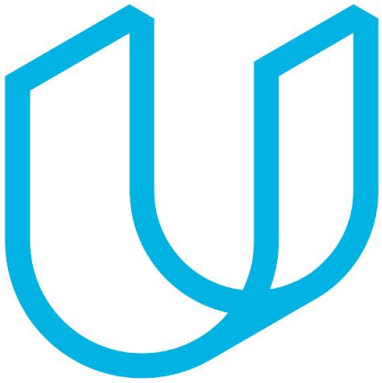
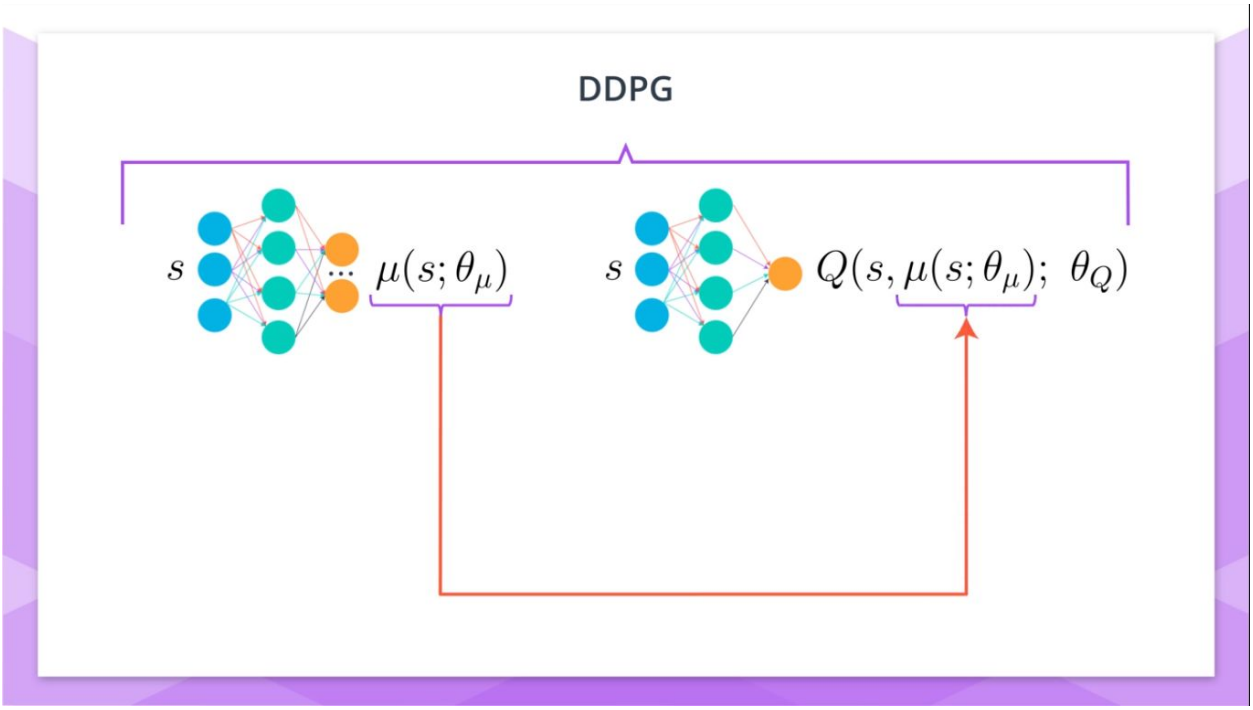


UDACITY

Continuous Control - Report

By Taimur Zahid

Model Architecture and Algorithm: For this project, the Deep Deterministic Policy Gradients (DDPG) algorithm was used. The following image is a screenshot taken from one of the lessons of the Deep Reinforcement Learning Nanodegree. 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.



Neural Network - Actor

```
self.fc1 = nn.Linear(state_size, fcs1_units)
self.bn1 = nn.BatchNorm1d(fcs1_units)
self.fc2 = nn.Linear(fcs1_units, fc2_units)
self.fc3 = nn.Linear(fc2_units, action_size)
```

Neural Network - Critic

```
self.fc1 = nn.Linear(state_size, fcs1_units)
self.bn1 = nn.BatchNorm1d(fcs1_units)
self.fc2 = nn.Linear(fcs1_units+action_size, fc2_units)
self.fc3 = nn.Linear(fc2_units, 1)
```

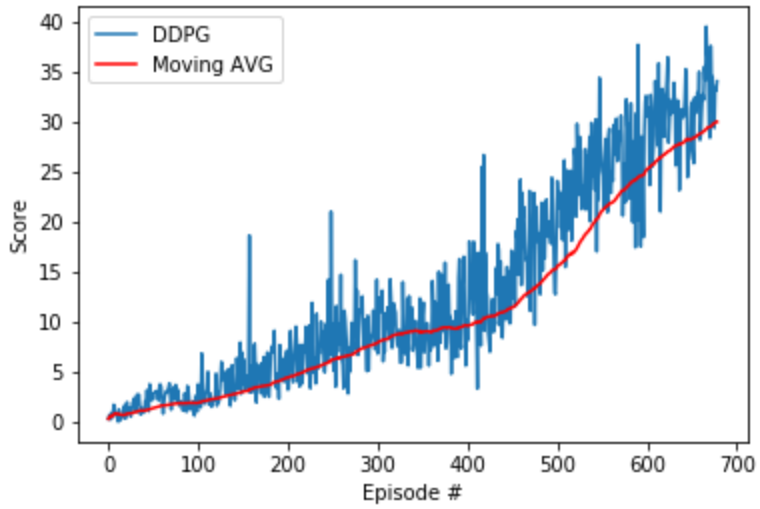
Hyperparameters: The Values for the Hyperparameters are as follows:

```
BUFFER_SIZE = int(1e6) # replay buffer size
BATCH_SIZE = 128 # minibatch size
GAMMA = 0.99 # discount factor
TAU = 1e-3 # for soft update of target parameters
LR_ACTOR = 1e-3 # learning rate of the actor
LR_CRITIC = 1e-3 # learning rate of the critic
WEIGHT_DECAY = 0 # L2 weight decay
LEARN_EVERY = 20 # learning timestep interval
LEARN_NUM = 10 # number of learning passes
GRAD_CLIPPING = 1.0 # Gradient Clipping
OU_SIGMA = 0.15 # Ornstein-Uhlenbeck noise parameters
OU_THETA = 0.05 # Ornstein-Uhlenbeck noise parameters
EPSILON = 1.0 # for epsilon in the noise process (act step)
EPSILON_DECAY = 1e-6 # For the decay if the epsilon over time
```

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

- Episode 100 (9s) Mean: 1.0 Moving Avg: 1.9
- Episode 200 (9s) Mean: 7.1 Moving Avg: 4.4
- Episode 300 (9s) Mean: 14.2 Moving Avg: 7.9
- Episode 400 (10s) Mean: 10.2 Moving Avg: 9.6
- Episode 500 (10s) Mean: 17.6 Moving Avg: 15.4
- Episode 600 (11s) Mean: 32.6 Moving Avg: 25.2
- Episode 679 (12s) Mean: 34.1 Moving Avg: 30.0

Environment solved in 579 episodes! Average Score: 30.04



Future Improvements: The following algorithms can be used to train a better model:

1. A3C - Asynchronous Advantage Actor Critic algorithm
2. A2C - Advantage Actor Critic algorithm
3. GAE - Generalized Advantage Estimation