Autonomous Driving through Imitation Learning

Ron Domingo

Stanford University Department of Electrical Engineering

Overview

Path planning is a complex yet essential aspect of autonomous driving that is required for vehicles to traverse the world around us. Finding paths is a complicated task that involves the avoidance of both static and dynamic obstacles, all while ensuring that the vehicle stays within the drivable area of the road.

This project aims to apply imitation learning algorithms to teach an autonomous vehicle to drive around a closed loop track while avoiding obstacles.

Problem Setup

Simulation Environment

- The Duckietown self-driving car simulator is an OpenAI Gym extension that serves as a simulator for the Duckietown hardware robots. The driving environment features marked roads, intersections, static objects such as buildings and trees, and finally dynamic obstacles like pedestrians that move around throughout the maps randomly.

Observations

Figure 1. In-game screenshot

- Observations from the driving simulator come as first-person perspective images of size (640x480x3)
- Raw images are downscaled using nearest-neighbor resampling to produce images of size (160x120x3)
- Downscaled images are then normalized

Actions

- Simulator uses continuous actions by default - for forward velocity and steering angle
- Action space during training has been discretized into four possible actions - left, right, forward, or stop

Learning Models

Two approaches for Convolutional Neural Networks (CNNs) were tested to create the autonomous driving agent in the simulator:

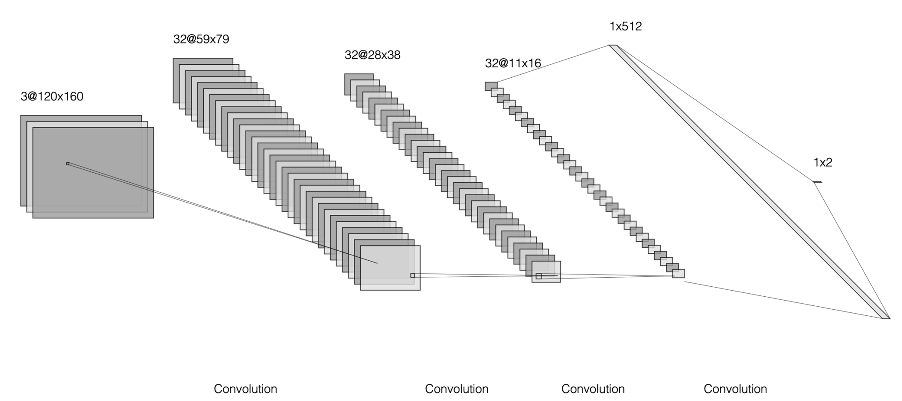

Figure 2. Single Frame CNN

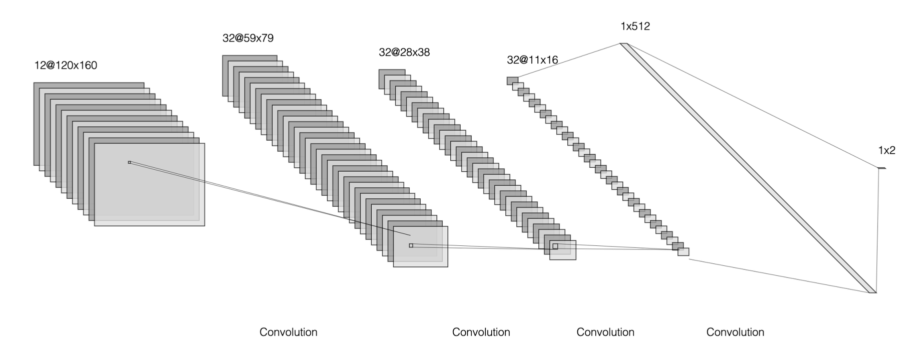

Figure 3. Multi Frame CNN

Single-Frame CNN

- Takes in a single image as input and produces an action based on current observations
- 4 convolutional blocks
- 2 fully connected layers

Multi-Frame CNN

- Takes in the last 4 image frames as input and produces an action based on the sliding observation window
- 4 convolutional blocks
- 2 fully connected layers

Evaluations and Results

Training Results

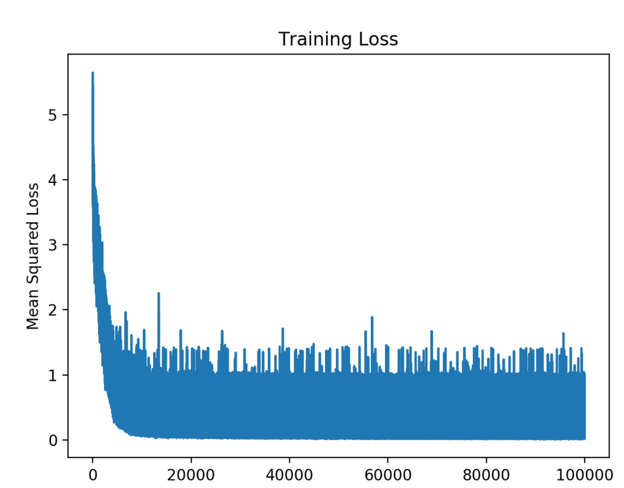

Figure 4. Single Frame CNN Training MSE

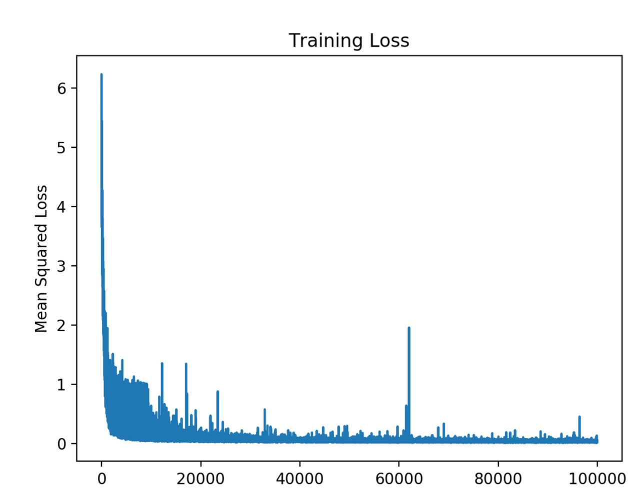

Figure 5. Multi Frame CNN Training MSE

Testing Results

Single-Frame CNN: 4.5%

Multi-Frame CNN: 0%

Discussion

Single vs Multi Frame CNN

In both models, the CNN was used to estimate the Q-function of the agent, where the camera observations were used as inputs and the best action was the output.

Single-Frame CNN Behavior

- Better memorization
  - 37% success rate on training tracks
- Makes decisions even on unseen trajectories

Multi-Frame CNN Behavior

- Unable to move away from unseen trajectories
  - Gets stuck and refuses to move or drives in circles when faced new observations

DAGGER

The DAGGER algorithm was applied to both models as an attempt to improve their driving performance, however, the opposite occurred. In both models, retraining with DAGGER did not lead to converged training losses as they did with the imitation learning training. This could be due to the setup of the correction recordings as they were simply overlaid onto a continuous video, leading to approximations.

Limitations

- Expert training data was collected by manually driving through the simulator while avoiding obstacles, leading to a small training dataset
- Camera position provides limited situational awareness
  - The sight of the road is often lost during turning maneuvers
  - Lack of peripheral vision when turning

Future Improvements

- Deeper CNN for better Q-function estimation
- Larger and more comprehensive expert dataset
- Improved DAGGER retraining
- LSTM approach
- Active Reinforcement Learning after Imitation Learning