Meredith Meyer

Project 4: Behavioral Cloning

Data Collection

My approach in data collection for the final model used was to supplement the provided sample data set. I trained the car in recovery laps for correcting back to center once off the center of the road. I augmented the data set by flipping the images and inversing the steering values. I used the recommended validation set split factor of 0.2.

Model

The architecture I used for the model was based off of the NVIDIA model. I fed the raw images into the keras model, which then both cropped and normalized the data for the images. The model includes 5 convolution layers, a flatten layer, and 3 dense layers.

To train my model, I used a generator function. Within the generator is where I actually loaded the images and steering values, and flipped them to double my data size. I focused on using a low number of epochs (5).

Results & Discussion

I was able to successfully train my car to stay on the road, with a final validation loss of 0.0177. But I had a lot of trouble getting to this point. I initially had tried to train the model solely with myself-gathered data. I had also been using the left and right images with offset values of 0.1-0.2 to supplement the data. I was able to train my model this way with low loss values, but the car would drive off the track at the first slight curve. I eventually opted to start from ground zero using the provided data set, thinking that my method of driving the car was not doing well in training- but this didn't solve the problem either. Finally, I left out my left and right images from the data set, and instantly got better results (loss ~0.01). The car still ran off the track consistently at one spot, so I supplemented the data with my own recovery data. This gave my a slightly higher loss (0.0177), but was able to keep the car driving on the road at all times.

I would narrow down my issues on this project to 2 main points: data management, and the side cameras. I didn't understand- and frankly still don't- how dating saving works with the simulator: what is lost when you exit the simulator, what is lost when you exit the workbook, or what is lost when exiting GPU mode. So, I wasted a lot of time re-collecting, downloading, and uploading data, because I needed to be sure I was using the right things. I also had to manipulate the file locations in my pipeline, because they didn't always save uniformly. As far as the side camera images, I'm not sure why they made my model so horrible. I played with the offset value, and the model

Meredith Meyer

never performed well. I'd be interested to see how someone successfully implemented those side camera images.