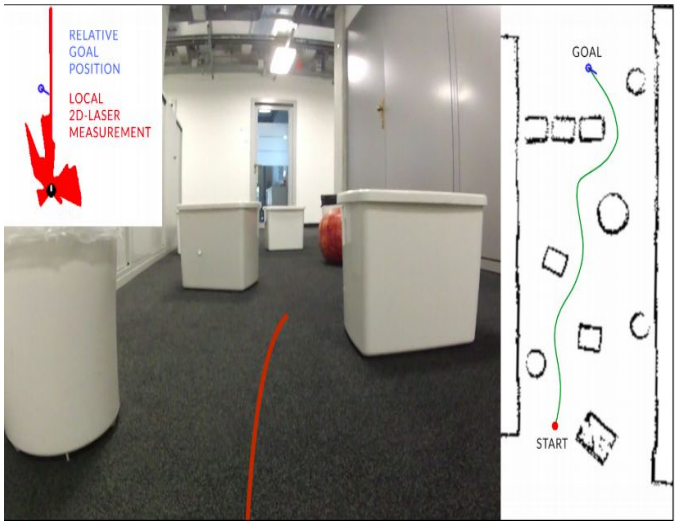
From Perception to Decision: A Data-driven Approach to End-to-end Motion Planning for Autonomous Ground Robots

Manzil Roy

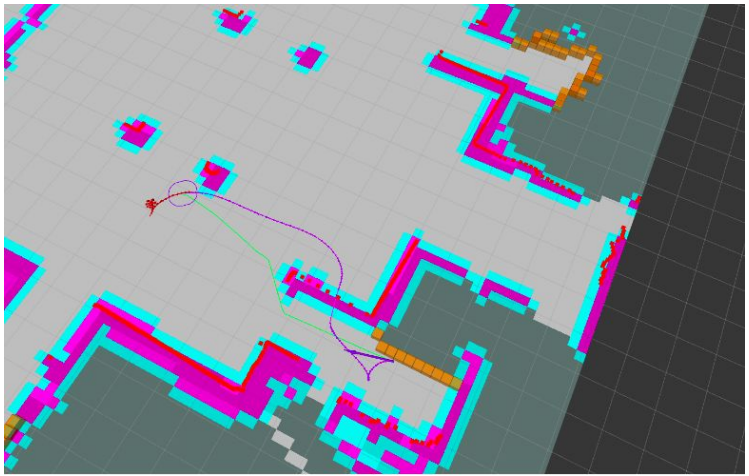
What is the problem statement ?

Make an autonomous robot navigate from an initial location to a target . You have 2D laser range findings and the target position .



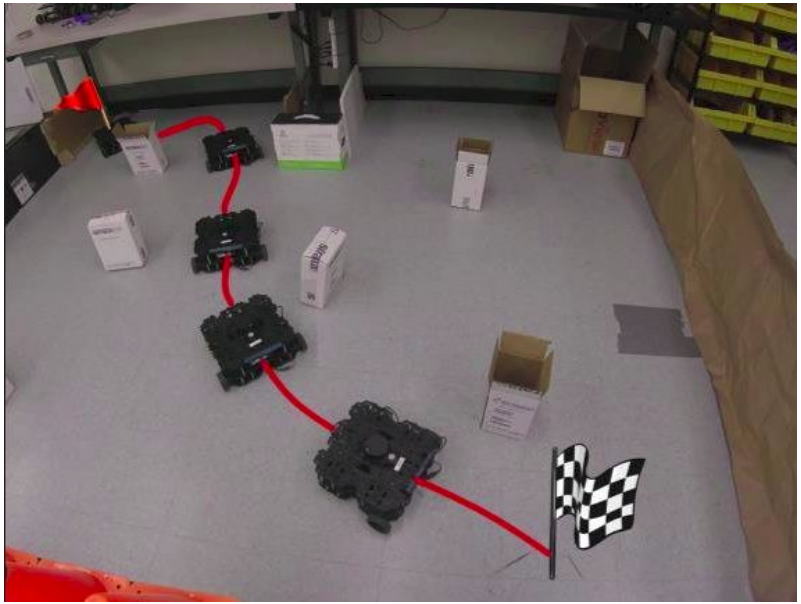
That’s quite general problem right ?

Yes !!! ROS has a navigation stack capable of doing this task. You have goals and obstacles and a global and local planner…



From algorithms to neural networks

ROS uses generic algorithms like Dijkstra for planning. The paper in question uses a neural network in its place .



Why a neural network ? Any benefits ?

A Deep neural network is a universal function approximator.

Once trained, it doesn’t require explicit programming for every separate scenario .

The function in question

The parameters are basically the features to train the network.

The output is the command (rotational and translational) to steer the robot.

$$\mathbf{u} = \mathcal{F}_\theta(\mathbf{y}, \mathbf{g})$$

The optimization criteria

We have to train a network with the following optimization criterion , and get the parameters .

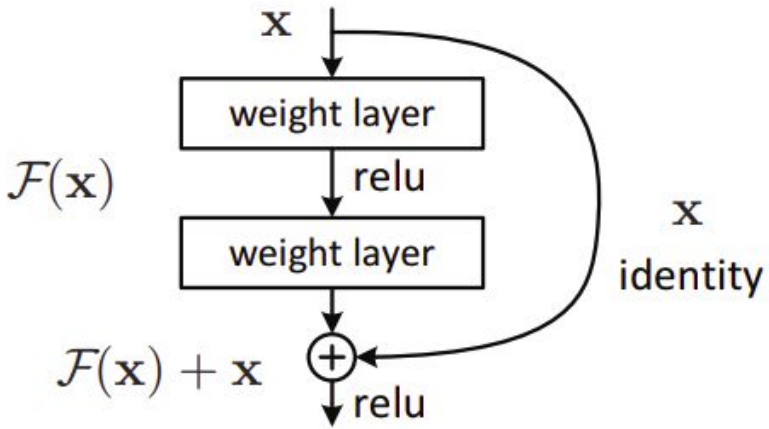
$$\left|\mathcal{F}_\theta(\mathbf{y},\mathbf{g}) - \mathbf{u}_{\text{exp}}\right|$$

Compare the output of the network (predicted output) with the training data.

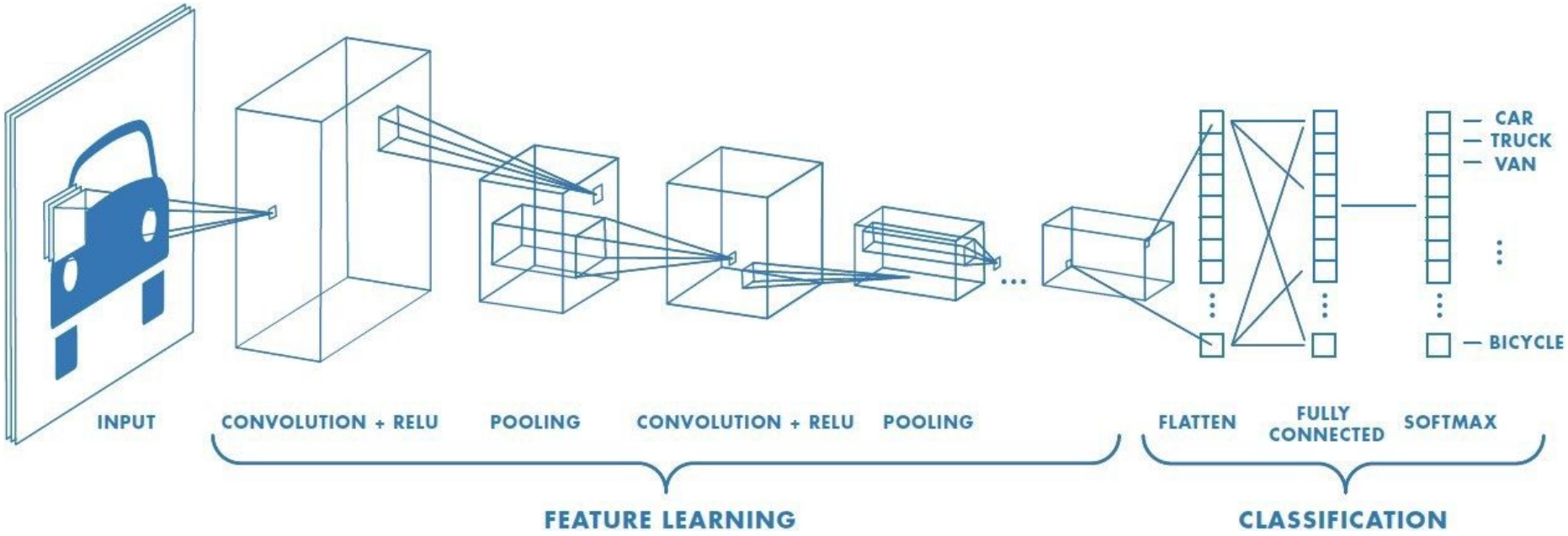
We will use a CNN with residual connections - A resNet

Yes, that's the model we will use for this particular problem .

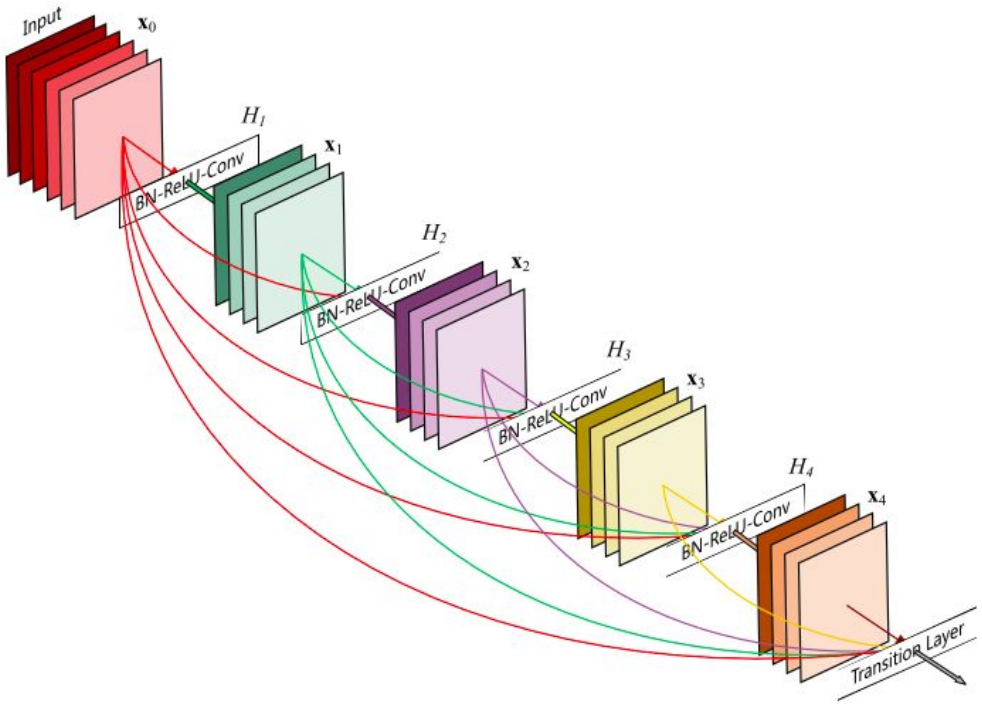
A basic residual block is shown alongside , an identity skip connection .



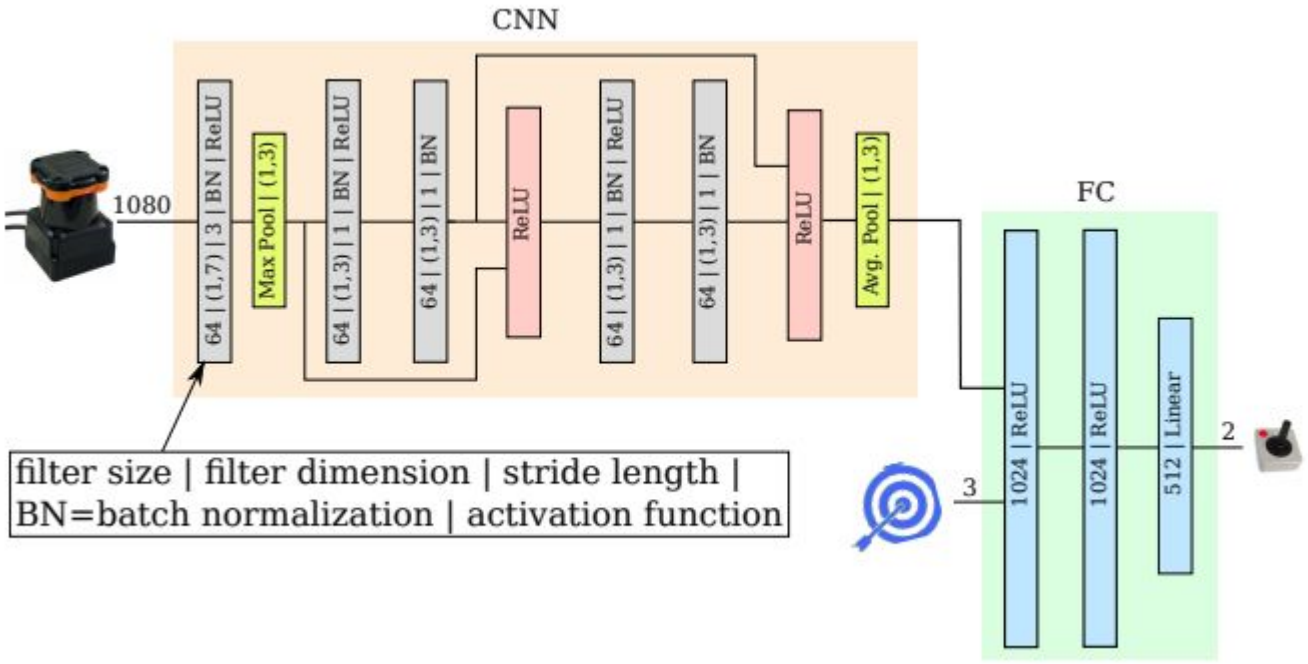
Layout of a Basic CNN



Layout of a Basic Residual CNN



Now...The Network in Question !



The Loss Function

$$J_k(\Gamma_B) = \frac{1}{N_B} \cdot \sum_{j=i}^{i+N_B} |\mathcal{F}_{\boldsymbol{\theta}_k}(\mathbf{y}_j, \mathbf{g}_j) - \mathbf{u}_{\text{exp},j}|$$

The tuples are randomly shuffled before training .

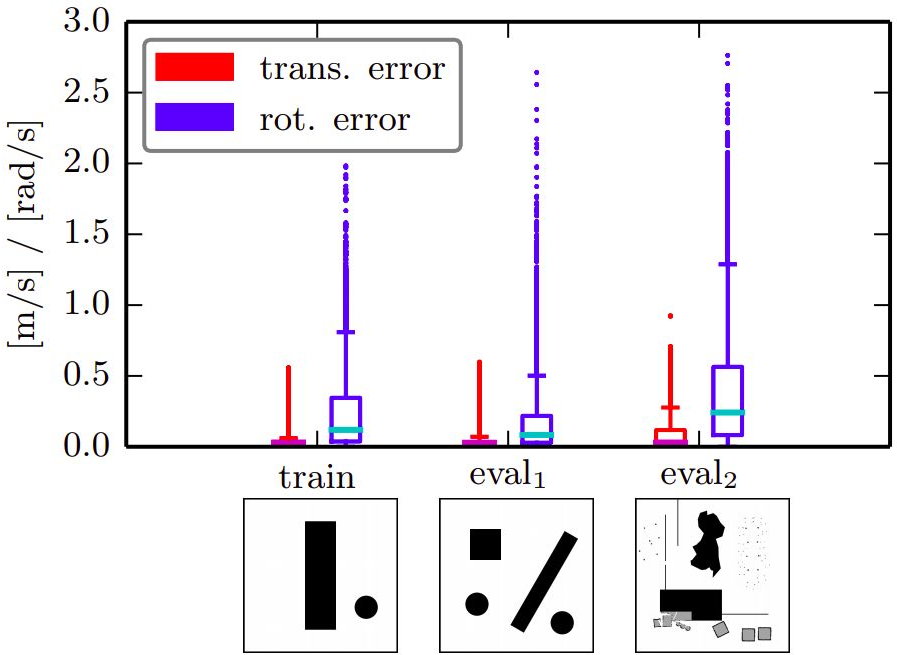
The optimization is done using Adam Optimizer , with mini-batch training.

The Training

Training is done in a simulation environment and train is the map on which network is trained.

ROS Stage 2D is used as the dynamic simulator.

The maps eval1 and eval2 are used for evaluation.

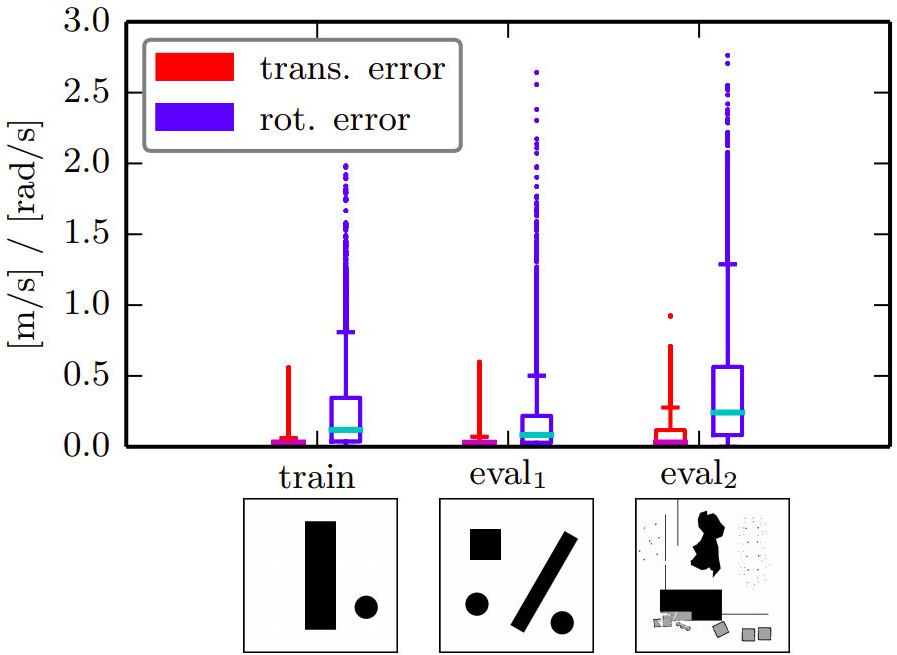


The Training

Training is done in a simulation environment and train is the map on which network is trained.

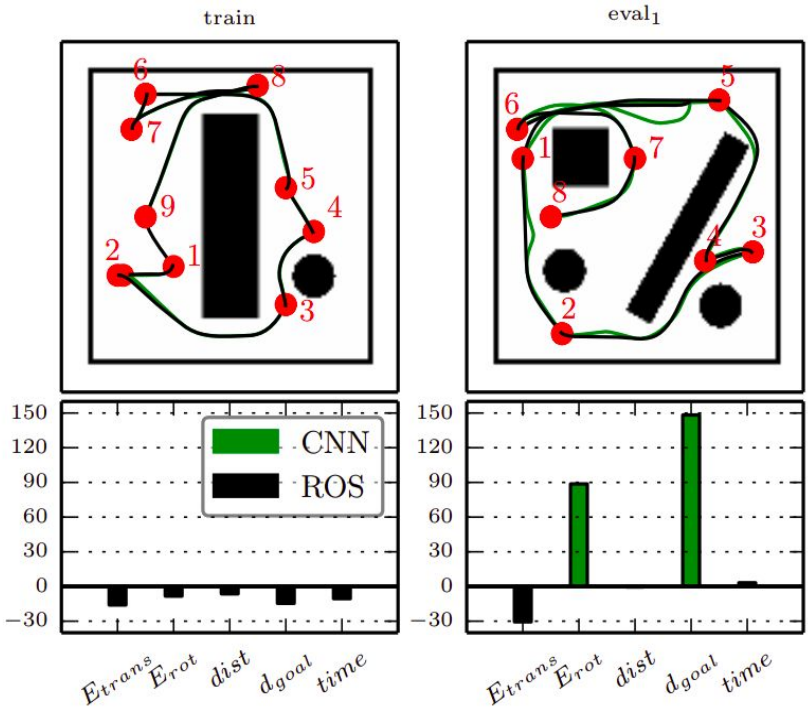
ROS Stage 2D is used as the dynamic simulator.

The maps eval1 and eval2 are used for evaluation.



Comparison of the trajectories

We can see that the trajectory generated by the CNN (on right) has deviated a bit from the trajectory generated by the expert planner.

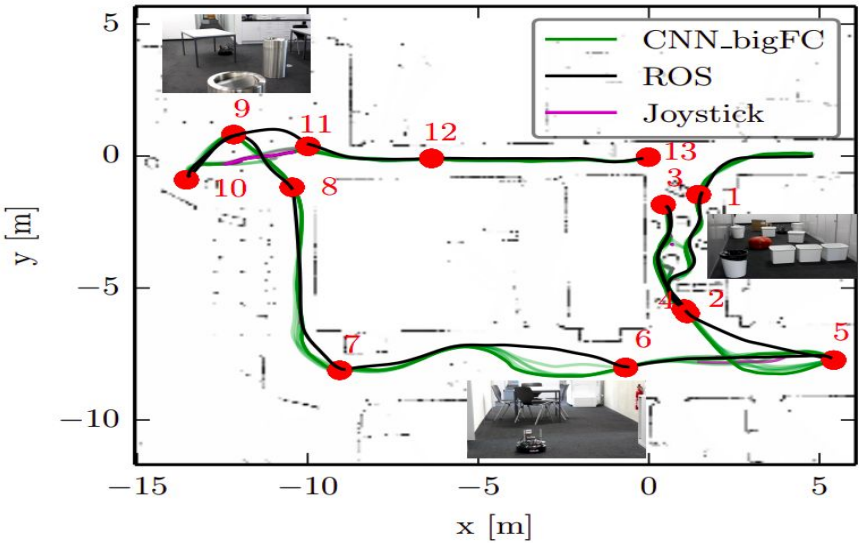


Running the trained network on real environment

Target	1	2	3	4	5	6	7	8	9	10	11	12	13
CNN_smallFC	1	0	1	0	4	0	0	0	2	4	6	0	0
CNN_bigFC	0	0	1	1	0	1	0	0	0	0	4	1	0

The FC dimensions of the FC layer has been increased.

Notice that this experiment has a human intervention (on purpose) .

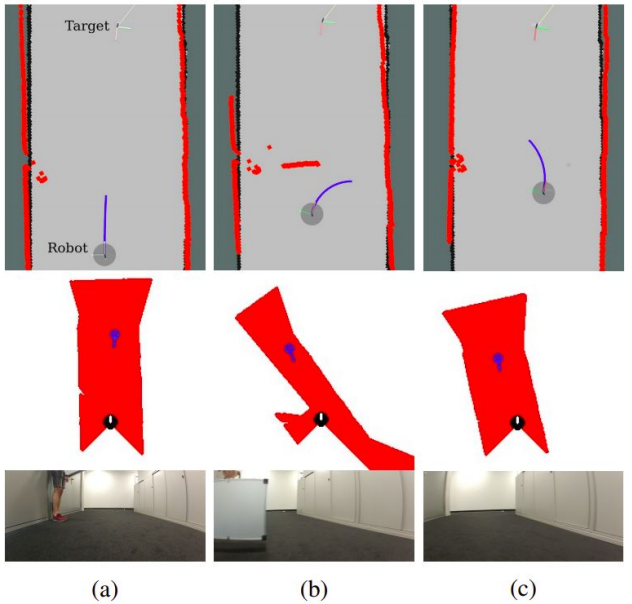


Reaction of the robot to a dynamically changing obstacle

If an obstacle is place in front of the robot it immediately changes its course.

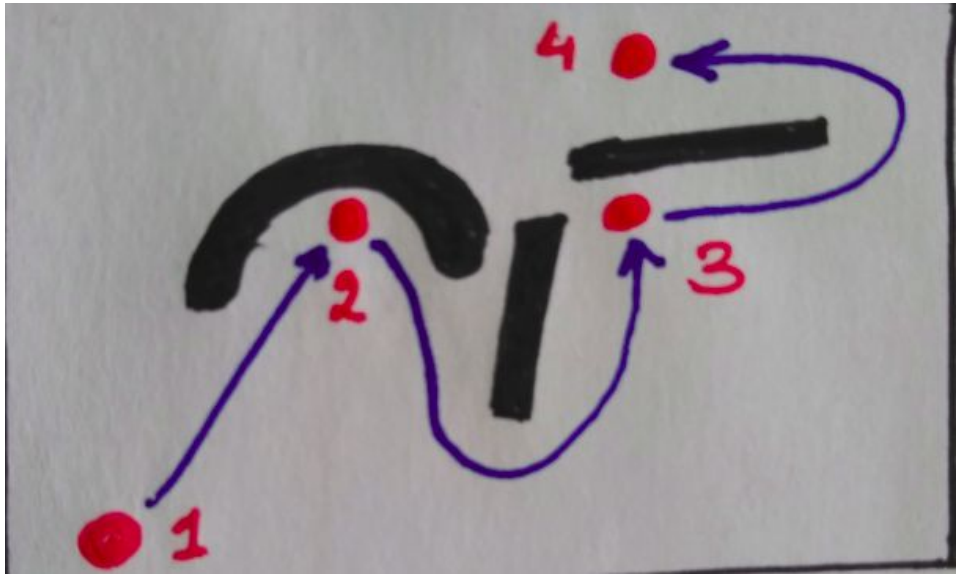
Upon removal, it changes back to its old path, without any intervention

This is an observation.



Dead-end problem observed

If the robot gets stuck in situation like this (a convex dead end), it is unable to get out of it.



What was the biggest advantage ?

The resNet has the capacity to transfer its knowledge from one map to another.

Any algorithm based motion planner needs global knowledge of the map, obstacles. Our model does not . There lies the biggest advantage of this approach.

We trained it in a simulation (low cost) and used it in real environment.

Is this the only network that will work in a motion planning problem ?

No !!! Not at all. This is a proposed and experimentally deduced research with results of limited feasibility.

We saw the drawbacks.

Lets see how we can improve our solution.

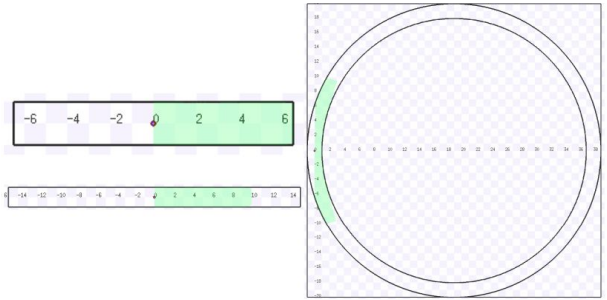
Improvements and further work

Multimodal Deep Autoencoder

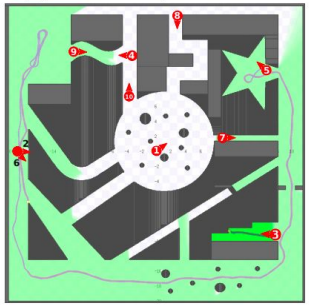
A multimodal deep autoencoder with RBM is used as the learning model.

Similar scenario , where training is done in a simulation and testing is done on real environment.

Works well in a complex environment



(a) Three Simple Corridors



(b) Complex Environment

MPNet (Motion planning network)

An iterative motion planning algorithm.

Combination of an encoder layer and a planning layer.

Shown to give much better results on 2D and 3D motion planning problems.

A very recent work.

Dimentionality Reduction

Remember this ?

$$\mathbf{u} = \mathcal{F}_{\theta}(\mathbf{y}, \mathbf{g})$$

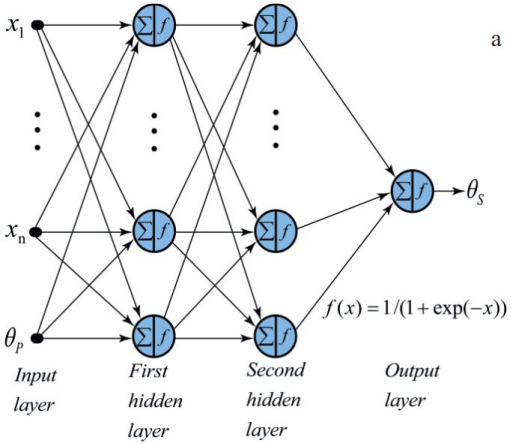
The features used in our experiment are composed of raw data from laser sensors.

We can use dimentionality reduction techniques on ‘y’ (like PCA) to extract meaningful features.

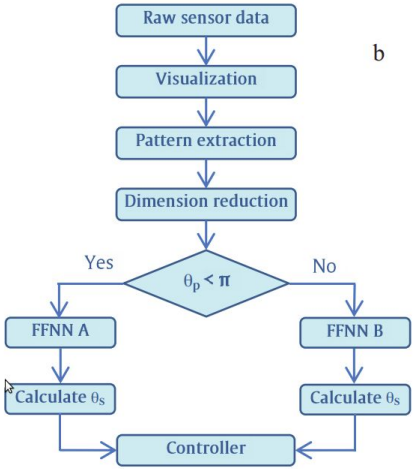
Train a network (it might be a simple one like an MLP) , on this feature set.

It has been done and comparable results have been observed for a similar problem.

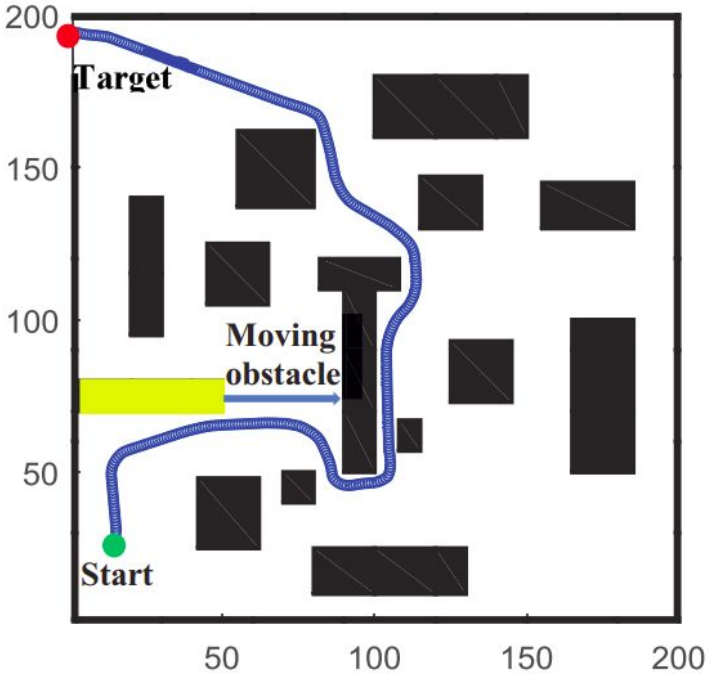
Dimentionality Reduction (Kernal PCA)



a



b



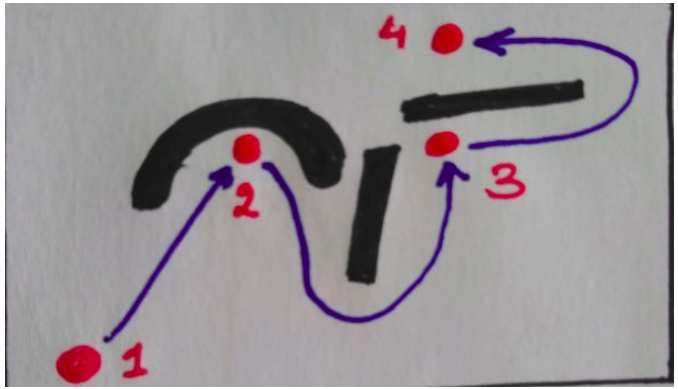
Recurrent Neural Networks

Recurrent neural networks with Short Term memory (LSTM) has been used as motion planners .

Highly complex maps have been tested upon.

Training and testing both has been done in real environment.

Seen to perform better in dead-end scenario.



Deep Reinforcement Learning

Much work has been done in this area to solve the motion planning problem.

Dynamic obstacle avoidance is solved most efficiently.

Gives better results than using DNN.

Thank you !