Project 3 Behavior Cloning

1. **Introduction**

   The goal of the behavior cloning project is to train a convolution network to drive a car in a simulation environment (Windows_sim). The training data was obtained by recording the driving images and control (throttle, break, steering angles) from a human driver driving through the same track in the simulator. There are three image available (left, center, right) for each time tick and the control inputs corresponding to the same images were recorded as well.

2. **Model Architecture**

   I started with the Nvidia network, with all the data augmentations (more details later) and parameter tuning (more details later), I got a working model that can drive the car through the track1. The problem however is that the model parameter file is very large, approximating 90Mbs. So I spent a lot of time on simplifying the model. The final version is about 17Mb. Main difference from the Nvidia model:

   a. Add lambda to normalize the input image
   b. Add cropping to remove pixles that not part of the driving track
   c. Reduce the fully connected layers from 1164 to 200, removed one fully connected layer (100)

   I used cv2's imread function to import image, which comes in BGR format, the drive.h however used RGB format. Without noticing this difference, I had a hard time in the beginning in getting a stable model to run the track accident free, after fixing this, the model is much more stable.

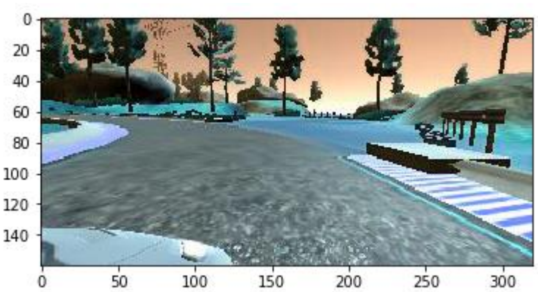
Layer	Description
Input	160x320x3 BGR image
Lambda	Normalization
Cropping	80x320x3 RBG image (top 60 and bot 20 removed)
Convolution 5x5x24	2x2 stride, Valid padding
RELU	
Convolution 5x5x32	2x2 stride, Valid padding
RELU	
Convolution 5x5x48	2x2 stride, Valid padding
RELU	
Convolution 3x3x64	1x1 stride, Valid padding

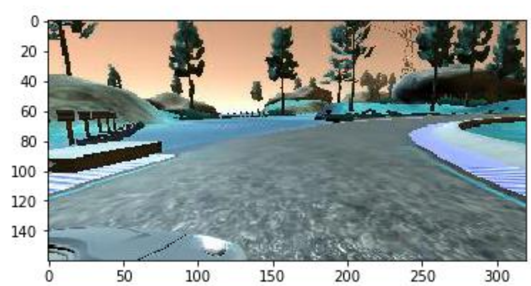
| RELU
|:-------------------------:|:--:|
| Convolution 3x3x64 | 1x1 stride, Valid padding
| RELU
|:-------------------------:|:--:|
| Flatten
|:-------------------------:|:--:|
| Dense | output 200
| Dropout
|:-------------------------:|:--:|
| Dense | output 50
| Dropout
|:-------------------------:|:--:|
| Dense | output 10
|:-------------------------:|:--:|
| Output | outputs 1

3. Data Augmentation

There are three images (left, center, right) for each steering angle. In order to use the left and right image, I added 0.09 to the left image and subtracted 0.09 to the right image just to help recover the car to the middle line. So a total of 24108 images from the original training dataset.

Next, I flipped every image to double the training dataset. In the training dataset, the car drive counter-clock wise around the track, so the image was left-turn biased. Flipping will generate right-turn images that helps on generalize the steering. Below is an example of the orignal and it's flipped copy.





As mentioned in model architecture section, the input BGR data was first normalized then cropped to focus on the driving track itself.

4. Attempts to Reduce Over-fitting

Dropout was the main thing, but still see some over-fitting. Also tried add L2 regularization, not much difference. I ended up train less number of epochs before over-fitting happen.

5. Model Parameter Tuning

Adam optimizer to 0.0001

dropout keep_prob: 0.4.

epochs : 3

6. The Other track

The mountain track was very challenging. The driving involves changings throttle, breaking in additional to steering left and right. In the track1, I can keep the throttle at the maximum then just steering left or right to keep the car staying in the track. For the mountain however, there were up or down hills that requires throttle changes and breaking. Also there were a lot of sharp turns, this make it very difficult to gather good training data. I haven't done coding for this track yet. But here are some things that I think could make it work:

The model should have 3 outputs instead of 1: throttle, break, and steering.

Change the cropping so to handle the up/down hills.

Tweak the steering angle

Augmentation of the brightness, a lot of shadows