Continues Control (Single Agent)

I used DDPG algorithm to train the agent so agent can select best action against each state.

Why DDPG:

DDPG is an actor-critic, model-free algorithm based on the deterministic policy gradient that can operate over continuous action spaces.

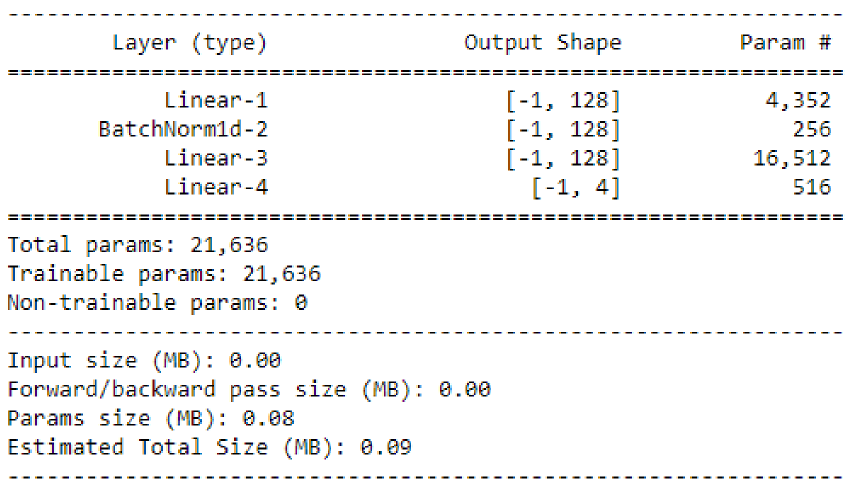
Reference Used:

1. https://arxiv.org/abs/1509.02971
2. https://arxiv.org/pdf/1509.02971.pdf

Summary of DDPG Network:

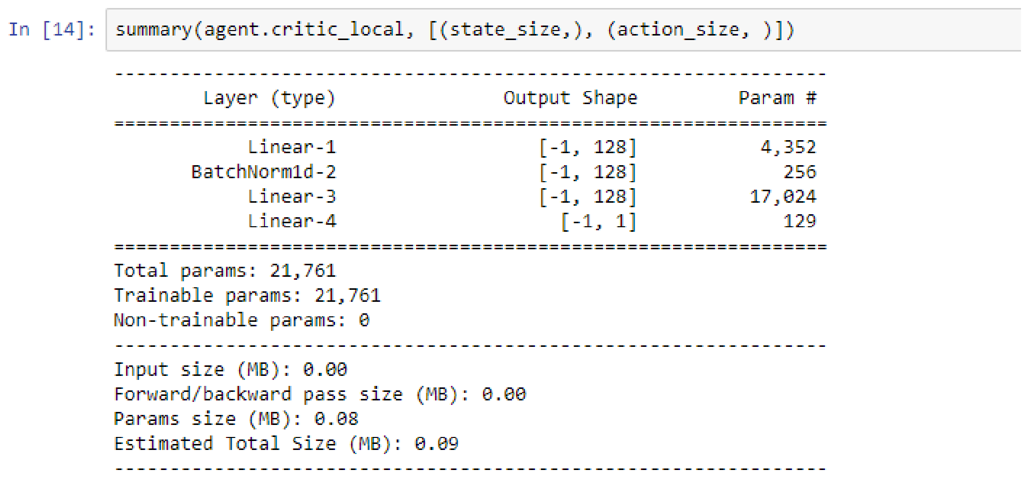
Used an Actor and Critic Network. As per the guidance of paper use Batch normalization as well. I used Adam optimizer to train the network. Fine tune the hyper parameters than mentioned in paper to converge faster.

- **Summary of Actor Network:**



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                 [-1, 128]           4,352
       BatchNorm1d-2                 [-1, 128]             256
            Linear-3                 [-1, 128]          16,512
            Linear-4                   [-1, 4]             516
================================================================
Total params: 21,636
Trainable params: 21,636
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.08
Estimated Total Size (MB): 0.09
----------------------------------------------------------------
```

- **Summary of Critic Network:**



```
In [14]: summary(agent.critic_local, [(state_size,), (action_size, )])
```

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                 [-1, 128]           4,352
       BatchNorm1d-2                 [-1, 128]             256
            Linear-3                 [-1, 128]          17,024
            Linear-4                   [-1, 1]             129
================================================================
Total params: 21,761
Trainable params: 21,761
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.08
Estimated Total Size (MB): 0.09
----------------------------------------------------------------
```

Summary Hyper parameter:

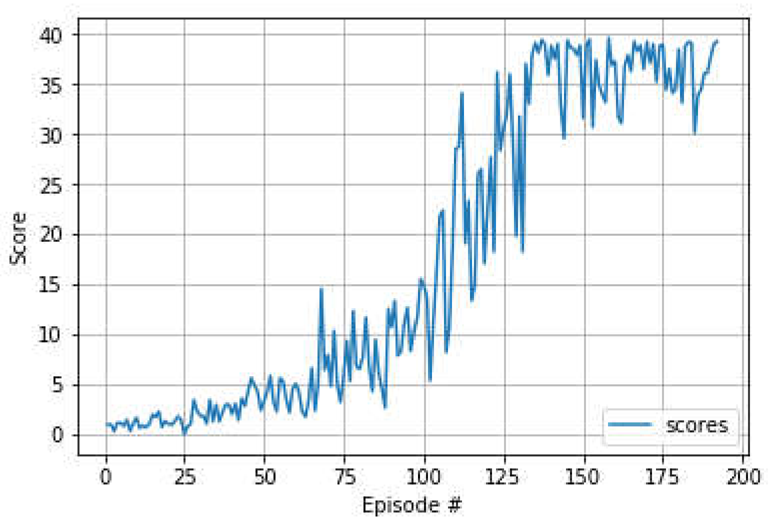
- **Maximum steps per episode:** 1000
- **Replay Buffer size:** 100000
- **Batch Size:** 128
- **Gamma:** 0.99
- **Tau:** 0.001
- **Actor Learning Rate:** 0.0002
- **Critic Learning Rate:** 0.0002
- **Weight Decay**: 0.0

Rewards Function Performance During Training:

- **Average Score after every 100 episodes:**

```
Episode 100     Average Score: 4.51
Episode 192     Average Score: 30.19
Environment solved in 92 episodes!      Average Score: 30.19
```

- **Plot shows average rewards against each episode:**



Rewards Function Performance During Prediction:

```
Episode 98      Average Score: 35.21
Episode 99      Average Score: 35.18
Episode 100     Average Score: 35.21
Average Score Of 100 Consecutive Episodes: 35.13582271759092
```

Networks want to try in future:

1. D4PG for multiagent.
2. Proximal Policy Optimization.