UDACITY

Collaboration and Competition - Report

By Taimur Zahid

Model Architecture and Algorithm: For this project, the Multi-Agent Deep Deterministic Policy Gradients (MADDPG) algorithm was used. The algorithm consists of two deep neural networks, one for the Actor and one for the Critic. The actor is used to approximate the optimal policy deterministically, i.e it outputs the best believed action for any given state. The critic learns to evaluate the optimal action-value function by using the actor's best believed action. Each agent receives its own, local observation and we use it to simultaneously train both agents through self-play. Each agent uses the same actor network to select actions, and the experiences were added to a shared replay buffer.

Neural Network - Actor

```
fcs1_units=256
fc2_units=128
# state_size = 24 for each Agent
self.fc1 = nn.Linear((state_size * 2), fcs1_units)
self.fc2 = nn.Linear(fcs1_units, fc2_units)
self.fc3 = nn.Linear(fc2_units, action_size)
```

Neural Network - Critic

```
fcs1_units=256
fc2_units=128
# state_size = 24 for each Agent
self.fc1 = nn.Linear((state_size * 2), fcs1_units)
self.fc2 = nn.Linear(fcs1_units + (action_size * 2), fc2_units)
self.fc3 = nn.Linear(fc2_units, 1)
```

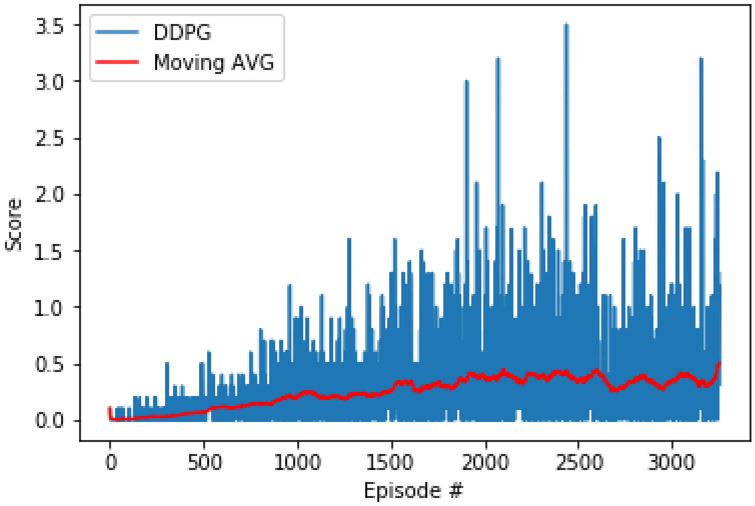
Hyperparameters: The Values for the Hyperparameters are as follows:

```
BUFFER_SIZE = int(1e6)
BATCH_SIZE = 128
GAMMA = 0.99
TAU = 8e-3
LR_ACTOR = 1e-3
LR_CRITIC = 1e-3
WEIGHT_DECAY = 0
LEARN_EVERY = 1
LEARN_NUM = 1
GRAD_CLIPPING = 1.0
OU_SIGMA = 0.2
OU_THETA = 0.15
EPSILON = 5.0
EPSILON_DECAY = 6e-6
```

Training Outputs and Plots: The Training output along with the graph are as follows:

- Episode 1 (0s)   Moving Avg: 0.1000   Best Score: 0.1000
- Episode 100 (0s)   Moving Avg: 0.0070   Best Score: 0.1000
- Episode 200 (1s)   Moving Avg: 0.0250   Best Score: 0.2000
- Episode 300 (0s)   Moving Avg: 0.0290   Best Score: 0.2000
- Episode 400 (0s)   Moving Avg: 0.0539   Best Score: 0.5000
- Episode 500 (1s)   Moving Avg: 0.0700   Best Score: 0.5000
- Episode 600 (0s)   Moving Avg: 0.1139   Best Score: 0.6000
- Episode 700 (2s)   Moving Avg: 0.1160   Best Score: 0.6000
- Episode 800 (1s)   Moving Avg: 0.1420   Best Score: 0.6000
- Episode 900 (1s)   Moving Avg: 0.1799   Best Score: 0.8000
- Episode 1000 (4s)   Moving Avg: 0.1999   Best Score: 1.1900
- Episode 1100 (1s)   Moving Avg: 0.2240   Best Score: 1.1900
- Episode 1200 (3s)   Moving Avg: 0.2079   Best Score: 1.1900
- Episode 1300 (2s)   Moving Avg: 0.2250   Best Score: 1.6000
- Episode 1400 (1s)   Moving Avg: 0.2178   Best Score: 1.6000
- Episode 1500 (2s)   Moving Avg: 0.2629   Best Score: 1.6000
- Episode 1600 (0s)   Moving Avg: 0.3219   Best Score: 1.6000
- Episode 1700 (1s)   Moving Avg: 0.2928   Best Score: 1.6000
- Episode 1800 (4s)   Moving Avg: 0.2939   Best Score: 1.6000
- Episode 1900 (6s)   Moving Avg: 0.3508   Best Score: 1.6000
- Episode 2000 (0s)   Moving Avg: 0.3819   Best Score: 3.0000
- Episode 2100 (1s)   Moving Avg: 0.4420   Best Score: 3.2000
- Episode 2200 (5s)   Moving Avg: 0.3276   Best Score: 3.2000
- Episode 2300 (2s)   Moving Avg: 0.3786   Best Score: 3.2000
- Episode 2400 (1s)   Moving Avg: 0.4227   Best Score: 3.2000
- Episode 2500 (1s)   Moving Avg: 0.3370   Best Score: 3.5000
- Episode 2600 (1s)   Moving Avg: 0.4390   Best Score: 3.5000
- Episode 2700 (1s)   Moving Avg: 0.2620   Best Score: 3.5000
- Episode 2800 (4s)   Moving Avg: 0.3340   Best Score: 3.5000
- Episode 2900 (1s)   Moving Avg: 0.3409   Best Score: 3.5000
- Episode 3000 (0s)   Moving Avg: 0.3399   Best Score: 3.5000
- Episode 3100 (2s)   Moving Avg: 0.3927   Best Score: 3.5000
- Episode 3200 (5s)   Moving Avg: 0.3188   Best Score: 3.5000
- Episode 3253 (7s)   Moving Avg: 0.5038   Best Score: 3.5000

```
Environment solved in 3153 episodes! Average Score: 0.50
```



Future Improvements:

- Proximal Policy Optimization
- Prioritized Experience Replay