Self Driving Car

Noah Buchanan
Sam Donaldson
CS 4143: Deep Learning
University of Arkansas - Fort Smith
Spring 2022

May 7, 2022

1 Methodology

Images and steering values were collected from the car while manually driving it around the track. Several different models were trained with different hyperparameters to see which generated set of weights preformed the best in the real world. In practice, there was only time to train one model with several different hyperparameters to determine the best out of the weights generated.

2 Data

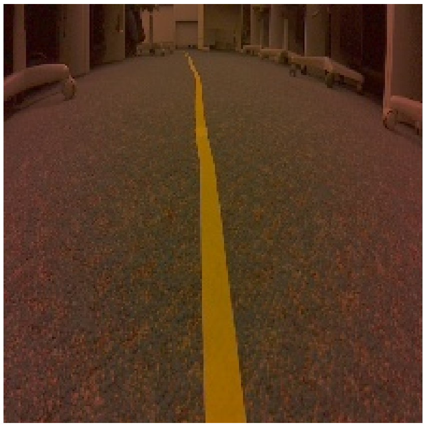
2.1 Data Gathering

Our team took a simplistic approach to gathering the data required to train our model. We reprogrammed a controllers right trigger button to take a picture and save the current steering value. The left and right sticks were used to navigate the vehicle. Pictures were taken intermittently at a multitude of angles relative to the track so that the model would have sufficient data to know the correct steering value in all situations.

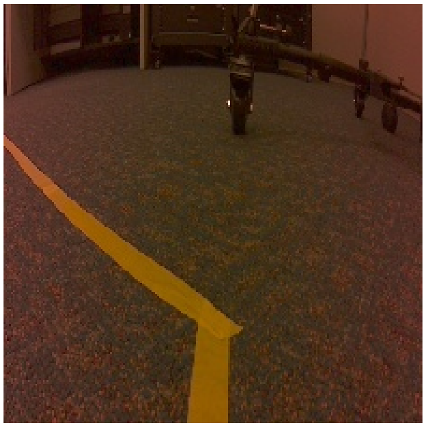
2.2 Data Examples

Throttle value was included for experimentation, it was ultimately scrapped and set to a constant for the duration of the cars driving. The track we chose to train on was a singular line as opposed to a multitude of lines or a lane following approach.

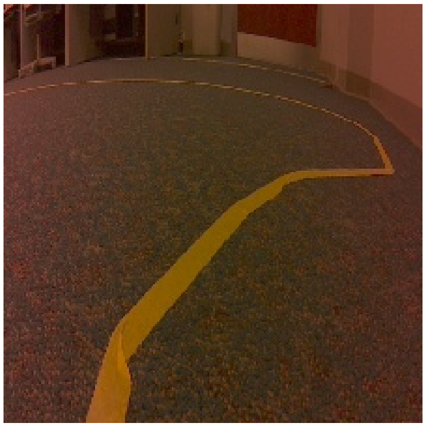
Below are some examples of the data we trained on:



Steering value: 0



Steering value: -1



Steering value: 0.79

3 Training

3.1 Model Architecture

- input shape: 224x224x3
- layer 1: Conv2D, 224x224x9, ReLu
- MaxPooling: 2x2
- layer 2: Conv2D, 112x112x18, ReLu
- MaxPooling: 2x2
- layer 3: Conv2D, 56x56x36, ReLu
- MaxPooling: 2x2
- layer 4: Conv2D, 28x28x64, ReLu
- layer 5: Conv2D, 28x28x128, ReLu
- Flatten
- layer 6: Linear (Dense), 100352x256, ReLu
- layer 7: Linear (Dense), 256x128, ReLu
- layer 8: Linear (Dense), 128,1, TanH

3.2 Training Details

The model trained for 50 epochs on the hand-made data. Increasing or decreasing the number of epochs, or using the original values for steering, harmed the car's ability to follow the yellow line.

4 Evaluation

The results from our selected model were acceptable with a MSE loss nearing 0.001.

5 Car Details

- Car Operating System: Ubuntu
- Neural Network Package: PyTorch
- Camera Package: OpenCV