UDACITY PROJECT

Udacity p1_navigation project DQN

Prepared for: Udacity Deep Reinforcement Learning Nano Degree

Prepared by: Divyanshu Kalra

1 January 2019

SUMMARY

Algorithm

The algorithm used to solve the problem is Double DQN, this is because DDQN uses a second network which helps overcome overestimation of Q value.

Explanation:

1) DQN uses neural network to estimate the q table.
2) In DDQN decoupling the parameters being updated from the ones that are producing the target values by using a second target network.
3) Epsilon greedy policy encourages exploratory behaviour

Network

The network used to estimate q values is a feed_forward neural network with the architecture being:

Input layer: 37 nodes

Hidden layer 1: 64 nodes, activation: ReLU

Hidden layer 2: 64 nodes, activation: ReLU

Output layer: 4 nodes

Hyperparameters

Tau = 0.001

Learning Rate = 3e-4

Optimiser = Adam

Gamma = 0.95

Initial Randomness = 1

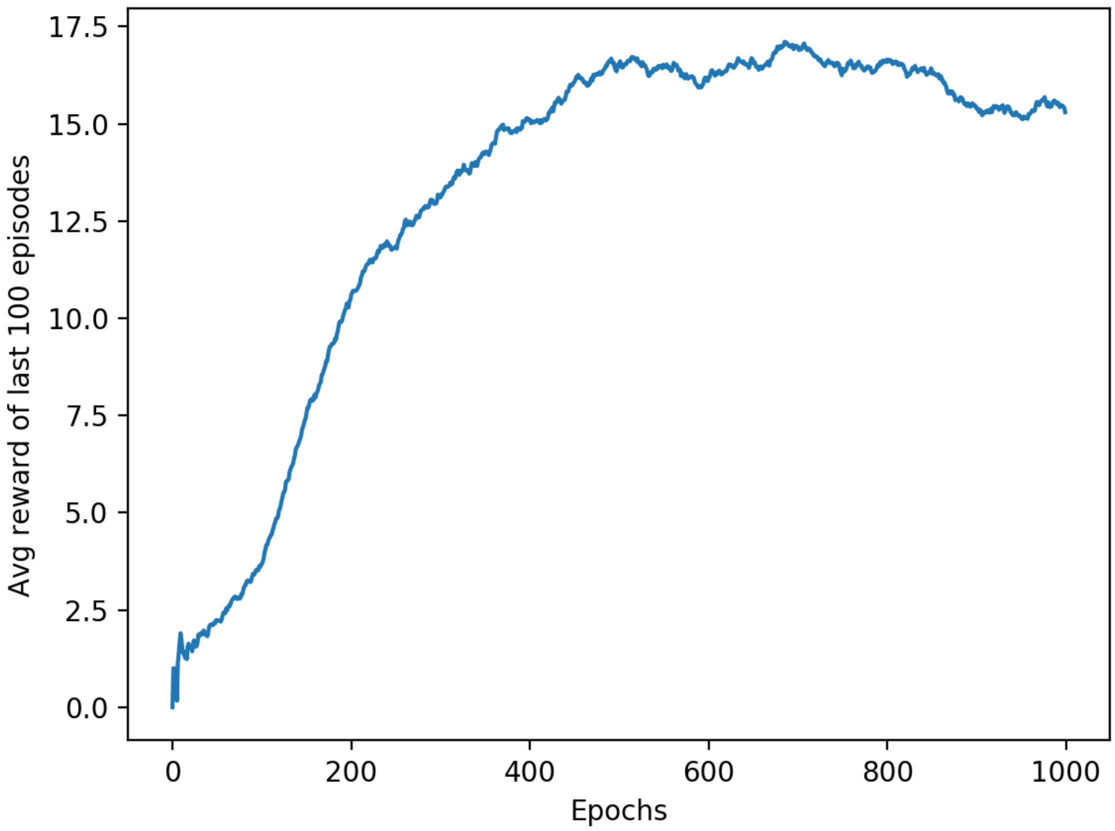
Randomness decay = 0.995

Least Randomness = 0.01

Files

- deepQAgent.py : This file contains the agent class
- network.py: Contains the neural network code
- checkpoint.pth: Contains trained weights
- train.py: used for training the agent
- watchTrained.py: used to watch the trained network interact with the environment.

AVERAGE REWARD PLOT



As it can be seen in about 300 epochs the average reward was more than +13, hence the environment was solved.

FUTURE IDEAS

1) Implementing Rainbow network to improve the agent

2) Training the agent on the images instead on state vector.