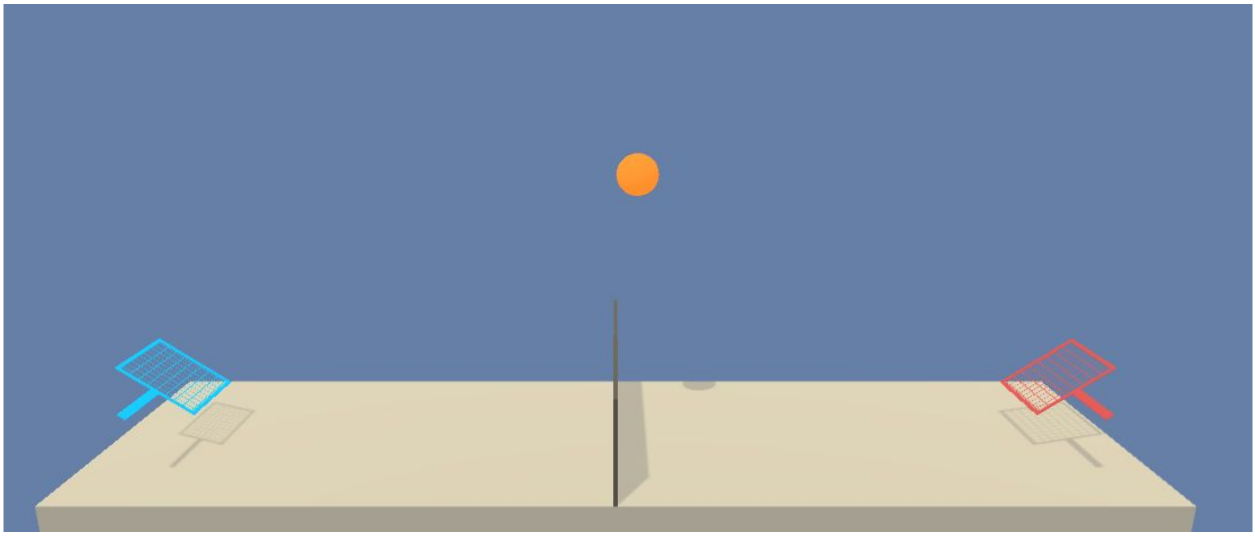
Tennis DDPG



Overview

This project is an assignment from the Udacity Deep Reinforcement Learning course. [p3_collab-compet] found here. The objective is to train the tennis agents to keep the ball in play for as long as possible. The environment consists of a state space of 24 (continues) and action space of 2 (continues). There are two agents which are acting together to keep the ball in play. Of note is the fact that the state space of each agent is relative to said agent and not the world. The environment is considered solved if the agents reach an average score of +0.5 over 100 episodes. The score is measured by taking the highest score over the two agents. To solve the environment the DDPG algorithm was used.

Implementation

1. agent.py
Contains the agent that is trained, it's learning code and the experience replay.

2. model.py
Contains the pytorch models wich are trained in Agent.py

3. Training notebook.ipynb
Contains the code to train an agent and save its model

4. Evaluation notebook.ipynb
Contains code to load the model and evaluate its performance

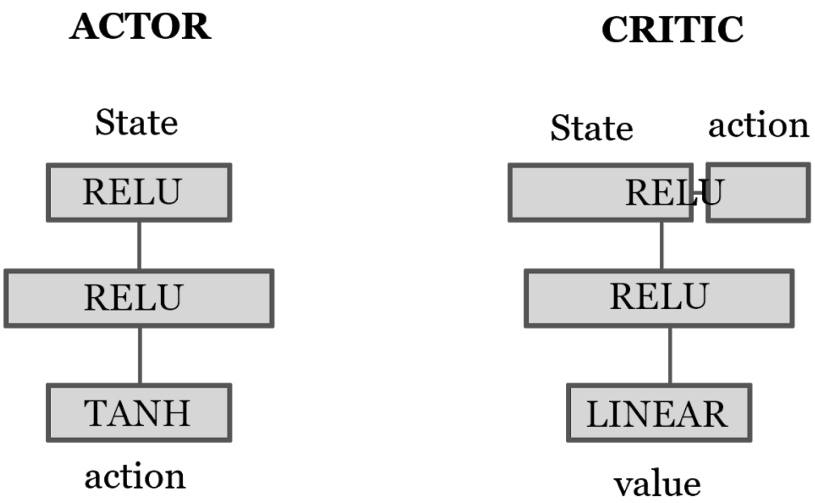
Learning algorithm and model architecture

The learning algorithm used is DDPG. Which trains two networks, one actor and one critic. Where the actor network learns an estimated optimal action to take by estimating an optimizer over the value function. And the critic which tries to estimate the value function over the current estimated best action.

Due to the states being relative to each agent the replay data for both agents are simply added to the buffer each timestep. Learning is done every step and uses a replay buffer and soft update to a target model.

The hyper parameters used are:

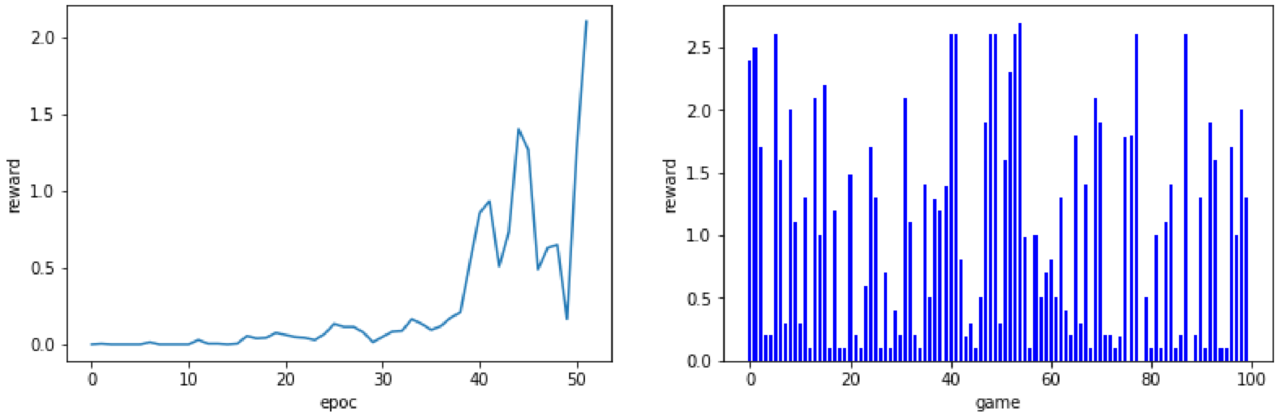
```
ACTOR_LEARNING_RATE = 1e-4
CRITIC_LEARNING_RATE = 1e-3
DISCOUNT_RATE = 0.99
TAU = 0.01
```



Performance

Training is done for 100 epochs or untill an average score of 1.5 is reached, where each epoc plays 20 games. The model reaches an average score of over 0.5 at 40 epocs and an average score of 1.5 at 50. Of note is the fact that the model is not stable and may achieve much worse performance if trained for the full 100 epocs alone. The average evaluation score is: **1.02**

1. **2.**



(1. Training performance, 2. Testing performance)

Improvements

While in a static environment the information in the replay buffer remains relevant to the training by representing the underlying state to state transition probabilities. But due to the opponent agent having a direct effect on how the environment behaves the underlying environment changes with the training of the agent. As such older replays may become inaccurate and hinder or derail training. To prevent this some kind of discounting may be helpful to lessen the weight these older replays have over time.