# Report- Navigation

## Learning Algorithm

About the Neural Network used:

I used a neural network with 3 hidden layers (64 nodes in each layer). ReLU activation was used after each layer, barring the last one. Adam optimizer was used for faster convergence, with a learning rate of 0.0005. State vector of size 37 is input for the neural network, and output vector (size 4) corresponds to how likely each action is. Batch size is 64. Weights of neural network are updated after every 4 time steps. We initialize 2 neural networks, one for target Q values and one for current local q estimates.

Inference: Neural network with less number of hidden layers and neurons tend to converge or learn faster for this particular problem in comparison to deeper and larger ones.

Policy:

Epsilon greedy policy was used to select an action given the current state. Value of epsilon started at 1 and was slowly decayed as the episodes progressed. (Epsilon decay 0.995)

Discount:

Rewards from later time steps are discounted by GAMMA=0.99

Experience Replay:

After each time step.. State, Action, Reward, next state and done (this is experience tuple) are always stored in a buffer. We train the neural network by taking samples (of size equal to batch size) from the buffer. By doing this our agent gets to see one experience tuple more than 1 time. And unnecessary correlations between consecutive state pairs are eliminated. The buffer size is 10^5, is a deque object.

Fixed Q targets:

We basically have 2 neural networks (NN) because, if we happen to have only 1 neural network every time we update our model weights our TD target also moves in the same direction. So inorder to keep the TD target separate from our Estimates we have a separate NN for each.

Soft update:

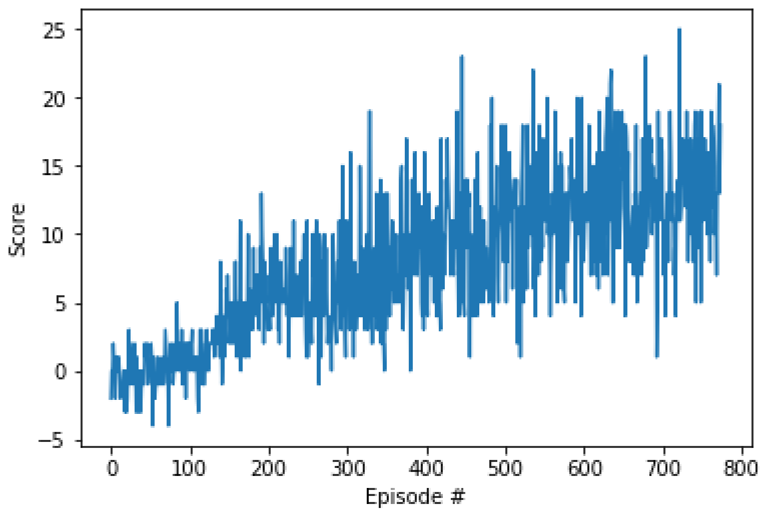
We update the target model weights using soft update. This soft update is governed by parameter Tau 0.001

<u>Double DQN:</u>

I have implemented Double DQN, which is prone to overestimation of state action values. Since I had 2 separate NN models already implementing this was easier.

<u>Plot of rewards</u>:

```
Environment solved in 674 episodes!     Average Score: 13.01
```



<u>Future Works:</u>

These techniques can be implemented to improve the performance further.

Prioritized experience learning

Dueling DQN

Distributed DQN

And of course Rainbow.