P3- Collab-Compet: Tennis

For solving this environment, I used DDPG algorithm for training neural network agent for both player and predicting action against each different state.

Learning from this project

Input of the DDPG is game state vector from both Unity Agents and output of the DDPG is action space for different state

a. Key Point about DDPG

- **Actor**
  1. Fully-connected layer - input: 33 (state size) output: 128
     - Linear Neural Network with followed by ReLU activation function
  2. Batch Normalization
     - To stabilize neural network, we used this. Please find more detail about it from here
  3. Hidden Fully-connected layer - input: 128 output: 128
     - Linear Neural Network with followed by ReLU activation function
  4. Fully-connected layer - input: 128 output: 4 (action size)
     - Linear Neural Network with followed by Tanh activation function
  5. Maximum steps per episode: 1000
  6. Update neural network after each step and batch size of 128
  7. Discount factor: 0.99

- **Critic**
  1. Fully-connected layer - input: 33 (state size) output: 128
     - Linear Neural Network with followed by ReLU activation function
  2. Batch Normalization
     - To stabilize neural network, we used this. Please find more detail about it from here
  3. Hidden Fully-connected layer - input: 128 output: 128
     - Linear Neural Network with followed by ReLU activation function
  4. Fully-connected layer - input: 128 output: 4 (action size)
     - Linear Neural Network with followed by Tanh activation function
  5. Maximum steps per episode: 1000
  6. Update neural network after each step and batch size of 128
  7. Discount factor: 0.99

- **Added Noise** to modify action against state to add some twist.
- **Replay Buffer Size:** 1e5
- **Why Batch Normalization is used?**
  - To reduce impact of very large weight or outliner data on next level neural network and Stabilize neural network we used this. Find more details here
- **Why ReLU is used at hidden layer?**
  - The ReLU is the most used activation function in the world right now. Since, it is used in almost all the convolutional neural networks or deep learning.
  - Relu has range between 0 to infinity
- **Why Tanh is used at last layer?**
  - tanh is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped).
  - tanh and logistic sigmoid activation functions are used in feed-forward nets.
  - The tanh function is mainly used classification between two classes.

Plot of Rewards

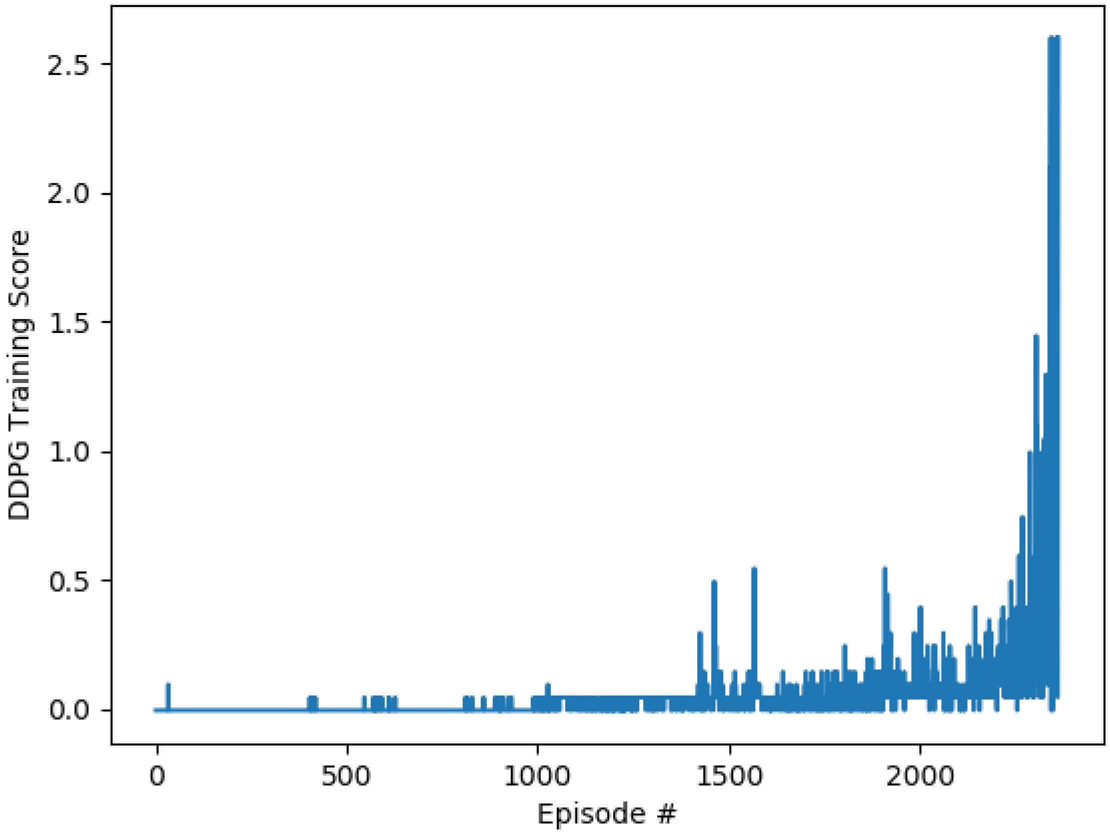
Plotting Average Score against Episode of the game.

DDPG

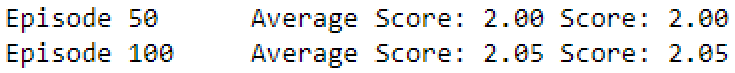
1. Training Phase

```
2. Episode 50      Average Score: -0.00 Score: -0.00
3. Episode 100     Average Score: -0.00 Score: -0.00
4. Episode 150     Average Score: -0.00 Score: -0.00
5. Episode 200     Average Score: -0.00 Score: -0.00
6. Episode 250     Average Score: -0.00 Score: -0.00
7. Episode 300     Average Score: -0.00 Score: -0.00
8. Episode 350     Average Score: -0.00 Score: -0.00
9. Episode 400     Average Score: -0.00 Score: -0.00
10.  Episode 450   Average Score: -0.00 Score: -0.00
11.  Episode 500   Average Score: -0.00 Score: -0.00
12.  Episode 550   Average Score: -0.00 Score: -0.00
13.  Episode 600   Average Score: 0.00e Score: 0.000
14.  Episode 650   Average Score: 0.00e Score: 0.00
15.  Episode 700   Average Score: -0.00 Score: -0.00
16.  Episode 750   Average Score: -0.00 Score: -0.00
17.  Episode 800   Average Score: -0.00 Score: -0.00
18.  Episode 850   Average Score: -0.00 Score: -0.00
19.  Episode 900   Average Score: 0.00e Score: 0.000
20.  Episode 950   Average Score: 0.01e Score: 0.01
21.  Episode 1000  Average Score: 0.00e Score: 0.00
22.  Episode 1050  Average Score: 0.01e Score: 0.01
23.  Episode 1100  Average Score: 0.03 Score: 0.033
24.  Episode 1150  Average Score: 0.02e Score: 0.02
25.  Episode 1200  Average Score: 0.01 Score: 0.011
26.  Episode 1250  Average Score: 0.02e Score: 0.02
27.  Episode 1300  Average Score: 0.03 Score: 0.033
28.  Episode 1350  Average Score: 0.04 Score: 0.043
29.  Episode 1400  Average Score: 0.04 Score: 0.044
30.  Episode 1450  Average Score: 0.04 Score: 0.044
31.  Episode 1500  Average Score: 0.05 Score: 0.055
```

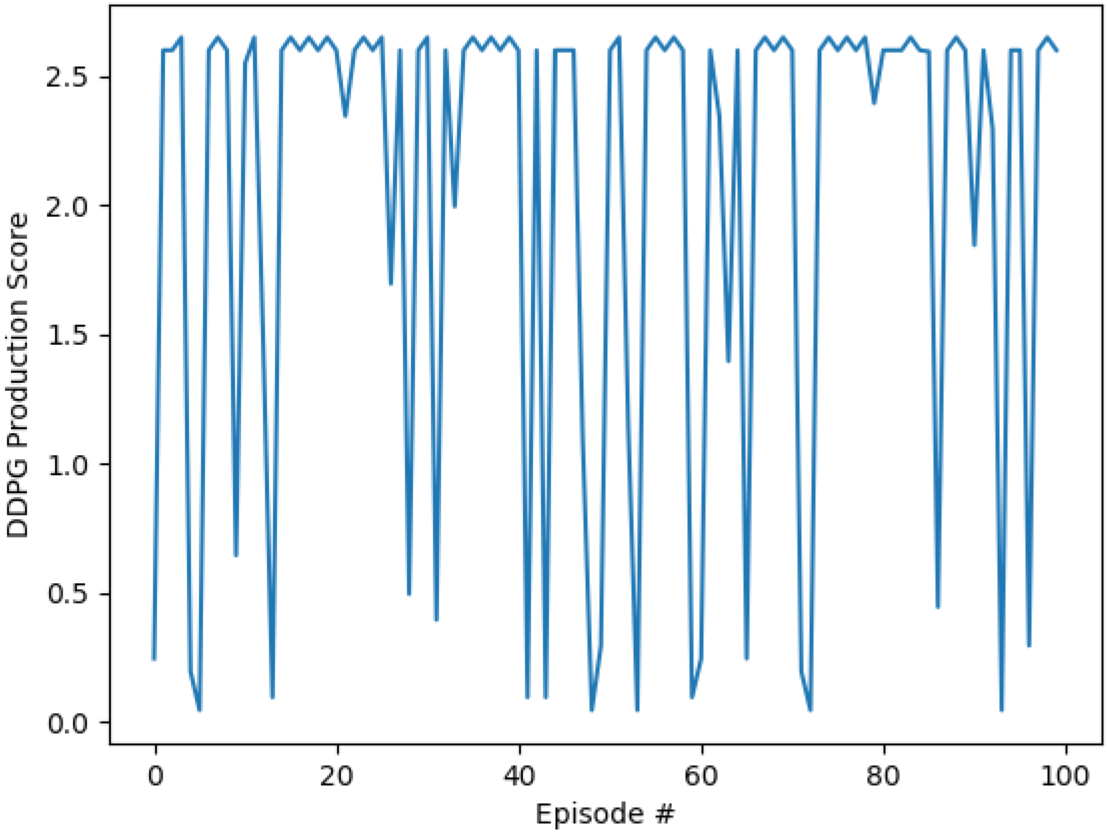
```
32. Episode 1550   Average Score: 0.05 Score: 0.055
33. Episode 1600   Average Score: 0.05 Score: 0.055
34. Episode 1650   Average Score: 0.05 Score: 0.055
35. Episode 1700   Average Score: 0.03e Score: 0.03
36. Episode 1750   Average Score: 0.04 Score: 0.044
37. Episode 1800   Average Score: 0.05e Score: 0.05
38. Episode 1850   Average Score: 0.06 Score: 0.066
39. Episode 1900   Average Score: 0.06 Score: 0.066
40. Episode 1950   Average Score: 0.08 Score: 0.087
41. Episode 2000   Average Score: 0.09 Score: 0.099
42. Episode 2050   Average Score: 0.09 Score: 0.099
43. Episode 2100   Average Score: 0.08 Score: 0.089
44. Episode 2150   Average Score: 0.07 Score: 0.076
45. Episode 2200   Average Score: 0.08 Score: 0.087
46. Episode 2250   Average Score: 0.13 Score: 0.138
47. Episode 2300   Average Score: 0.19 Score: 0.193
48. Episode 2350   Average Score: 0.37 Score: 0.375
49. Episode 2363   Score: 2.60    Average Score: 0.51
50. Environment solved in 2363 episodes! Average Score: 0.51
```



51. Production Phase



```
Episode 50      Average Score: 2.00 Score: 2.00
Episode 100     Average Score: 2.05 Score: 2.05
```



Future Implementations

1. Implementation and try another algorithm (ex. PPO)
2. Changing neural network size
3. Tune Hyper parameters for better learning
4. Attempt and solve the Football environment. this would be challenging as the environment not only collaborative between players in the same team but is also competitive between players of opposite teams.