Project Report on “Navigation”

Learning Algorithm

I used the Deep Q Learning algorithm for this project. This algorithm was also taught in the course lessons, hence, I thought this should be given a try first. I was able to finish the project successfully with this algorithm.

Hyperparameters

I did not experiment much with hyperparameters as I the agent was able to learn well using the default hyperparameters which were provided with the sample implementation of the algorithm.

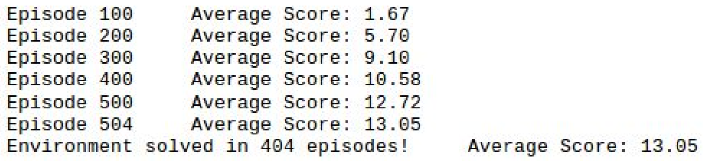
- BUFFER_SIZE = int(1e5)   # replay buffer size
- BATCH_SIZE = 64   # minibatch size
- GAMMA = 0.99   # discount factor
- TAU = 1e-3   # for soft update of target parameters
- LR = 5e-4   # learning rate
- UPDATE_EVERY = 4   # how often to update the network

Neural Network Architecture

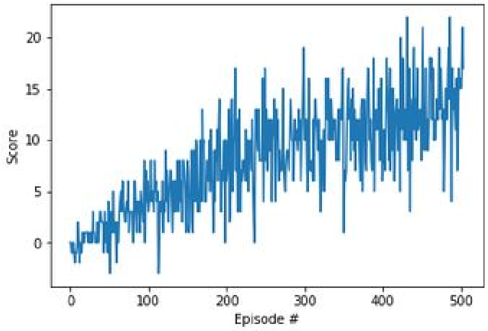
- I used 2 hidden layers for the neural network which is also same to the sample implementation provided.
- I changed the size of the layers. Layer 1 comprises of 512 units and layer 2 comprises of 64 units. Good performance was achieved with this.

Plot of Rewards

The score obtained by the network was good and satisfied the requirement of the project.



```
Episode 100     Average Score: 1.67
Episode 200     Average Score: 5.70
Episode 300     Average Score: 9.10
Episode 400     Average Score: 10.58
Episode 500     Average Score: 12.72
Episode 504     Average Score: 13.05
Environment solved in 404 episodes!     Average Score: 13.05
```



Ideas for Future Work

- As suggested in the course, I would try double DQN in future to improve the performance of the agent.
- Priority replay buffer is also an option which must be tried.