Let's Drive: Learning Autonomous Driving through Behavior Cloning

Introduction

Context: Autonomous Vehicles are the next biggest challenges in AI and ML.

Challenge: Learn to output steering angle to have the vehicle stay within the lane via supervised learning on driving examples.

Approach:
- Obtained driving data through Udacity’s Self Driving Vehicle Simulator
- Several data augmentation techniques to overcome bias in the dataset
- Implemented, trained and tuned Convolutional Neural Network
- Acceptable error and driving performance given the relatively small number of data

Data Augmentation

- Cropping of top and bottom border to input only the road image
- Random rotation + random crop
- Random crop + change in brightness
- Random rotation + hue change
- Gaussian noise (added as a layer)
- Images were split into YUV channels as inputs to the neural net

Data Structure

- 17,350 images of the road (320px x 160px) from the left, center and the right side of the vehicle

- .csv file of image locations, steering angle, applied brakes and throttle, and vehicle speed was input to the neural net

- Based data; most of our driving examples are along the straight lane. Overcame by augmenting driving examples along the curvy roads.

Model

Model is trained based on MSE error of steering angle prediction, i.e.,

\[
\text{Error} = \frac{1}{N} \sum_i (\text{steer}_{true} - \text{steer}_{output})^2
\]

- Model: 5 convolutional layers for feature extraction
- 3 fully connected layers works as the ‘controller’
- 9 total layers

Results

- 0.025 training error
- 0.025 validation error

Comparative result with huber loss
- Agent can run autonomously without human interaction 86% of time
- Agent struggles more when tested in completely different environment (challenge tracks in the simulator)

Discussion

- Relatively small number of images (17k datapoints with random augmentation at each epoch) was enough to fully train the CNN with the acceptable MSE error
- Relatively simple architecture and certainly less complicated design than conventional component-by-component design approach
- Simulation setup greatly simplifies the problem and limits validation
- Lower vehicle speed
- No other vehicles on the road
- Relatively homogeneous environment
- Lower performance on ‘challenge’ tracks in the simulator indicate that the model potentially has issues with generalization
- MSE error is not representative of how safe the output steering angle is

Future Work

- Reduce the number of CNN parameters
- Use more expressive loss functions (total variation) to reduce jerkiness in steering output
- Input the series of camera frames as the training example instead of a single YUV channel of camera images using RNN

Reference