ENPM 808A: Introduction to Machine Learning

Final Project: Data Driven Motion Planning using Machine Learning Algorithms

Prepared by

Saketh Narayan Banagiri

UID: 118548814

Email: sbngr@umd.edu

M.Eng in Robotics

University of Maryland College Park

1. Introduction

Robot navigation entails the ability of a robot to establish its location within its frame of reference and plan a route to reach its desired location. Path planning is a crucial element of autonomous mobile robot navigation. It involves generating a map of the environment through the analysis of sensor data, which entails intensive computational processing. To minimize this computation, this project proposes a data-driven motion planning approach.

The goal of this project is to develop a machine learning model for a four-wheeled toy car with non-holonomic constraints, as it navigates through corridor and box environments using real-time sensor data collected through laser scanning. This approach is referred to as "Data-Driven Motion Planning."

1.1 Problem Statement:

- ➔ The main aim of the project is to develop a machine learning model for a four wheeled car with non-holonomic constraints for a certain environment with moving obstacles.
- ➔ The data is obtained through real-time experimentation utilizing laser scanning.
- ➔ It is expected that we compare various Machine learning techniques and choose a pipeline that gives out minimum error.

2. Data Preprocessing

The data is organized into columns as follows:

1. First 1080 columns represent the laser range data.
2. Next 4 columns represent final goal information.
3. Next 4 columns represent the local goal information.
4. Next 4 columns represent the robot's current position and pose information.
5. Final 2 columns represent the commanded actions.

Two environments were considered,

1. A corridor scenario with moving obstacles.
2. An open box/hall environment with moving obstacles.

2.1 Feature Extraction:

- ➔ Each column in the laser data holds 0.25-degree information. To minimize the dimensions, the laser data for a 22.5-degree range is taken into consideration by computing the average of 90 columns. This reduces the number of dimensions in the laser data from 1080 to 12.
- ➔ The relative distances between the local and final goal positions and orientations with respect to the robot's local and orientation are considered, reducing the goal and orientation information from 12 columns to 8. The last two columns (Cmd_vel_v and Cmd_vel_w) were left untouched.
- ➔ Thus the number of columns were reduced to 22 from 1094.

2.2 Data Splitting:

- ➔ The data was already separated into training and test sets.
- ➔ Further, I have separated the training set with a split of 0.1 to compare training and validation losses for my final model.

3. Models Used

I evaluated multiple Machine Learning Regression pipelines and chose the best one based on the lowest mean absolute error generated.

3.1 Linear Regression:

- ➔ Linear Regression is a statistical technique used for predictive modeling. It establishes a linear relationship between a dependent variable (y) and one or more independent variables (x).

- ➔ It is one of the most basic machine learning algorithms based on supervised learning.
- ➔ Mathematically, we can represent a linear regression as:

$$y = a_0 + a_1 x + \varepsilon$$

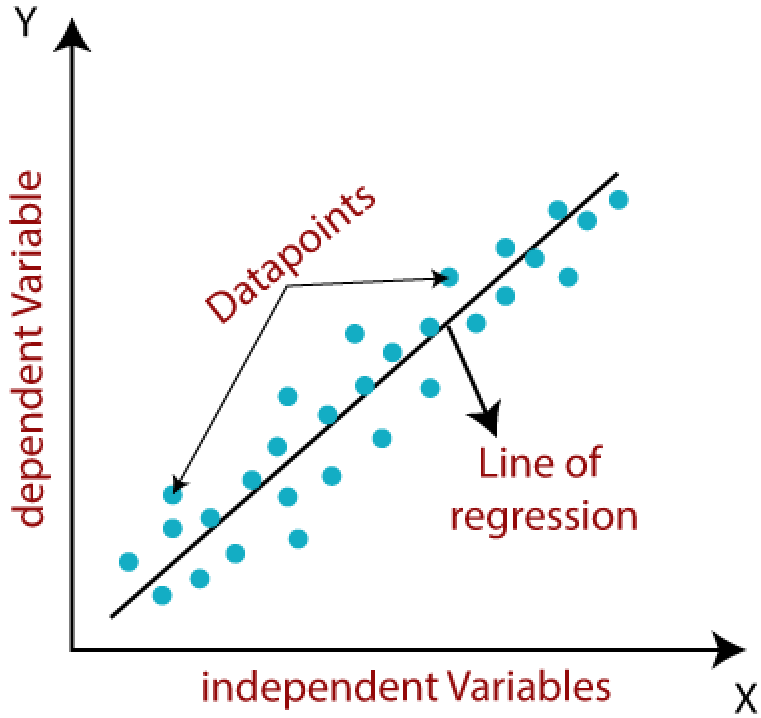

Fig 1: Linear Regression

3.2 Support Vector Regression:

- ➔ Support Vector Machines are a type of supervised learning model and its related algorithms used for both classification and regression analysis. In Support Vector Regression, the line used to fit the data is called a hyperplane.

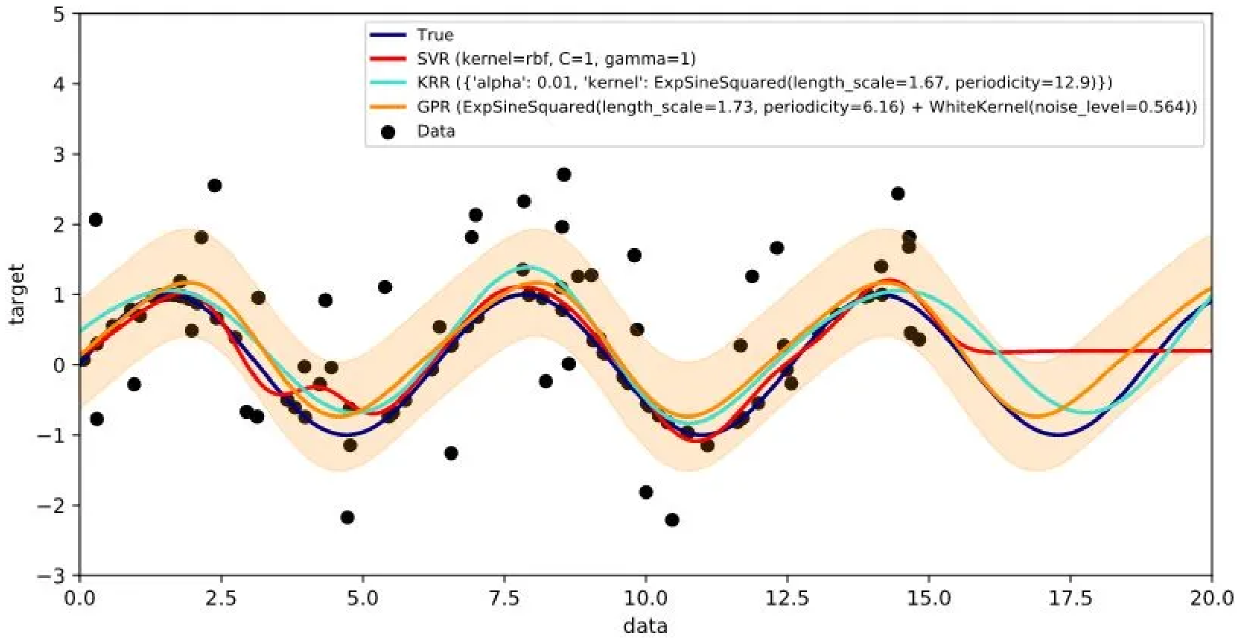

Fig 2: Support vector regressor

- The goal of the Support Vector Machine algorithm is to identify a hyperplane in an n-dimensional space that separates the data points distinctly. The data points closest to the hyperplane on either side are referred to as Support Vectors and play a crucial role in determining the position and orientation of the hyperplane, thus constructing the SVM.
- Support Vector Regression is a type of supervised learning algorithm used to forecast continuous values. It follows the same principles as Support Vector Machines (SVMs). The main objective of SVR is to find the line of best fit, which is the hyperplane with the greatest number of points.
- Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value.

3.3 Gradient Boosting Regression:

- "Boosting" in machine learning is a way of combining multiple simple models into a single composite model.

- ➔ Gradient Boosting is an ensemble of Decision Trees which works by sequentially adding predictors to an ensemble, each one correcting its predecessor. Decision trees are used as the weak learners in gradient boosting. Decision Tree solves the problem of machine learning by transforming the data into tree representation.
- ➔ Gradient Boosting Regression starts by computing the difference between the current prediction and the actual target value, referred to as the residual. Then, it trains a weak model to predict the residual based on the features. The residual predicted by this weak model is added to the current model input, which helps guide the model closer to the correct target. This process is repeated multiple times, improving the overall prediction of the model.

3.4 Convolutional Neural Network:

- ➔ A neural network is a collection of algorithms designed to identify patterns and relationships within a data set, by mimicking the functioning of the human brain. The term "neural network" refers to systems of artificial or organic neurons.
- ➔ Within Deep Learning, a Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification.However, we can also apply CNN with regression data analysis.

3.5 Error Comparison and Model Selection:

Mean absolute errors for all the four pipelines are obtained and given below. The test set used for the analysis is ‘Aug14_Box_5.csv’.

Machine Learning Pipeline	Mean absolute error
Linear Regression	0.24189557811305168
Support Vector Regression	0.26905703357030397
Gradient Boosting Regressor	0.20074039777749236
Convolutional Neural Network	0.18509529328224894

Clearly, Convolutional Neural Network outperforms the rest in terms of mean absolute error. Thus I selected CNN for total analysis of data.

4. Final model

➔ CNN has been used on the entire data and the errors obtained from the final model are as follows are as follows:

- ◆ E_{in_train} = 0.14334958132026288
- ◆ E_{in_test} = 0.17088120598505221

Initially, the model I created was overfitting, but I corrected it by adjusting the hyperparameters and implementing regularization.

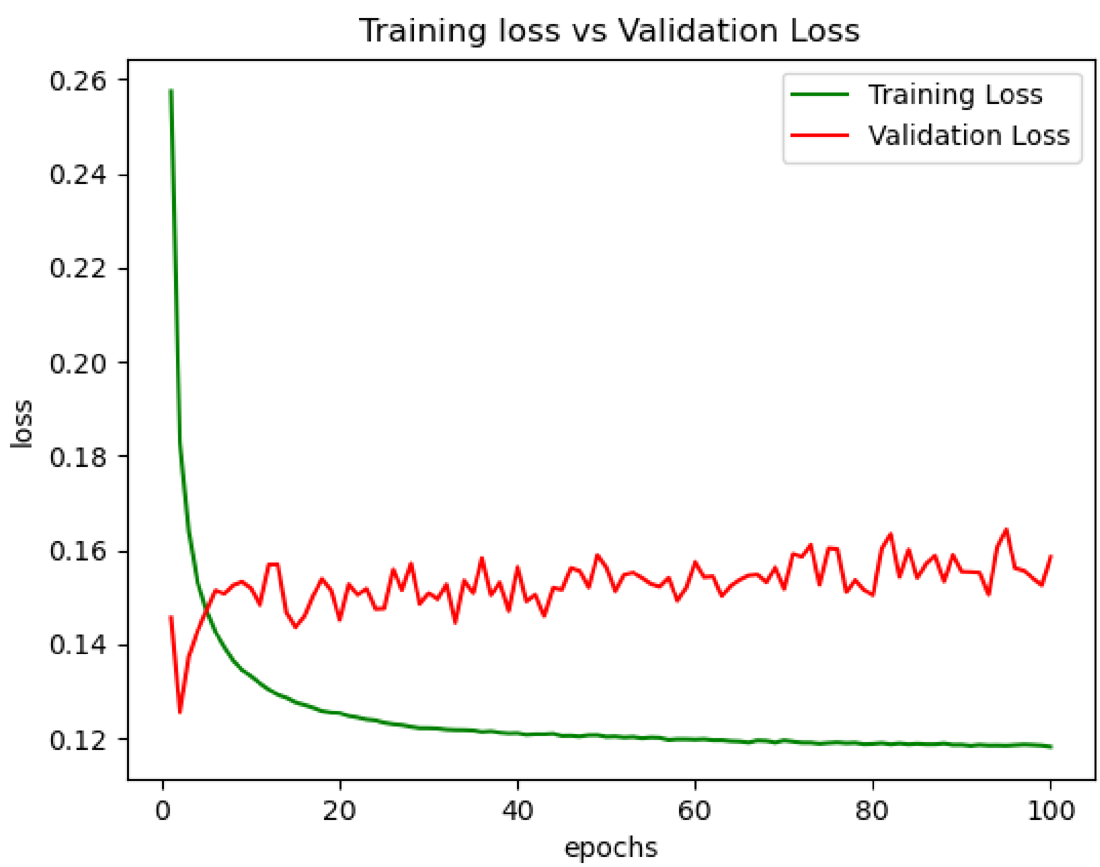

Fig 3: Training loss vs Validation loss before regularization

4.1Hyper parameter tuning:

The following hyperparameters were adjusted.

- **Depth**: Increased the number of layers and nodes. Used 6 layers in total.
- **Batch size**: Iterated the algorithm for various values of batch size.
- **Epochs**: Iterated the algorithm for various values of epochs and it was finally set to be 100

4.2 Regularization:

- Implemented L1 Regularization in the initial layer of the neural network.

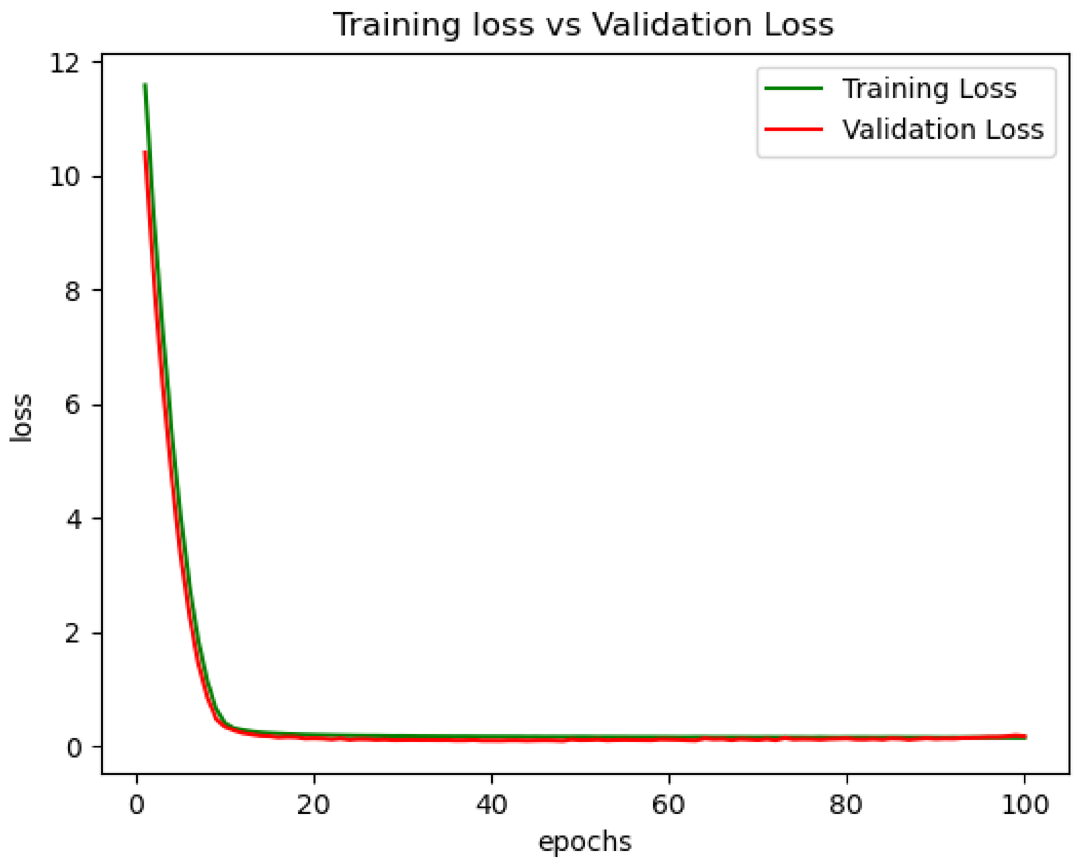

Fig 4: Training Loss vs Validation Loss after Regularization

5. Out of sample error using Hoeffding's Inequality

The total number of test examples (N)= 12, 220.

We know that;

$E_O \leq E_I + \varepsilon$

where, $\varepsilon = \sqrt{\frac{1}{2N} ln \frac{2}{\delta}}$

considering $\delta =$ 0.05% i.e. 95% probability

Therefore, $E_O \leq 0.01228$

6. Conclusion

Successfully developed a machine learning model for a four-wheeled toy car with non-holonomic constraints, using Data-Driven Motion Planning approach using Convolutional Neural Networks.

References

https://www.javatpoint.com/linear-regression-in-machine-learning

https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0

https://towardsdatascience.com/support-vector-regression-svr-one-of-the-most-flexible-yet-robust-prediction-algorithms-4d25fbdaca60

https://blog.paperspace.com/implementing-gradient-boosting-regression-python/

https://github.com/rsyamil/cnn-regression/blob/master/cnn_regression.ipynb

https://www.datatechnotes.com/2019/12/how-to-fit-regression-data-with-cnn.html