Neural Network Based Navigation of Mobile Robot in Maze Environment

Michal Kramarczyk 20154922

EE837 Neuro-Robotics course
KAIST, Fall 2015

Abstract

This project focuses on neural network based method for navigation of a autonomous mobile robot in a maze-like environment. The project was divided into several parts. First, simulation environment was implemented, including forward kinematics of the robot, generation of different maze environments, gathering of training data for neural network and visualization of the robot driving through the maze. Further, a neural network was introduced to the project in a form of multilayer perceptron with two hidden layers. Next task was to teach the neural network how to behave in certain situations using history of the laser measurements and wheel velocities corresponding to those measurements. Finally, several experiments were performed in order to test the robots' motion based on neural network for both known (training) and unknown (test) environments.

I. Introduction

Motion planning is one of the most fundamental issues in mobile robotics. There is vast theory on the topic and many different algorithms and methods have been proposed. However, these methods are very often complicated and difficult in implementation and are usually intended to perform only very specific task. Since popularity of neural networks in various problems is rapidly growing because of their versatility and adaptation abilities, in this project, an alternative approach to classical methods is proposed, with a use of a multilayer perceptron with backpropagation training algorithm.

II. Model description

The whole model was implemented in MATLAB. Simulation environment created for this project consists of robot kinematics implementation, creating different maze environments and simulating behavior of the robot in these environments. Further, neural network was introduced and used to teach robot specific behaviors.

i. Robot

Robot implemented in the project is a differential drive vehicle, i.e. vehicle with two parallel, separately driven wheels located in the back of the vehicle and one caster wheel in front. Sensors used in the robot where three laser distance sensors placed in front of the vehicle. Both wheels could move in either direction with any speed, however, for simplicity sake, in this projects movement of the wheels was limited to two states - move forward or remain still.

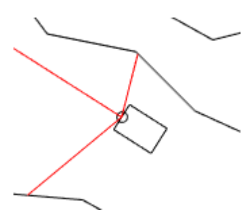

Figure 1: *Differential drive robot*

ii. Environment

All mazes used for the project were drawn in Microsoft Excel and then imported to the simulation via a function written in MATLAB. Mazes consist of simple constructs like starting and ending position, walls and obstacles.

iii. Neural network

Neural network chosen for this project was a multilayer perceptron with two hidden layers. Each layer containes 20 hidden neurons. Inputs to the perceptron were laser measurements and wheel speeds were calculated at the output. Perceptron was trained using backpropagation algorithm with 0.1 learning rate and 500 epochs per training.

III. Experiments setup

There were three types of experiments performed. First task was to make the robot go through a tunnel without crashing into the wall. Secondly, robots' task was to drive from starting to end position while avoiding obstacle on the way. Finally, the robot was supposed to drive through a maze-like environment.

Each experiment was done in following steps:

- training environment was generated,
- robot was run through the environment without the use of neural network and the training data (laser measurements and corresponding wheel speeds) was gathered,
- the perceptron was trained with the training data,
- robot controlled with NN was driven in training environment,
- robot controlled with NN was driven in newly generated test environment.

Test data was gathered in two ways. For the tunnel driving and maze-like environment, simple algorithms (wall repulsion and wall following) were used. In the case of obstacle avoidance, a static route for the robot was created. Although this specific approach was presented in the project, both methods could have been used in either task.

IV. Simulation results

First, the robot was tested for the tunnel driving task. Following results were obtained for the training environment and test environment (blue line is the trajectory of robot with training algorithm, red line corresponds to neural network control trajectory):

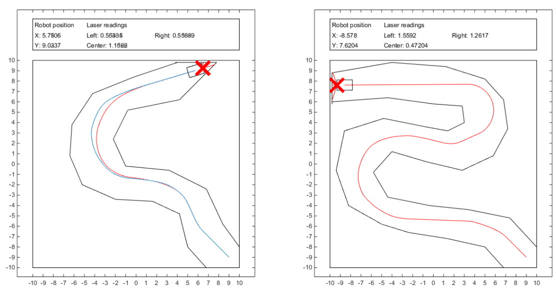

Figure 2: *Test (left) and training (right) environment for tunnel driving*

In the second experiment, the task was to teach the robot to get to the end of a tunnel, while avoiding an obstacle on its way.

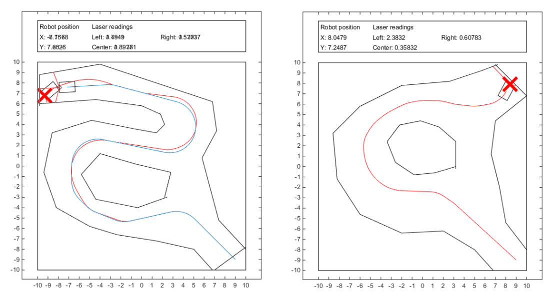

Figure 3: *Test (left) and training (right) environment for obstacle avoidance*

Finally, robot was taught to go through a maze-like environment:

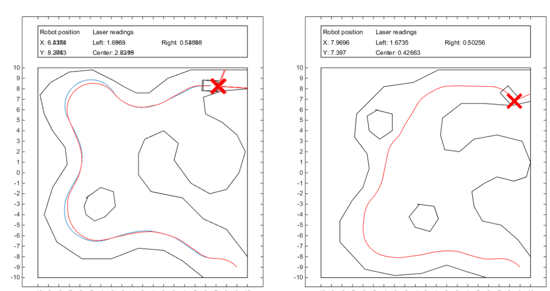

Figure 4: *Test (left) and training (right) environment for maze driving*

V. Discussion

One can see from the previous section that for the presented examples, robot driven by the neural network performed properly for both training and test environment. Mapping of the behavior in the training part is almost identical to the original trajectory and the robot performs each task as expected in the environments that he was unfamiliar with.
However, some problems were encountered during training and running the simulations. In some cases, robot had problems with distinguishing some situations from others that were similar, which resulted in some unexpected crashes. It also presented some general unwanted behavior in some rare cases, probably due to imperfections in generalization of the teaching algorithms. Moreover, due to low versatility of some training maps, robot did not recognize some situations and presented random or unexpected reactions. Furthermore, fairly poor maneuverability of the robot was problematic, since even when its' reaction was correct, but occurred to late, it would crash into the wall or the obstacle on the way.

VI. Conclusions

One can see from the experiments results that in general, multilayer parceptron might be a good tool for mapping mobile robots' algorithms and behavior and could be successfully used to teach a robot various tasks, like obstacle avoidance or wall following. However, for this project several improvements are proposed for further development. Firstly, implementing additional sensors would certainly help the robot to react faster and better distinguish similar situations, although it might significantly increase learning time, which should be taken into consideration. Secondly, bigger and more versatile environments should be created for training the neural network in order to avoid situations in which robot doesn't recognize certain situations and does not know how to behave. Moreover, future work should include introducing negative wheel velocities to improve robots' maneuverability, which should increase robots' reaction time and significantly decrease crashing instances.

References

[1] S. Hiking, 'Neural Networks and Learning Machines', Third edition, Pearson Inc, 2008.
[2] R. Siegward, I. R. Nourbakhsh, D. Scaramuzza 'Introduction to Autonomous Mobile Robots', Second edition, MIT press, 2011
[3] R. Rojas, 'Neural Networks', Springer-Verlag, Berlin, 1996