

Overview:

Aim is to get the maximum combined score, hence play as long as possible.

Approach:

I am using DDPG to solve this multi agent problem. I used all 20 workers since it made me train very fast.

Parameters:

- BATCH_SIZE = 128 # Minibatch size
- GAMMA = 0.99 # Discount factor
- TAU = 1e-3 # For soft update of target parameters
- LR_ACTOR = 2e-4 # Actor learning rate
- LR_CRITIC = 2e-4 # Critic learning rate
- WEIGHT_DECAY = 0 # L2 weight decay

I use one linear layer then a batch normalization layer, and then again two linear layers. Used a batch normalization to remove noise and avoid overfitting.

My Policy network is designed like this:

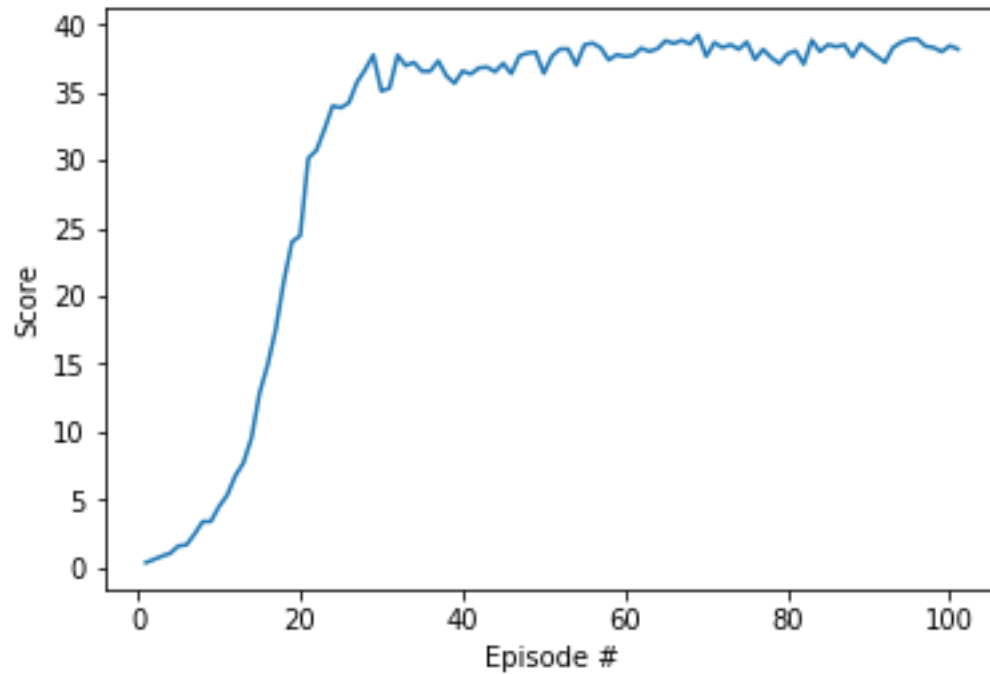
- `self.fc1 = nn.Linear(state_size, fc1_units)`
- `self.bn1 = nn.BatchNorm1d(fc1_units)`
- `self.fc2 = nn.Linear(fc1_units, fc2_units)`
- `self.fc3 = nn.Linear(fc2_units, action_size)`

Hope this explains the network.

Training Graphs

Since we use a 20 worker model, the model trains in less number of overall steps but each step take a while to run.

Model was fully trained when average score was above 31 .



Idea for Future Improvement

I would want to improve the DDPG algorithm by adding Priority experience replay (PER) in which important or relevant experiences will be sampled more often for training.

I think based on the following theory , may plan to improve.

<https://medium.freecodecamp.org/improvements-in-deep-q-learning-dueling-double-dqn-prioritized-experience-replay-and-fixed-58b130cc5682>