

Project three of deep reinforcement learning from Udacity.

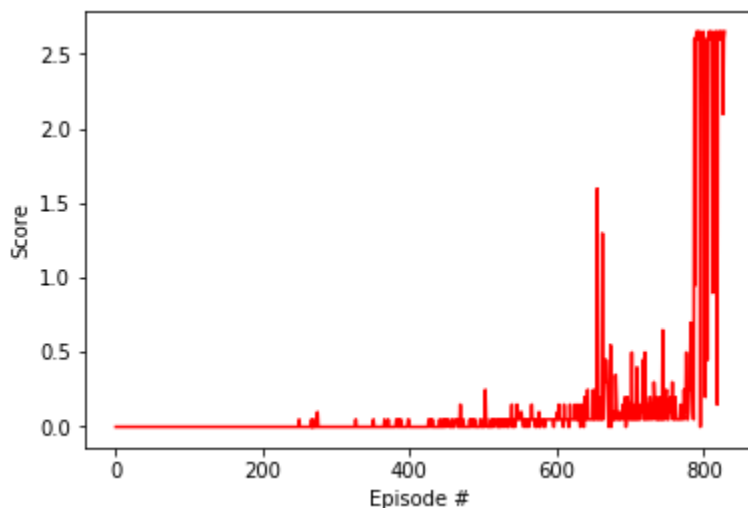
This project leverages a lot of code from the previous project Continuous control. That mean Actor Critic algorithms is being use for the Agent. The general algorithms remain the same except that Batch normalization is adding into both Actor and Critic network this time.

Since we have two agents and they need to learn together, I implemented a new function call **share_learning** within the Agent class in the `ddpg_agent.py` file. This function will force them to share the neural network model weighting for both actor and critic. In additional, I also want them to share the experience; therefore, they both use the same ReplayBuffer.

The following are the hyper-parameter that I used.

```
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 = 5e-4         # learning rate of the actor
LR_CRITIC = 5e-4        # learning rate of the critic
WEIGHT_DECAY = 0       # L2 weight decay
```

With the parameters above, the entire training took exactly at 828 episodes to complete.



Future Enhancements

I have experienced that different seek value can sometime enhance the overall performance. This is similar to the second project (Continuous Control) that we did last time. In addition, algorithms like PPO, A3C, and D4PG that use multiple (non-interacting, parallel) copies of the same agent to distribute the task of gathering experience. Another thing that we can do is try a slightly different neural network with different number of hidden layers. That may help coverage faster.