Udacity Deep Reinforcement Learning Nanodegree

Project 1: Navigation

Phan Anh Tu

I. Approaches

For this undertaking, I opted to begin with straightforward deep Q-learning to establish a baseline and gauge the effectiveness of the most basic deep learning approach. Subsequently, I introduced double DQN, and its performance was remarkably effective.

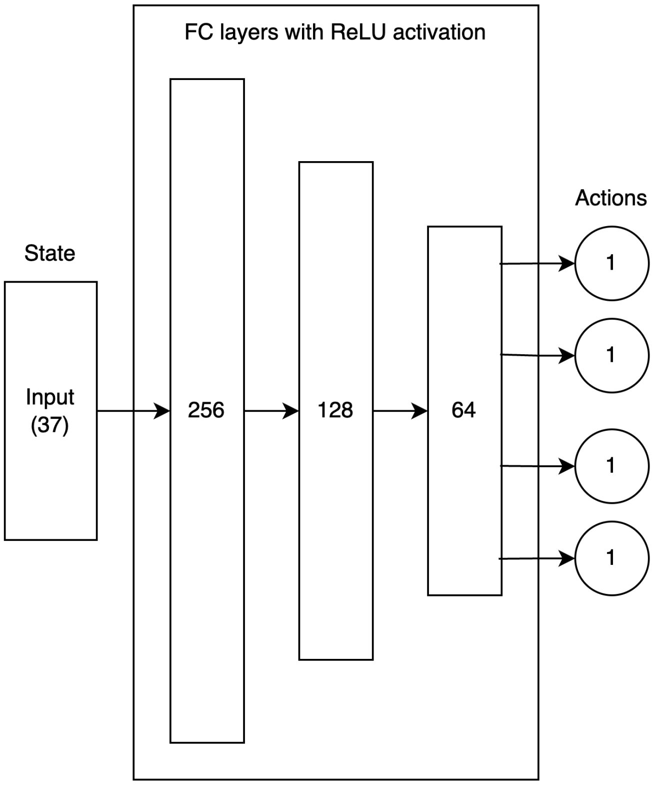
Q-Network architecture

1. Inputs: 37 units (state_size)
2. Fully-connected layers:
    a. Fc1: 256 units (ReLU)
    b. Fc2: 128 units (ReLU)
    c. Fc3: 64 units (ReLU)
3. Outputs: 4 units (linear, action_size)

Optimizer: torch.optim.Adam (LR=5-4)

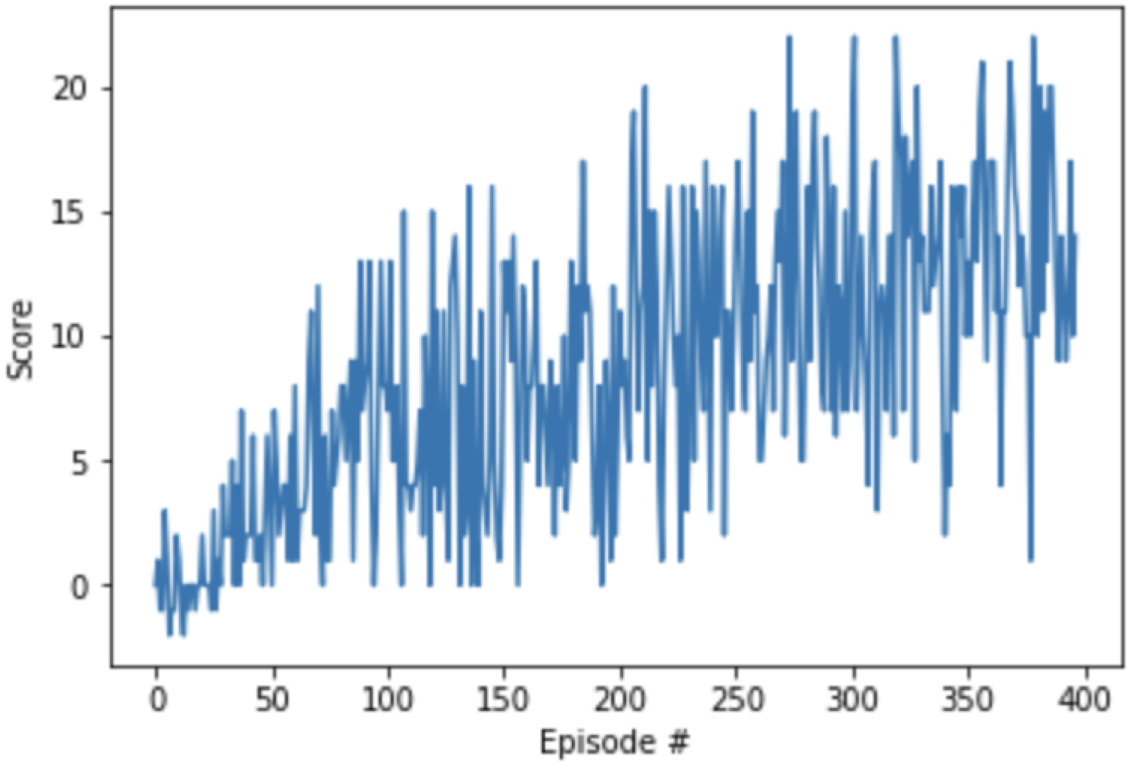
Training hyperparameters:

- Max episode length: 500
- Epsilon-greedy decay: 0.95 (start=1.0, end=0.01)
- Memory buffer size = 100000
- Batch size = 32
- Update every = 4



II. Results

Plotted reward of DQN



III. Ideas for improvement

Develop the classes more abstract to be in others environments