Motion Planning in Unknown Environments

Vikranth Dwaracherla, Varsha Sankar, Radhika Patil
vikranth@stanford.edu, svarsha@stanford.edu, radhikap@stanford.edu

Motivation and Introduction

Autonomous navigation in unseen environments has many applications including disaster, rescue and exploration. In this project, we developed a method to obtain a path to a desired destination in an unseen environment based on local sensor data as input. We are using Q-learning approach to make an optimal decision. Q value is obtained from function approximation using a neural network. Approach and results have been presented below.

Approach

The problem is modeled as a Q-learning problem. In our method, at each step, the robot obtains sensor data and computes it current heading and desired heading and uses these measurements to compute the optimal action, i.e, speed and angular velocity. Optimal action is obtained from maximizing the Q-value.[2]

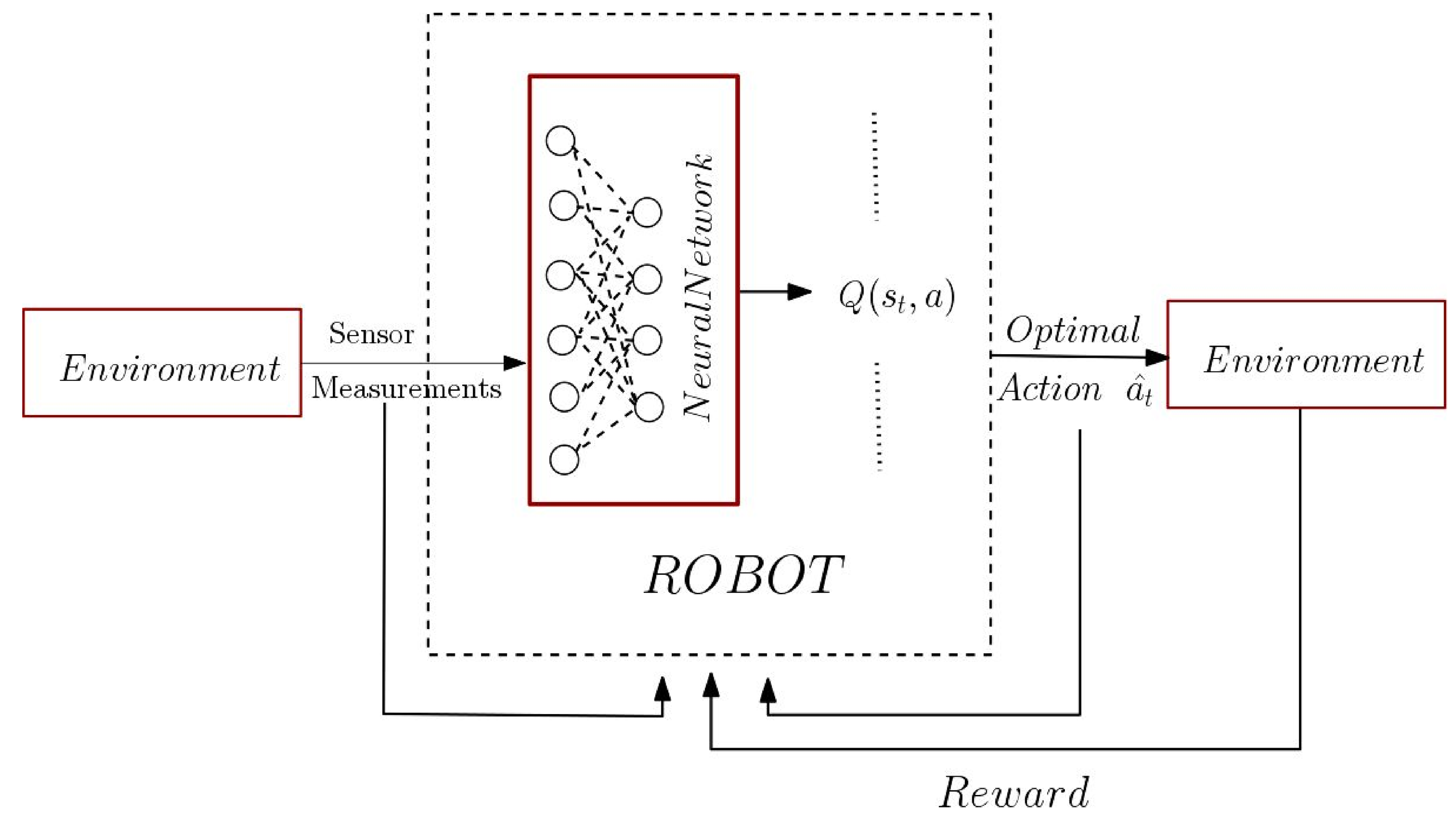
$$\hat{a}_t(\hat{v}_t, \hat{w}_t) = \arg\max_{a_t=\{v,w\}} Q(s_t, a_t)$$

Q-value is obtained from a neural network which is trained from the previous sensor data and the actions taken, with rewards from the environment approximated using the following metric.[1]

$$Reward = -\rho(\frac{d+D_n-D_c}{D_c}) - \gamma(dt - \frac{d}{Vmax}) - \beta|w|$$

Here, ρ,γ,β are hyper parameters which affect the choice of taking a locally optimal path vs choosing maximum velocity vs a straight path. d, Dc,Dc represent distance travelled in the current time step by taking the present action, shortest distance to the destination before and after taking the action.

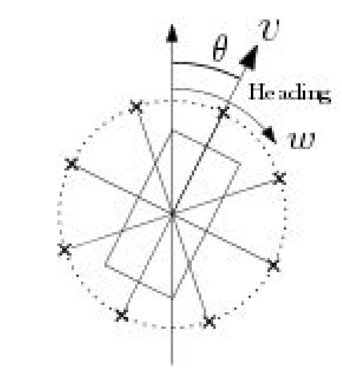
Model



Data

Initially, in order to train the neural network a warm startup procedure is implemented. In this, we generate a large amount of data in the form of (sensor data, reward) pair by randomly choosing a feasible point on the map and choosing a feasible action. This data is used to train the neural network to estimate Q values.

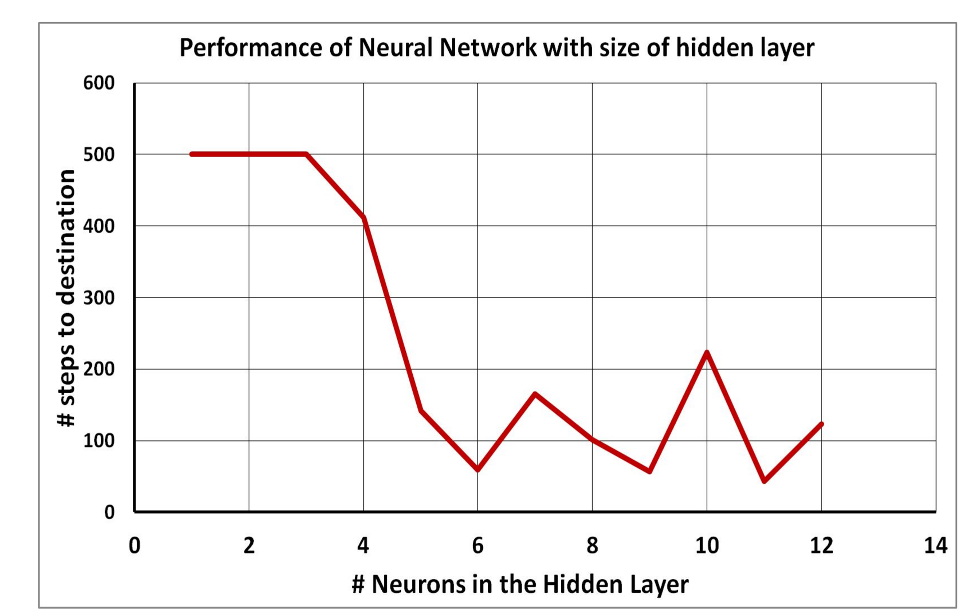
At each iteration, while running the algorithm we also compute the reward, for the state and action and this data is used to train the neural network after a fixed number of iterations. This improved our training speed drastically as,the time taken to generate by warm start procedure is much smaller than the actual method.

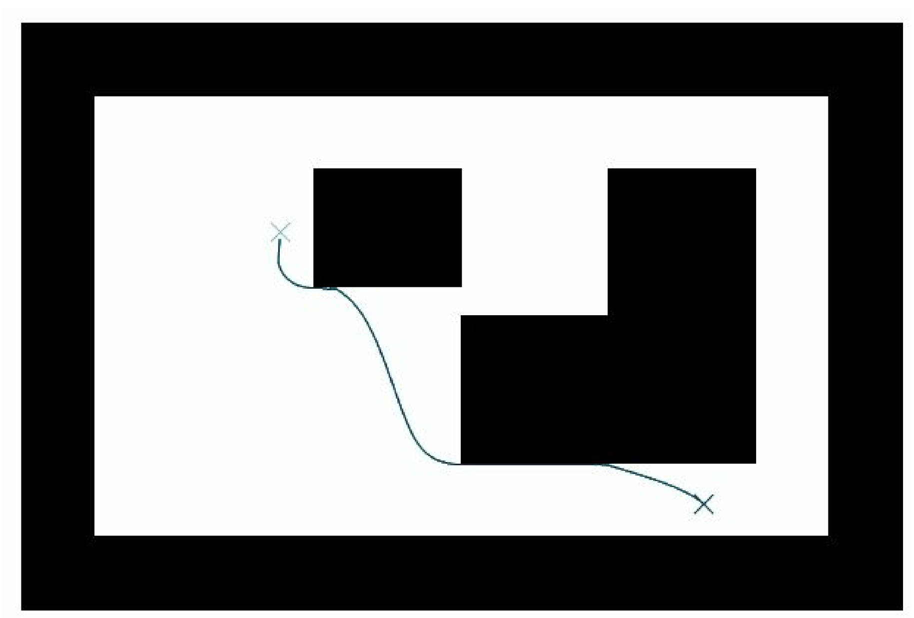


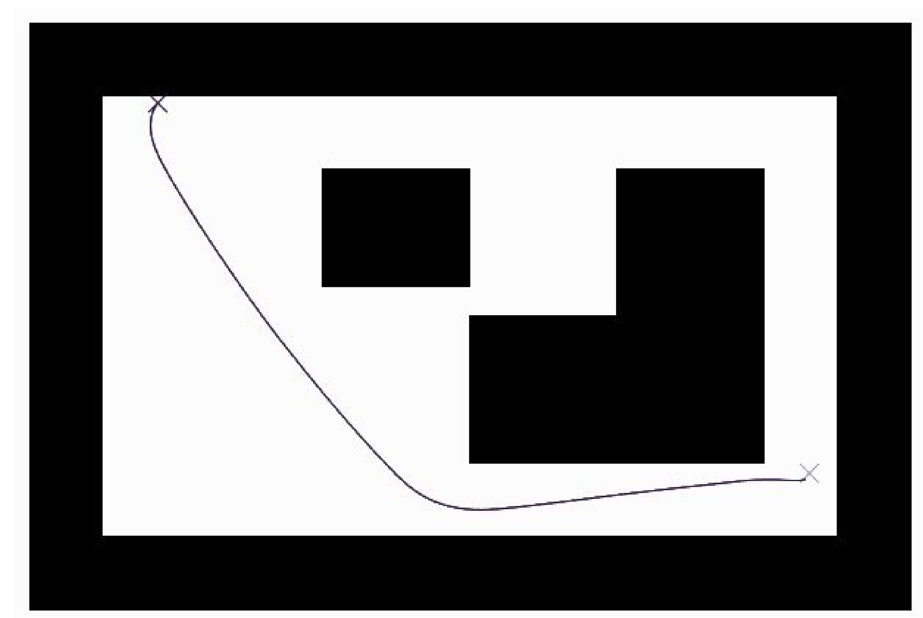
Results and Discussion

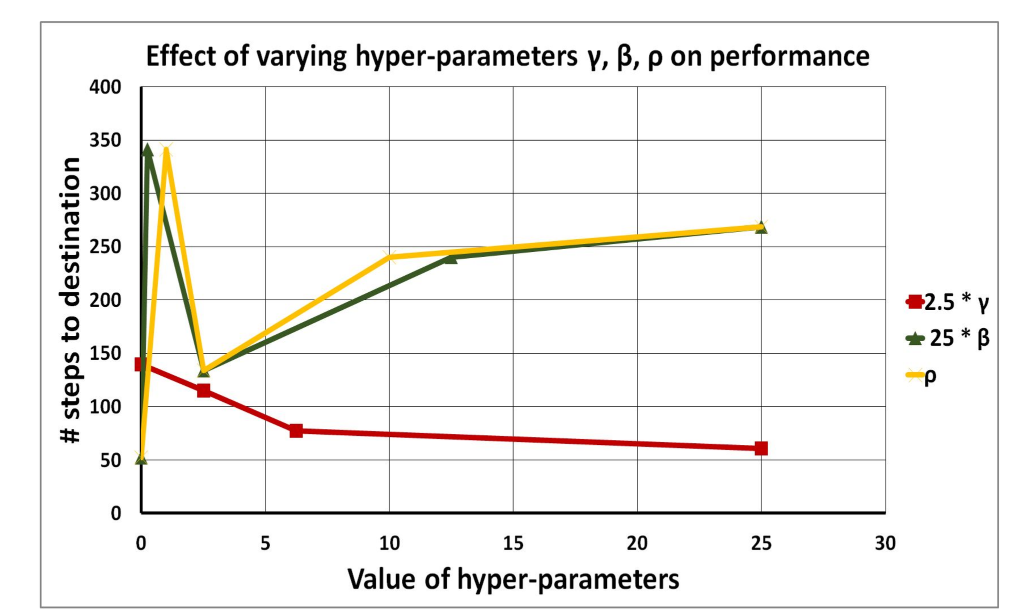
Simulation results obtained by implementing the proposed algorithm are shown below. Cross marks indicate destination and start positions. We have analyzed the effect of hyper parameters on the performance of our method. The observations have been plotted as number of steps required to reach a destination vs value of ρ,γ,β as shown.

We have also tested our method using multi layer-hidden neural network. The effect on number of hidden neurons for a single hidden layer is as shown on the right. This suggest us that the optimal number of neurons in the hidden layer is around 6 (hidden latent variables). 7 neurons are used in the simulations below.









FUTURE WORK: One of the possible directions in which we can explore, is to implement the above procedure on hardware and also check for more realistic rewards based on the actual properties of the available sensors like Ultra sonic sensors, LIDARS, GPS measurements for localization.

REFERENCES: **1.** "Richter, Charles; Roy, Nicholas; ",Learning to Plan for Visibility in Navigation of Unknown Environments. ISER 2016

2. "Bing-Qiang Huang, Guang-Yi Cao and Min Guo", Reinforcement Learning Neural Network to the Problem of Autonomous Mobile Robot Obstacle Avoidanc, ICMLC 2005