Agent Collaboration

1 Introduction

In this project, multi-agent deep deterministic policy gradients were implemented, similar to DQN and DDPG, MADDPG is used for multiple agents that learn simultaneously. Each agent only has direct access to local observations. While each agent only has local information and local policies to train, there is an entity that oversees the entire system of agents, advising them on how to update their policies.

Like DDPG, the algorithm also uses the replay buffer for efficient out-of-policy training, where it stores experiences such as state, actions, and rewards. Another tool would be to update an agent's centralized critical, we use a one-step look-ahead TD error.

This algorithm is similar in that it derives from Deep-Q Learning, but to be used across multiple agents, just like DDPG solves the problem for continuous control environments.

2 Method

Networks of actors and critics with continuous action space are used. In the critical case, a fully connected layer produces the state value. For the actor, a fully connected layer provides the average of a normal distribution for each action.

The input to the actor network consists of an 8 × 2 matrix produced by the environment (the state space size times the number of agents) the first layer is fully linearly connected with 512 neurons. It is followed by a hidden layer also composed of a fully connected linear layer with 400 neurons. The critical network is composed of a linearly connected layer of 512 neurons, with an output also of 400 neurons.

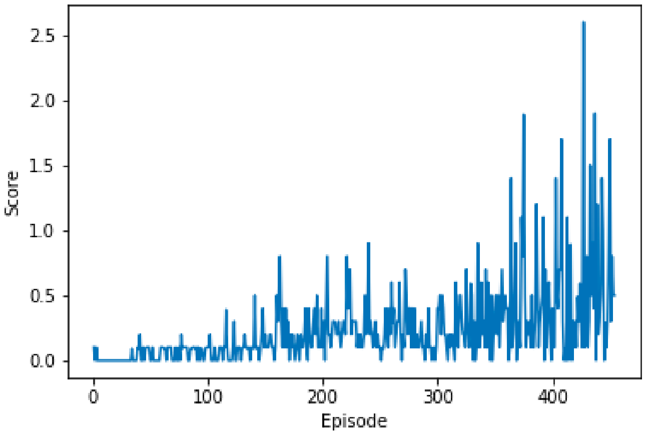
The output layer of the critical network includes a fully connected linear layer with a single output for each agent followed by a tanh activation function that naturally constrains values between [1, 1].

The hyperparameters used for training in this environment are:

- The size of the repeat buffer: Repeat buffer = 2e5
- The actor's learning rate: Learning rate = 5e-4
- The critic's learning rate: Learning rate = 5e-4
- The discount factor: Gamma discount = 0.99
- The size of the minilot: Lot Size = 256
- The soft refresh parameter of the target:Tau = 0.3
- Exploration: Epsilon = 1

3 Results

The MADDPG agent takes 454 points to reach an average reward of 0.5 points in about 1 hour of training on the CPU instance. The stop condition is met once, as are the validity evaluation points of 0.5. The training results are shown in the figure.



4 Improvements

Although MADDPG presents good results for the environment with only 2 agents, there are other algorithms that approach this problem with different techniques. They are:

- Trust Region Policy Optimization (TRPO)
- Truncated Natural Policy Gradient (TNPG)
- Proximal Policy Optimization (PPO)
- Distributed Distributional Deterministic Policy Gradients (D4PG)