Competitive Environment Training using Deep Deterministic Policy Gradient (DDPG) agent

Author: Henry Chan

Learning Algorithm

This project makes use of the DDPG agent used in my previous project "Learning Continuous Control in Deep Reinforcement Learning". The difference is it deals with a more complex competitive environment involving 2 tennis players competing with other.

The learning algorithm is based on the paper "Continuous control with deep reinforcement learning" with my own modification based on my experiment and intuition in solving the "Tennis ML Agent" environment. My DDPG implementation is modified from the vanilla DDPG agent in solving single agent pendulum environment.

Inspired from the "AlphaGo Zero" that learns to play Go game by just training the agents playing against each other from scratch, I attempted to use the DDPG agent developed in the previous project and applied it to the competitive Tennis environment with 2 players using single DDPG agent (single brain). The DDPG agent collects experience from both 2 players with shared "Reply Buffer". A couple of hyper parameter / Neural Network modifications are made to get the training more efficiently. Details are below.

I also used the AWS EC2 p2-xlarge GPU instance for better training performance.

Model Architecture

The Actor model is a neural network with 2 hidden layers with size of 512 and 256, Tanh is used in the final layer that maps states to actions.

The Critic model is similar to Actor model except the final layer is a fully connected layer that maps states and actions to Q-values. A dropout of 0.2 is added before the output layer to help the network to learn more efficiently and avoid overfitting.

Hyperparameter

The learning rate for Actor is 1e-4 while Critic is 3e-4 with soft update of target set to 2e-1. I found that a bigger soft update results in faster training.

Gamma discount is 0.99 with weight decay set to 0. Replay Buffer has size of 1e5. Batch size is 512 with max time step of 2000 in each episode. A large Replay Buffer is crucial in the success of the learning. Number of time steps plays an important role as the agent needs to have enough time steps to have a good balance between exploitation and exploration while training efficiently. The agent starts learning every 5 steps. Ornstein-Uhlenbeck noise is used.

Plot of Rewards

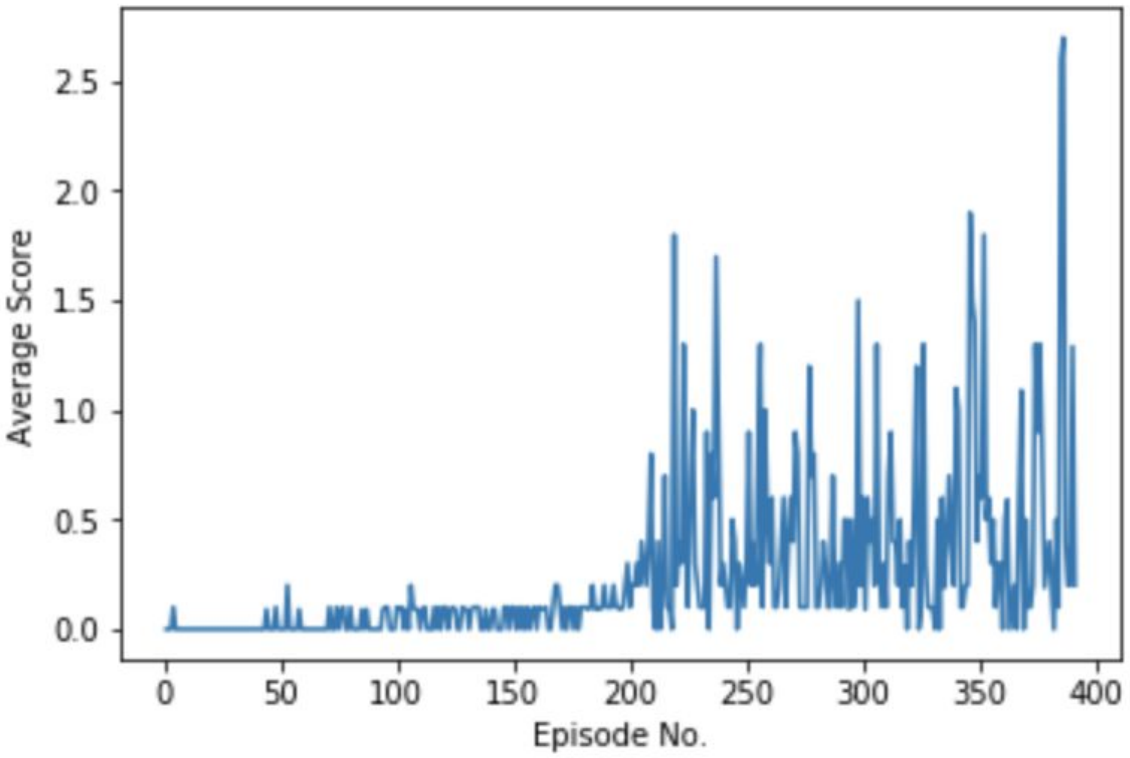
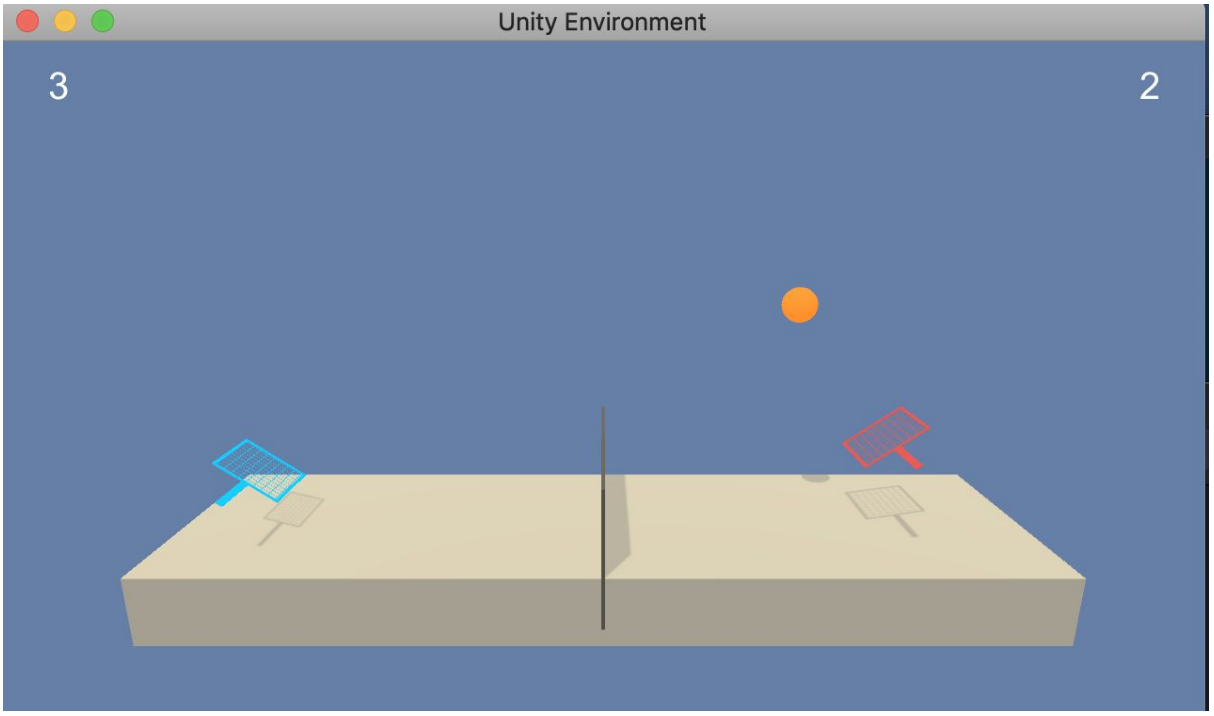
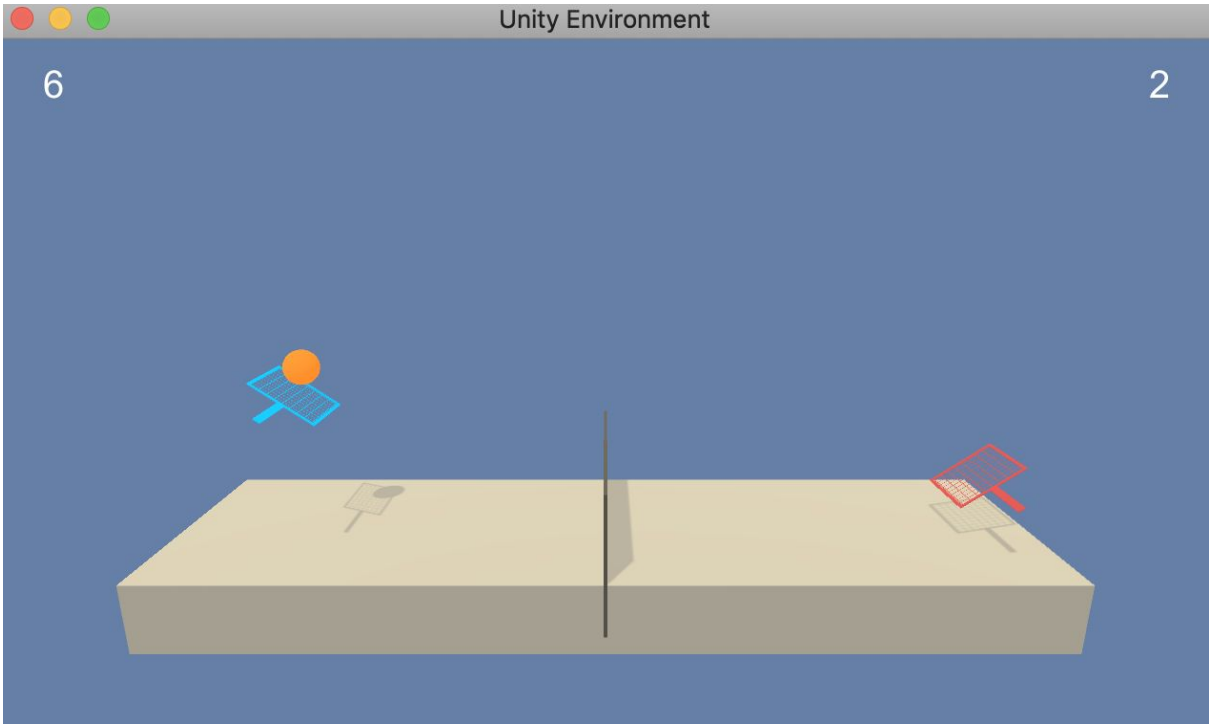


Figure 1 - No. of episodes / Scores

The DDPG agent takes 391 episodes to achieve an average max rewards of 0.5 scores in about 1 hour 15 mins trained in GPU instance. The score fluctuates a lot after 210 episodes but the average max score rises gradually. A sharp increase of scores from 0.13 to 0.40 happens between 220th and 300th episode. It stays at score at about 0.41 between 300th and 345th episode as if it was exploring new techniques of playing tennis. Then once it mastered the the new techniques, the score jumps significantly to 0.50 in the last 45 episodes. I am sure it continues to train it will achieve even better scores.

Agents Playing Tennis





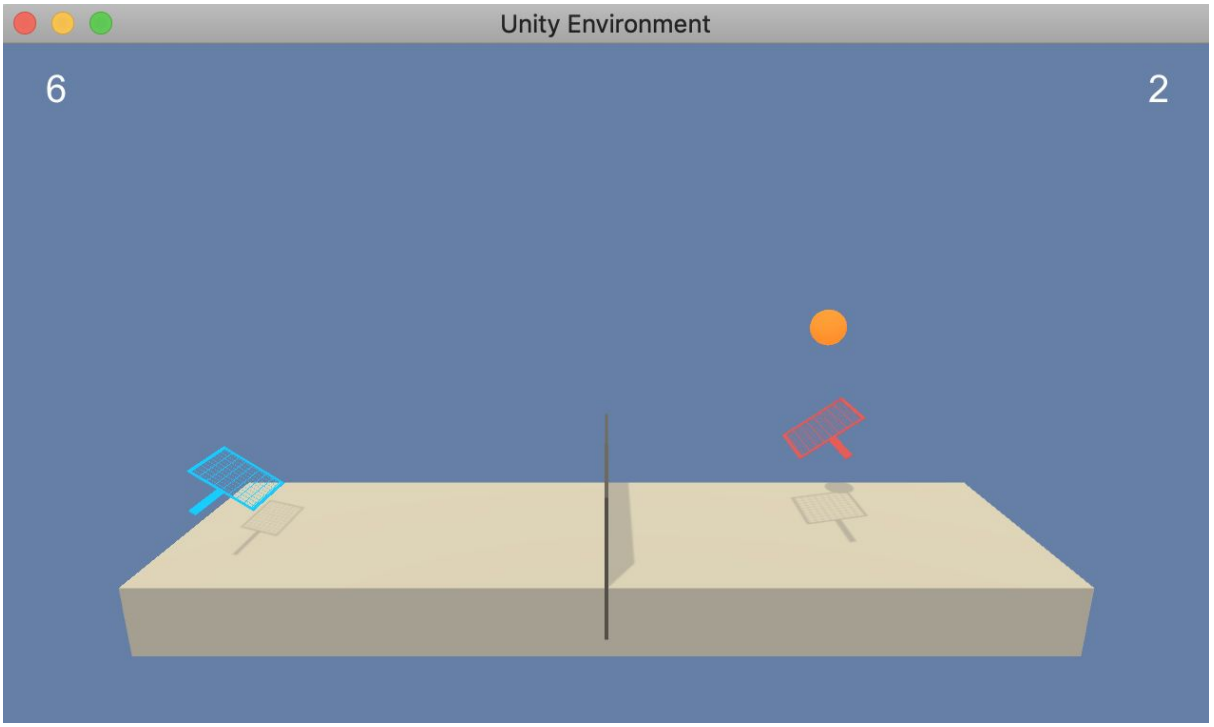


Figure 2a,b,c - Running the DDPG Tennis agent with saved weights

Future Improvements

Novel DDPG algorithm can be improved by applying Priority Experienced Replay based on the paper "A novel DDPG method with prioritized experience replay", it can reduce the training time, improve the stability of the training process and is less sensitive to the change in hyperparameters. We could also try training with individual DDPG agent using separate Actor / Critic networks. The other way is to explore "Parameter Space Noise for Exploration" by adding noise directly to the agent's parameter. It enables the agent to train more efficiently in continuous control task with DDPG.

Credits

- Continuous control with deep reinforcement learning https://arxiv.org/abs/1509.02971
- Deep Deterministic Policy Gradient Actor-Critic Model https://github.com/udacity/deep-reinforcement-learning/tree/master/ddpg-pendulum
- A novel DDPG method with prioritized experience replay https://www.semanticscholar.org/paper/A-novel-DDPG-method-with-prioritized-experience-Hou-Liu/027d002d205e49989d734603ff0c2f7cbfa6b6dd
- Parameter Space Noise for Exploration https://arxiv.org/abs/1706.01905