Programming Assignment 2
MADDPG

Ji In Kwak
Carnegie Mellon University
jiink@andrew.cmu.edu

November 07, 2022

1 Task 1 : Implementation of MADDPG

In maddpg.py file, the code for implementation of task 1.1 to 1.6 are presented.

2 Task 2 : Testing the Implementation

I have implemented the MADDPG in the given environment with two scenarios, which are simple and ranger-poacher. In simple scenario which is a task 2.1, there are one agent navigating towards the landmark. The plot of the learning curve representing the average rewards is shown in Figure 1.

In task 2.2, we need to train the MADDPG in ranger-poacher scenario. There are three agents each represents a poacher, ranger, and UAV. Also, there are 8 wild animals in the fixed location. The model was trained for 3000 episodes and the learning curves for each agent is shown in Figure 2.

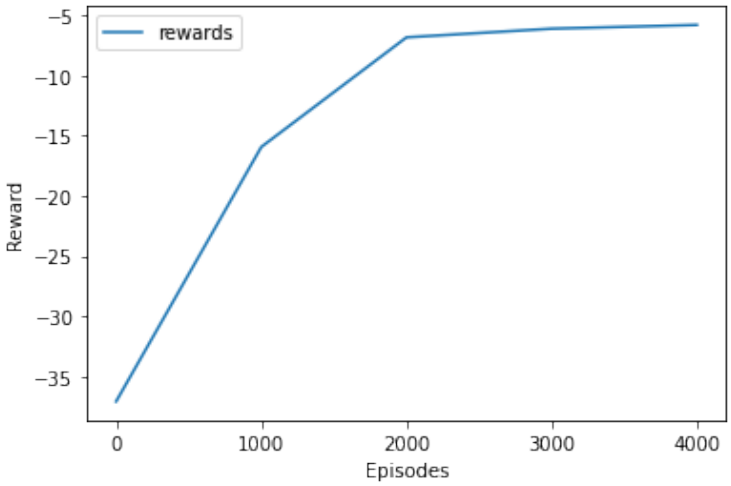

Figure 1: Learning curve of the mean episode reward in simple scenario

3 Task 3 : Reward Shaping

In the result scenario after implementing MADDPG in task 2, trained three agents pretended to go outside the given environment world. To improve this problem, I added the distance measurements in the reward function. If the game did not end since any agent cannot caught anyone, then we calculate the distances from the origin point for ranger and uav as follows.

$$dist(ranger, origin) = \sqrt{\sum(ranger.positions)} \tag{1}$$

$$dist(uav, origin) = \sqrt{\sum(uav.positions)} \tag{2}$$

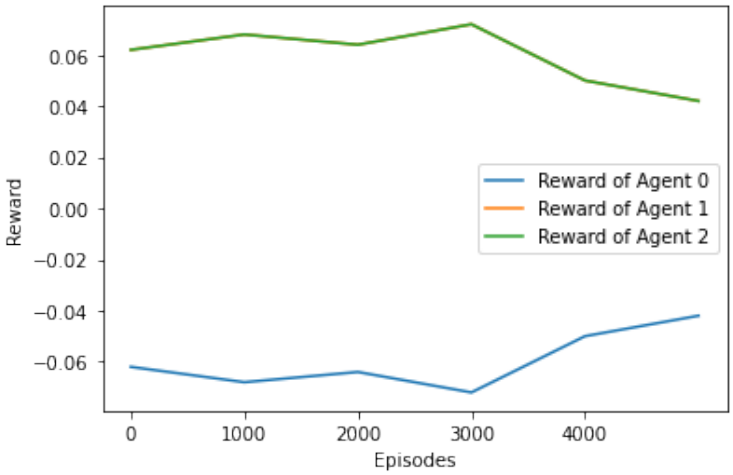

Figure 2: Learning curve of the mean episode reward in ranger-poacher scenario

If the distance is larger than 0.8, the negative reward -0.005 is given to each agent. For the case of poacher, I gave the reward for the minimum distance between the landmarks. By calculating the distance between poacher and every 8 landmarks, the positive reward is given to make it closer to one of the landmarks as follows.

$$distances[i] = \sqrt{\sum(poacher.positions - landmark[i].positions)} \tag{3}$$

$$reward\ for\ poacher = \exp(-min(distances)) \tag{4}$$

The third approach gives negative reward to UAV if the ranger cannot caught the poacher even if UAV is watching it. The given reward value is -0.15.

The learning curve for this reshaped reward function is shown in Figure 3. In the generated gif, compared to the result in task 2, it seems to perform better for poacher case. However, in case of ranger, it does not follow the poacher although the poacher is visible to UAV. It means that the negative reward function for visibility of UAV is not working well.

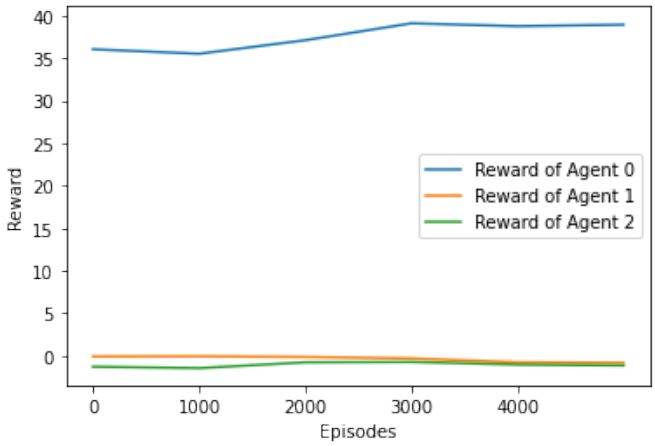

Figure 3: Learning curve of the reshaped reward function in ranger-poacher scenario