Project Report

Learning algorithm

The learning algorithm used is vanilla Deep Deterministic Policy Gradient Q Learning as described in this paper https://arxiv.org/abs/1509.02971.

The actor network has following layers:

- Fully connected layer - input: 33 (state size) output: 256
- Fully connected layer - input: 256 output 128
- Fully connected layer - input: 128 output: 4 (action size)

The critic network has following layers:

- Fully connected layer - input: 33 (state size) output: 256
- Fully connected layer - input: 260 output 128
- Fully connected layer - input: 128 output: 1

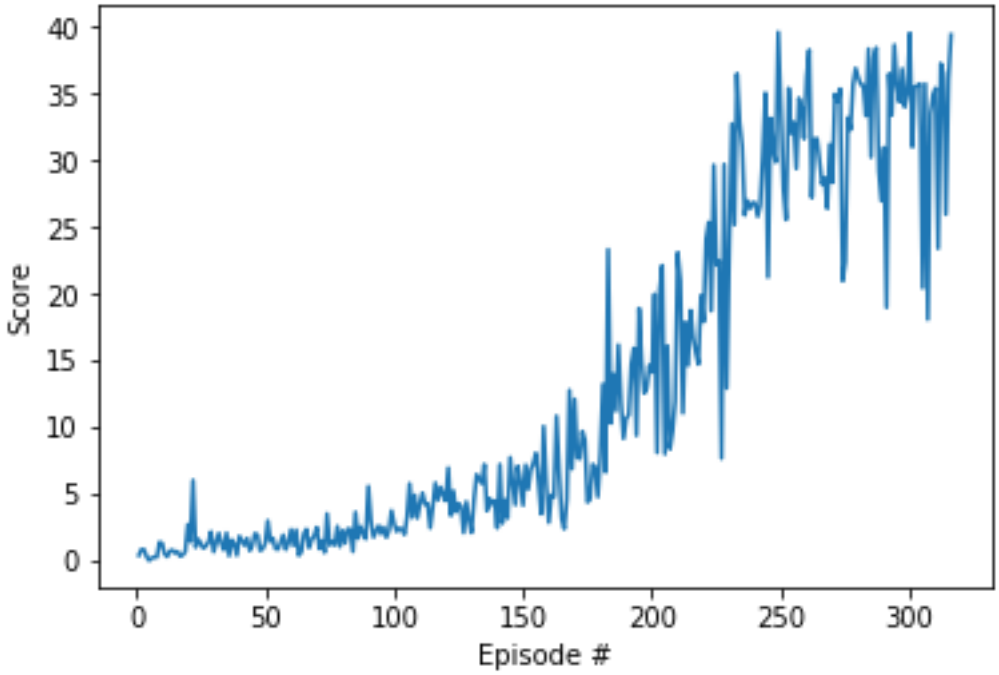
Parameters used for the DDPG agent:

- BUFFER_SIZE = int(1e5) # 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-4 # learning rate of the actor
- LR_CRITIC = 1e-4 # learning rate of the critic
- WEIGHT_DECAY = 0 # L2 weight decay

Results :

```
Episode 0      Average Score: 0.39
Episode 50     Average Score: 1.17
Episode 100    Average Score: 1.54
Episode 150    Average Score: 3.20
Episode 200    Average Score: 7.06
Episode 250    Average Score: 16.19
Episode 300    Average Score: 27.66
```

```
Environment solved in 315 episodes!  Average Score: 30.22
```



Future work:

- Better hyperparameter tuning
- Solving version 2 of the problem zith 20 simultaneous agents
- Try other algorithms like REINFORCE, TNPG, RWR, REPS, TRPO, CEM, CMA-ES and cmp are them tp DDPG