Development of a Level 2 Autonomous Vehicle Using Convolutional Neural Networks and Reinforcement Learning

Brendon Matusch

1. Introduction

In the autonomous vehicle industry, most research is done inside private companies. As such, it is difficult for a newcomer to the field to determine the best path to developing an autonomous vehicle. My goal is to test different techniques to gain knowledge on the most reliable, safe, and efficient techniques for autonomous driving.

This project builds on a set of general machine learning concepts, extending them and combining them with new innovations to develop a physical self-driving vehicle. These techniques were tested and their performance was compared both in a simulation and a real-world vehicle, helping to understand their relative merits.

My work primarily focuses on the steering system, but includes the development of systems for stop sign detection and location of vehicles ahead as a secondary goal for the sake of producing a complete level 2 autonomous vehicle.

2. Procedure

Before any testing of techniques could take place, a simulation was created using Unity and C#, including a track several kilometers in length and a vehicle constructed out of geometric components with physical proportions and mechanical imperfections similar to those of the real-world vehicle. Next, a physical vehicle was constructed, based on a modified go-cart with a servo motor for steering and a camera and LIDAR sensor for perception.

Several techniques were used for the steering system, which consists of vision and path planning components.

The first technique, the end to end convolutional neural network (CNN), calculates steering angles based on road images. It is trained on a large set of images of the road ahead alongside corresponding steering angles.

The second technique, a lane detection system, was developed, based on the use of a sliding window CNN to predict the locations of road lines within the image. Multiple algorithms were developed to calculate the centre of the line, arriving at an iterative process which searches out to the edges of the lane, locates points along the centre, and completes a set of outlier rejection passes, removing points distant from the calculated line of best fit.

The third set of techniques includes several methodologies for path planning. Initially, proportional-derivative (PD) control (a classical negative feedback loop) was utilized. This was augmented by an evolutionary algorithm in which reinforcement learning was used to optimize the proportional-derivative parameters (and later the weights of a two-layer neural network). Next, a deep Q-network and several variations on it were developed and configured to learn to drive on their own in the simulation. This was followed by a long short-term memory (LSTM) recurrent neural network, which was trained on time sequences of centre lines and corresponding steering angles.

Finally, a speed control system was created, which handles adaptive cruise control using a LIDAR sensor to detect vehicles ahead, and uses a sliding window CNN alongside conventional computer vision blob detection to find the location and size of, and approximate the distance to, stop signs.

3. Results

After numerous experiments and collection of new training datasets, the end to end CNN was successful at driving indefinitely in the simulation with a minimum standard deviation from the lane centre of 0.604 m. Despite similar optimizations with datasets collected in the real world, and improvements to the loss function, it was not able to drive more than 73.3 m in the real world (up from 45.7 m with the initial training dataset).

The first time that the lane detection system was tested in the real world, it followed the road indefinitely at 5 km/h with a positional standard deviation of 40.4 pixels. The parameters of the lane detection system were optimized over a series of 33 real-world testing runs. The final iteration achieved a standard deviation of 18.4 pixels at 15 km/h.

The evolutionary optimization technique was successful in the simulation; it improved the standard deviation from 20.16 pixels (with manually optimized parameters) to 15.37 pixels. However, when it was generalized to a two-layer neural network, it produced a standard deviation 14.4% greater than with proportional-derivative control.

The deep Q-network demonstrated the ability to drive for over an hour, albeit with significant oscillation and a standard deviation of 0.793 m. Many subsequent tests using various parameters failed to improve performance.

LSTM steering failed. It was unable to correct back to the lane centre, never driving for more than 9.3 seconds.

4. Conclusions

1. Producing a full level 2 autonomous vehicle is attainable with limited resources. This included a steering system, stop sign and forward vehicle recognition.
2. CNN Sliding Windows combined with PD control (optimized by hand or with evolutionary algorithms) proved the most effective and practical technique for steering the vehicle, both in the simulation and the real world.
3. The end to end CNN worked in the simulation but proved too fragile to be used in the real world. The black-box nature of the technique means that it is exceedingly difficult to troubleshoot. It is academically interesting but impractical.
4. Deep Q-Networks demonstrated potential but require further work. There is some question about whether the discrete action space of deep Q-networks is appropriate for this application.