

Udacity Deep Reinforcement Learning Nanodegree

Project 1: Navigation

Submitted by: Tamoghna Das

1. Network architecture and Learning Algorithm

A Deep Q-Learning algorithm has been used for this project. A replay buffer has been used for using experience replay. Also, it has two separate Q-Networks with identical architectures to implement fixed Q-targets.

Each Q-Network consists of two hidden layers and one output layer.

Q-Network Architecture:

- Inputs: 37 units (state_size).
- hidden layers
 - fc1: fully connected. 37 x 64. (ReLU)
 - fc2: fully connected. 64 x 64. (ReLU)
- Output layer: fully connected layer. 4 linear output units for 4 actions(action_size).

Optimizer used: Adam optimizer (torch.optim.Adam), LR = $5e-4$

Hyperparameters:

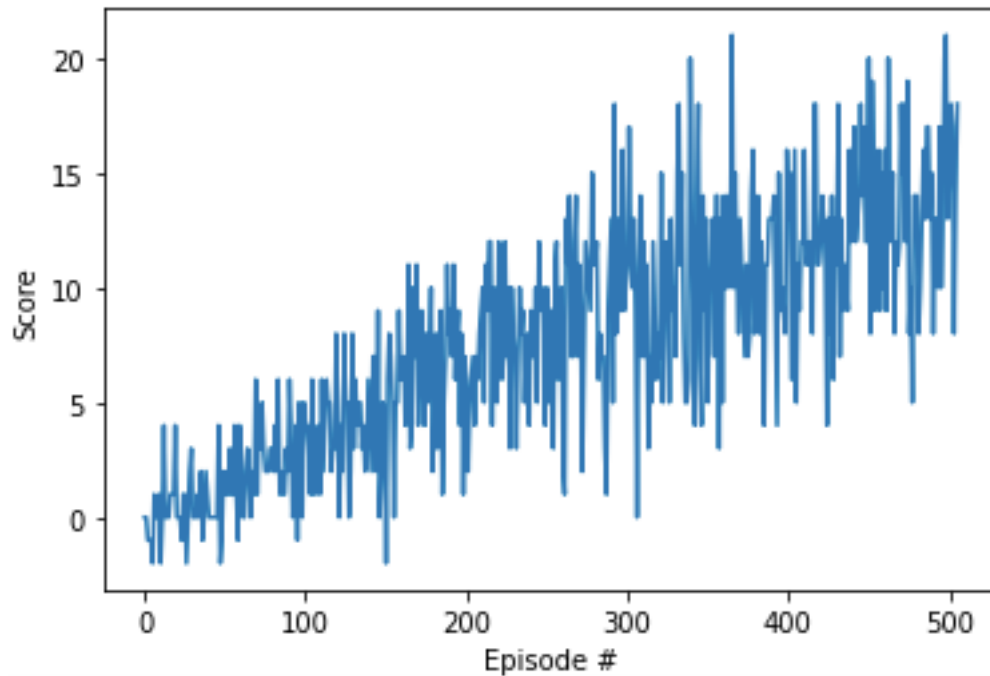
- Epsilon decay factor: 0.995. Start epsilon Value = 1. Minimum epsilon value=0.01
- Replay buffer size: 100000.
- Batch size: 64
- Target Q-Network's weight are updated every 4 actions.

2. Results

The environment is solved after 406 episodes:

```
Environment solved in 406 episodes! Average Score: 13.00
```

Plot of Scores vs episodes:



3. Ideas of Future Work

- To implement Double DQN, Prioritized Experience Replay, Dueling DQN for performance improvement.
- To implement rainbow DQN.
- To generalize the architecture for other applications/environments.