Tennis Game (Multi 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. Use the same network as used in previous project but unlikely previous project it is used to train multiple agents.

- **Summary of Actor Network:**

```
In [12]: from torchsummary import summary
         summary(agent.actor_local, (state_size,))

C:\ProgramData\Anaconda2\envs\udacity\lib\site-packages\torch\nn\functional.
ted. Use torch.tanh instead.
  warnings.warn("nn.functional.tanh is deprecated. Use torch.tanh instead.")

----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                [-1, 128]           3,200
       BatchNorm1d-2                [-1, 128]             256
            Linear-3                [-1, 128]          16,512
            Linear-4                  [-1, 2]             258
================================================================
Total params: 20,226
Trainable params: 20,226
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): 0.08
----------------------------------------------------------------
```

- **Summary of Critic Network:**

```
In [13]: summary(agent.critic_local, [(state_size,), (action_size, )])

----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Linear-1                [-1, 128]           3,200
       BatchNorm1d-2                [-1, 128]             256
            Linear-3                [-1, 128]          16,768
            Linear-4                  [-1, 1]             129
================================================================
Total params: 20,353
Trainable params: 20,353
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): 0.08
----------------------------------------------------------------
```

Summary Hyper parameter:

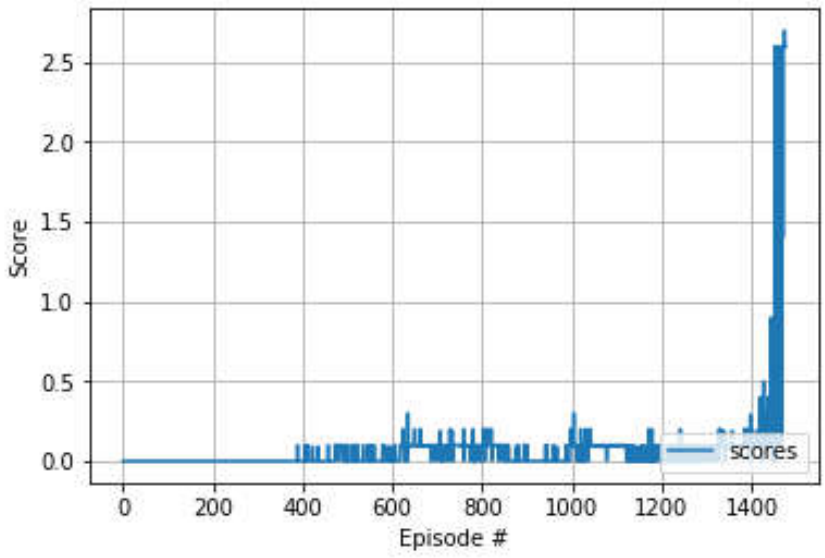
- **Maximum steps per episode:** 1000
- **Replay Buffer size:** 100000
- **Batch Size:** 128
- **Gamma:** 0.99
- **Tau:** 0.001
- **Actor Learning Rate:** 0.0002
- **Critic Learning Rate:** 0.0002
- **Weight Decay**: 0.0

Rewards Function Performance During Training:

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

```
Episode 100     Average Score: 0.0000
Episode 200     Average Score: 0.0000
Episode 300     Average Score: 0.0000
Episode 400     Average Score: 0.0010
Episode 500     Average Score: 0.0260
Episode 600     Average Score: 0.0194
Episode 700     Average Score: 0.0811
Episode 800     Average Score: 0.0882
Episode 900     Average Score: 0.0624
Episode 1000    Average Score: 0.0099
Episode 1100    Average Score: 0.0890
Episode 1200    Average Score: 0.0524
Episode 1300    Average Score: 0.0340
Episode 1400    Average Score: 0.0973
Episode 1475    Average Score: 0.5219
Environment solved in 1475 episodes!    Average Score: 0.5219
```

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



Rewards Function Performance During Prediction:

```
Episode 92      Average Score: 2.13
Episode 93      Average Score: 2.11
Episode 94      Average Score: 2.11
Episode 95      Average Score: 2.12
Episode 96      Average Score: 2.10
Episode 97      Average Score: 2.10
Episode 98      Average Score: 2.11
Episode 99      Average Score: 2.11
Episode 100     Average Score: 2.12
Average Score Of 100 Consecutive Episodes: 2.1279026303537676
```

Networks want to try in future:

1. I want to try MADDPG.