Behavioral Cloning Project Report

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
- Summarize the results with a written report

My project includes the following files:

- clone1.py containing the script to create and train the model
- drive.py for driving the car in autonomous mode
- model_correct.h5 containing a trained convolution neural network
- writeup_report.pdf summarizing the results

I used drive.py provided by Udacity without any modifications for the purpose of this project

My clone1.py file contains the code for training and saving the convolutional neural network. The file shows the pipeline i used for training and validating the model. I clearly commented all the sections of code and how it works.

Model Architecture and Training Strategy:
Reference Architecture - CNN architecture

- My architecture contains 9 layers which includes a normalization layer , 5 convolutional layers and 3 fully connected layers (Line 82 to 115 - clone1.py)
- First Layer is image normalization.
- CNN layers are used for image extraction.
- First 3 convolutional layers are with a 2*2 stride and 5*5 kernel
- Non strided convolutional layer with a 3*3 kernel size in last two layers
- 3 fully connected layers are said to be used for controlling the steering (as mentioned in nvidia blog)

Attempts to reduce overfitting

- The Dropout layers can reduce over fitting
- Multiple data sets are used to train and validate the network.
- The model was tested by running it through the simulator and ensuring that the vehicle could stay on track.

Model Parameter tuning

- My architecture used an adam optimizer so the learning rate was not tuned manually

Training Data

- 2 laps of lane driving trying to be in center of the lane
- 1 lap of driving the track in opposite direction trying to be in center
- In multiple scenarios - i tried to go out to edges and record the recovery back to track
- Re drove the steep curves multiple times to drive better.

Architecture and Training Strategy

Solution Design Approach

- I Initially started without a CNN layer and tested the simulator with just a flatten and Dense layer to test the simulator working in autonomous mode. My car was moving fine but it has lot of error where it started going off the track sometimes. The validation loss was rising instead of decreasing as the epochs increased.
- Then i added the convolutional neural network layer learnt from the tutorial and nvidia.
- At the end after several improvements in the network architecture and training data , the car was moving fine without moving off the track.

Final Model Architecture

- Final model architecture was hyperlinked in the previous section where architecture was discussed.

Creation of Training Set and Process

- To capture good driving behavior, I first recorded two laps on track one using center lane driving.
- I then recorded by driving in opposite way on the same first track.
- Recovery lap where i went off the track and then recorded how to recover from the off track.
- I had around 60,000 data points after all the data collection process.
- I randomly reshuffled the data set and put 25% of the data into a validation data set.
    1. I used 10 epochs for training the model with above mentioned training data.
    2. The validation set made me determine the exact no of epochs to not over or under fit the data.

I included several images in the examples directory related to architecture, center lane driving, recovery driving etc.