Project-3
Behavior Cloning

Model Architecture and Training Strategy

1. Solution Design Approach

The overall strategy for deriving a model architecture was to drive the car autonomously through the track.

My first step was to use a convolutional neural network model similar to the network we used to train the images in order to identity the images with appropriate labels. I thought this model is appropriate because the project was very similar to traffic sign classifier project and in place of labels we used appropriate measurements.

To combat the overfitting, i modified the model so that it could reduce overfitting. In this model i used dropout to prevent the zero valued node and relu for non linearity.

Then i used 0.2 and 0.5 values for the dropout. The input image shape to the neural network is (160,320,3). After accepting the image with this resolution i used the cropping2D to crop the image of 70 percent from the top and 20 percent from the bottom.

The final step was to run the simulator to see how well the car was driving around track one. There were a few spots where the vehicle fell off the tracks and i used the other datasets to improve behavior in these cases. I also used other two dataset to improve the driving performance of the vehicle. The datasets are:

- Two laps of center lane line dataset
- Dataset from the project resources.

2. Final Model Architecture

The final model architecture(model.py line 90-112) consisted of a convolutional neural network with the following layers and layer sizes of (160,320,3) as input. Used lamba for normalization of the images. After normalizing the images, the images will be cropped using cropping2D of 70 percent from the top and 20 percent from the bottom. Then i used five convolutional2D with strides of (2,2) and filter size of 24,36,48,64 and 64. For non linearity and to reduce overfitting both relu and dropout have been used. Then the model is flatten and fully connected until it reaches the with value 1. Adam optimizer is used as the algorithm since we do not want to tune the learning rates. The training model is split into **validation mode**l of sample by 20 percent.The number of epochs is 5.

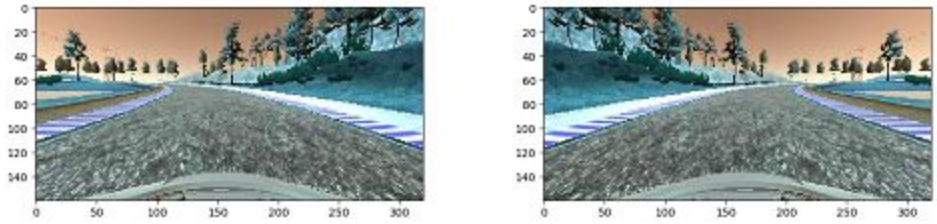
3. Creation of the Training Set and Training Process

This is a 320*160 image.

I then recorded the vehicle recovering from the left side and right sides of the road back to center so that the vehicle would learn to adjust by itself to center when driving autonomously. These images shows what a recover looks like starting from:

This images are center camera image.

To augment the dataset, i also flipped images and angles thinking that this would improve better understanding for the neural network to learn things. For example, here is an image that has then been flipped:



The first picture is the original picture and the second picture is flipped one.

After the collection process, i had 60,000 above samples. I then preprocessed this data by using lamba layer of values 127.5 and subtracting with standard deviation of 1.

I finally randomly shuffled the data set and put 20% of data into validation set.

I used this training data for training the model. The validation set helped determine if the model was over or underfitting. The ideal number of epochs was 5 as evidenced by the python file(model.py 133). I used an adam optimizer so that manually training the learning rate wasn't necessary.