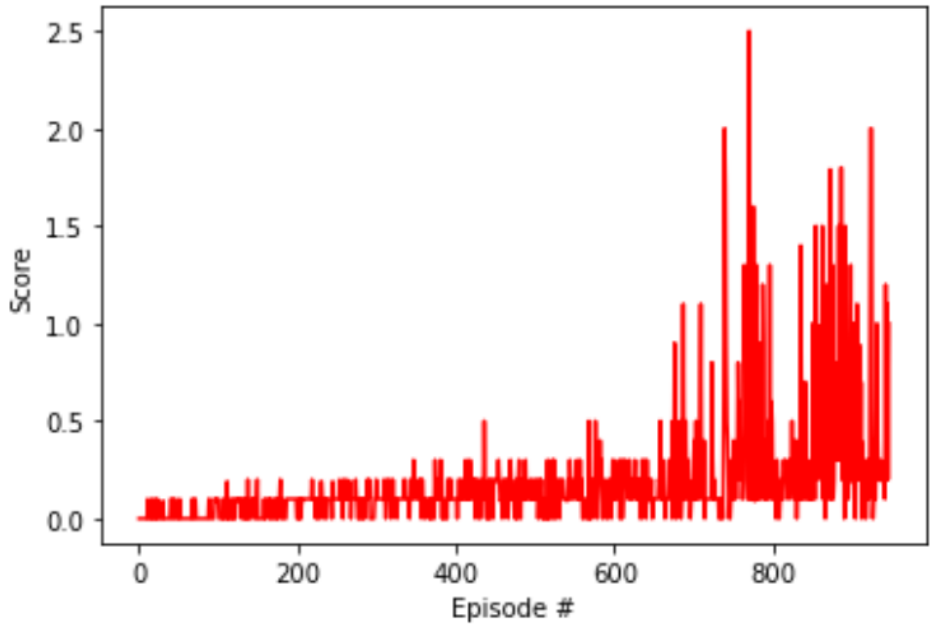
Update_every = 1

A three layer neural network of units fc1 =512, fc2 = 256, fc3 = 128 is used for actor.

A three layer neural network of units fcs1 = 512, fc2 = 256 is used for critic.

Result

The environment was solved in 945 episodes.



Future Improvements

This problem can also be solved using Proximal Policy Optimization (PPO), Asynchronous Advantage Actor-Critic (A3C).

References

1. Multi Agent Actor Critic for Mixed Cooperative Environments.