

Report

This project MADDPG model, agent implement project. (Tennis)

Learning algorithm

MADDPG (Multi Agent Deep Deterministic Policy Gradient)

I modified ddpq_agent, model in continuous-control project. DDPG is an actor-critic algorithm; it has two networks: actor and critic. Multi agent just use multiful Agent. It is almost same with continuous-control project.

Technically, the actor produces the action to explore. During the update process of the actor, TD error from a critic is used. The critic network gets updated based on the TD error similar to Q-learning update rule.

Hyper Parameters

BUFFER_SIZE=int(1e6): replay buffer size

BATCH_SIZE=256: minibatch size

GAMMA=0.99: discount factor

TAU=1e-3: for soft update of target parameters

LR_ACTOR=1e-3: learning rate of the actor

LR_CRITIC=1e-3: learning rate of the critic

WEIGHT_DECAY=0: L2 weight decay

EPSILON_DECAY = 0.99

Model architecture

Actor

Two hidden layer fc1 size 256, fc2 size 128, and use relu for each hidden layer

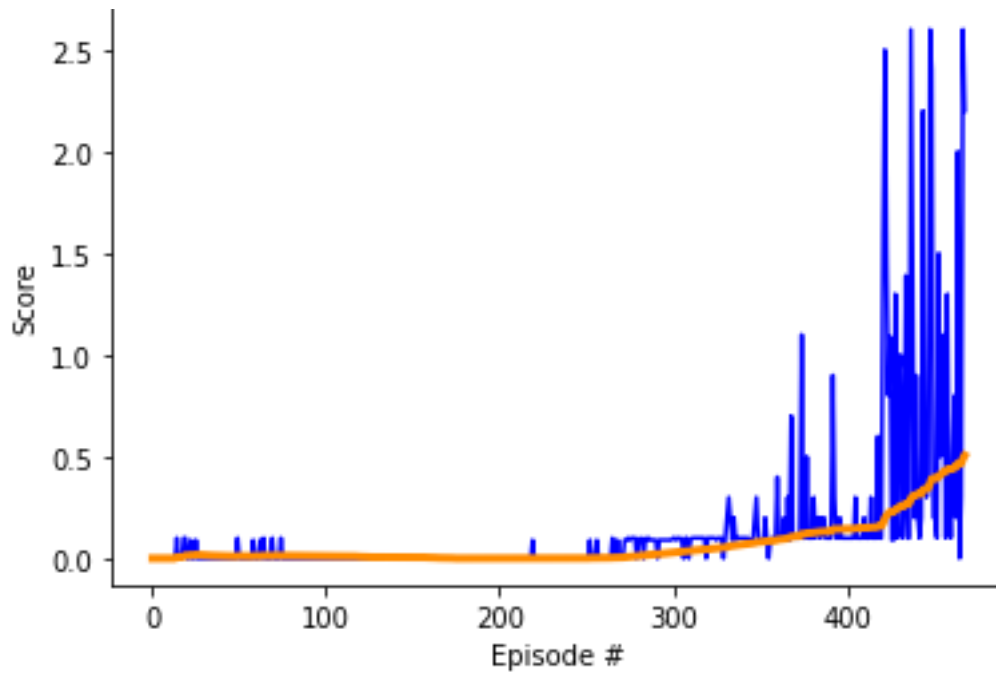
Finally connect to tanh activation function. And use batchnorm

Critic

Two hidden layer fc1 size 256, fc2 size 128, and use relu for each hidden layer

And use batchnorm

Plot of Reward



```
Episode 100      Mean_reward: 0.00 Average100 Score: 0.01
Episode 200      Mean_reward: 0.00 Average100 Score: 0.00
Episode 300      Mean_reward: 0.09 Average100 Score: 0.03
Episode 400      Mean_reward: 0.10 Average100 Score: 0.15
```

Environment solved in 469 episodes! Average100 Score: 0.51

Future Improving Idea

We can change and test some hyper parameter. And I think this MAagent project more important model config. Test and change hidden layer parameter, and activation function.