Collaboration and Competition

Project 3

engmuhammadhakami@gmail.com

2021-11-01

Contents

1 Problem

The problem at hand is the collaboration problem with two tennis agents bouncing the ball as goal to keep it floating. the start ball location is random and changing every episode reset.

2 Algorithmic Choice

The problem is a continuous state-space of 24 values, and a continuous action-space with 2 values per agent.
Such problem can be solved with discretization or with policy based approaches. I decided to go with Soft Actor-Critic (SAC) which is an off-policy algorithm that was published in 2018 by Berkeley team. it is similar to Deep Deterministic Policy Gradient (DDPG) which is an Actor Critic method that is based on Deep Q-Networks (DQN) algorithm that can handle continuous action spaces. the main contribution within SAC is the entropy regularization which add bonus reward proportional to the policy's entropy of current time step. SAC consist of 4 networks. an Actor, a Critic, and two Q networks. The Actor network(known as policy network) is responsible for deciding the best action. While the Critic criticize the chosen action producing the value function which measures the action decided by the Actor. followed by two Q networks that is used to train the critic.

3 Model Structure

Tried multiple model structures for both Actor and Critic networks, the best performance i got was similar structure for both. With batch size of 64 training from replay buffer.

- The Actor network structure consists of 3 linear layers with units of 64. along with tanh activation. for each agent with shared optimizer.
- The Critic network structure consists of 3 linear layers with units of 64. along with relu activatation.
- The two Q-networks consists of similar structure. it has three linear layers of unit size 64.

4 Hyperparameter Specified

- While tuning the parameters i notice that the weight decay(L2) actually stagnated the learning. so i turned i it off.

- i tried increasing the batch size along with layer units, but it was taking longer to converge so i lower it to 64.

- Likewise for learning rate. i found it to preform better with similar learning rate for both structures and finalize it as 1e-4.

- For the SAC entropy alpha. i went with 0.1 to encourage stability. as lower numbers kept rewards going back and forth.

5 Environment based optimization

First start to configure the training network and training episode function to start training and investigating.
i started went with the code of SAC from my favorite book about DeepRL and changed things to account for tennis environment and MARL structure of sharing states. i had to convert the enviroment into gym wrapper to integrate with ptan library.

I started training the model with shared reward but separate states. but during that each agent couldn't understand scenarios were the other agent messed up. Leading to difficulty to solve the environment. where in some scenarios the agent would not intercept the ball.

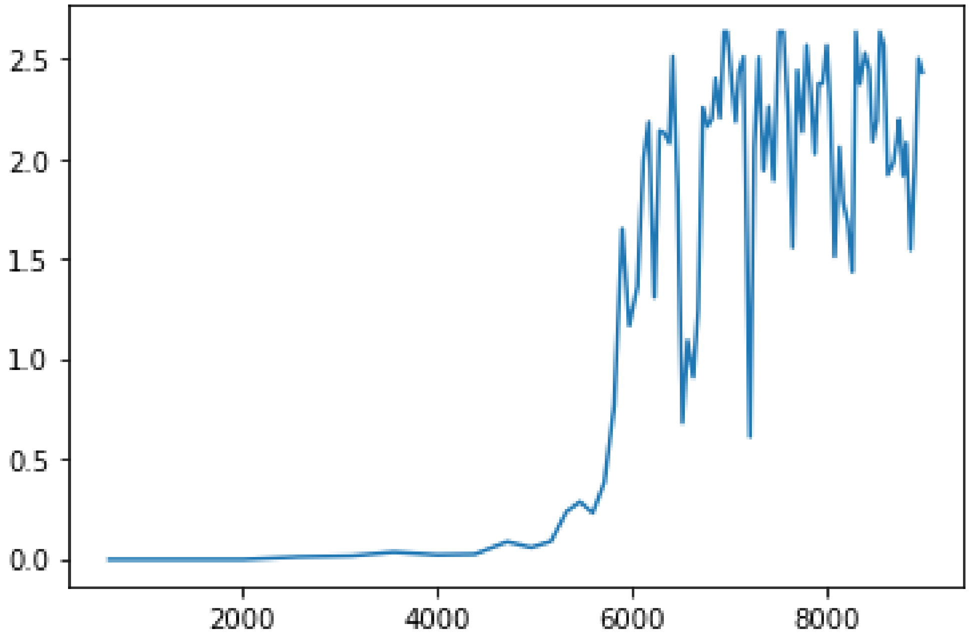
Given that the Environment is collaborative. i decided to share the states of both agents with each other but still go with shared reward. even though one agent would be penalized for the mess of the other. Surprisingly the agents start cooperating to make it easier for one another. and the result were good.

After that i decided to split reward for each agent to be penalized alone if it messed up. i retrained after that and got the highest performance model allowing me to clear the Environment.

6 Results

Given that the model is trained over off-policy method. it took a bit more episode before could converge. but it gave great results soon after.

Here is the score for the 2 agents during training:



you can see that the performance was somewhat stable after 6k episodes leading convergence

The current system will do perfectly against most initial ball locations. but there are still some niche start locations that the model would be bad at. i expect after longer training session that we would have even better generalization for those niche starting locations. the model cleared the scenario in about 5.8k episodes

7 Future Work

There are still initial state locations that the model is bad at. but they are very rare. i'm interested to train more and see if the model can figure those out. also i wanted to test some other model structures like(PPO, TRPO, ACKTR, etc..) i expect SAC to be better in generalization but interested to see how they compare. there is also some new algorithms that poped out these years like MAPPO(2021). i want to test it out with the current system in the future.