Behavioral Cloning

Vefak Murat Akman

Behavioral Cloning Project

The goals / steps of this project are the following: * Use the simulator to collect data of good driving behavior * Build, a convolution neural network in Keras that predicts steering angles from images * Train and validate the model with a training and validation set * Test that the model successfully drives around track one without leaving the road

Personal Notes: I was unable to collect new data from simulator. The FPS was not good and It made collecting proper data impossible

1. Included Files

My project includes the following files: * model.py containing the script to create and train the model * drive.py for driving the car in autonomous mode * model.h5 containing a trained convolution neural network * writeup_report.md or writeup_report.pdf summarizing the results * run1.mp4 final output video

2. Autonomous Mode

Using the Udacity provided simulator and my drive.py file, the car can be driven autonomously around the track by executing

```
python drive.py model.h5
```

To record video, there is a fourth argument `run1` for collecting frames

```
python drive.py model.h5 run1
```

3. Creating Video

With video.py file, collected frame converted to video file.

```
python video.py run1
```

Model Architecture and Training Strategy

1. Model architecture

I have used the CNN model architecture described in *End-to-End Learning for Self-Driving Cars* paper of nVIDIA, which is published in 2016. First two layer of model is image processing layers. First layer is about normalizing and second one is about cropping image. I only considered half bottom of image to discard unnecessary features such as trees or clouds. In addtion, I added two dropout layers with 0.25 key_prob to reduce overfitting.

alt text

Figure 1: nVIDIA Model

Layer	Description
Input	160, 320, 3 RGB Image
Lamda	Normalizing Image to 0-1
Cropping	Output Shape (65, 320, 3)
Conv2D 5x5x24	2x2 stride, outputs: 31x158x24
ELU	
Conv2D 5x5x36	2x2 stride, outputs: 14x77x36
ELU	
Conv2D 5x5x48	2x2 stride, outputs: 5x37x48
ELU	

Layer	Description
Conv2D 3x3x64	outputs: 3x35x64
ELU	
Conv2D 3x3x64	outputs: 1x33x64
ELU	
Flatten	Output: 2112
FC1	Output: 100
Dropout	Prob: 0.25
ELU	
FC2	Output: 50
Dropout	Prob: 0.25
ELU	
FC3	Output: 10
ELU	
FC4	Output: 1
ELU	

Total params: 348,219 Trainable params: 348,219 Non-trainable params: 0

2. Attempts to reduce overfitting in the model

The model contains dropout layers in order to reduce overfitting. The model was trained and validated on various images. Unfortunately, I cannot collect data by myself due to connection problem. So, I wrote codes for data augmentation to increase variety of dataset.

3. Model parameter tuning

The model used an adam optimizer, so the learning rate was not tuned manually.

4. Appropriate training data

Training data was chosen to keep the vehicle driving on the road. I used a combination of center lane driving, recovering from the left and right sides of the road.
alt text

Pre-processing

I have only used data provided by Udacity. Due to connection problemi I was not able to collect more data. So, I had to increase variety of my dataset. First, I have used both three camera images (Center, Left and Right). Then I applied some image processing techniques. They are random brightness change, flipping only center images, sharping and translation. The streering angles calculations for translation are made by Vivek Yadav.(His blog)

Other important aspect is that I added correction value to right and left camera images in order to prevent model to make bad steering move. I determined variable named `corr` for this;

```
 if i==0:   #If image is taken by center camera
    corr = 0
elif i==1: #If image is taken by left camera
    corr = 0.2
else:      #If image is taken by right camera
    corr = -0.2
```

In the below, there are outputs of image augmentation methods.

++Original Image:++
alt text

++Sharped Image:++` alt text

++Flipped İmage++ alt text

++Bright İmage++ alt text

++Translation İmage++ alt text

Final Video – Results

Please find youtube link of the final video.

Potential Improvements on the Model

- The last image shows the only part where the car drives above the lane markings and then come back to the track. The car is still on the driveable area however the model needs improvement in that section.
- That part is right after the bridge where the driveable area color and lane markings are slightly different than each other which requires additional attention/training for the model.
- The model can drive car in first simulator track. There should be collected more variety of data.
- The accuracy rate is so low. However, the model finished the track successfully. The real-life data mayn't match ones in training section.