Behavioral Cloning SDCND-Term1 Project3

Rubric Specifications:

Required Files:

My project submission includes the following files:

- model.py containing the script to create and train the model
- Behavioral_Cloning.ipynb containing the executed jupyter notebook code
- drive.py for driving the car in autonomous mode
- modelw.h5 containing a trained convolution neural network
- run1.mp4 trial run video on track1
- Project3_report.pdf summarizing the results

Quality of Code

Is the code functional?

I have submitted the functional code of model.py and Behavioral_Cloning.ipynb.

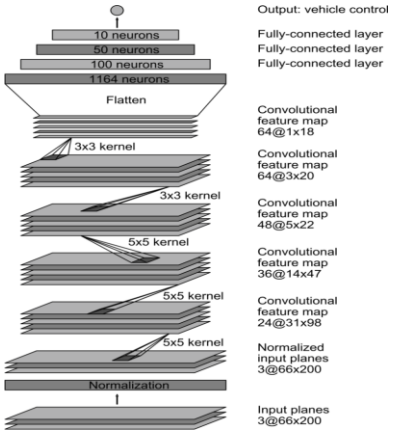
Is the code usable and readable?

The model.py code is organized and commented to make easy understanding and additionally can be verified with the Behavioral_Cloning.ipynb file. Particulary I did not use the generator because I already had about 25600 images and running a generator showed me errors which I could not rectify even after multiple hours of debugging and I switched to increasing the number of data samples and training without a generator and produced the modelw.h5 file.

Model Architecture and Training Strategy

Has an appropriate model architecture been employed for the task?

I used a NVIDIA network architecture with five convolutions with first three of kernel sizes 5x5 and last two with kernals 3x3 and the depths range from 24 to 64. The network has fully connected layers with neurons reducing from 1164 to 1 in the output as similar to a regression model.Non- Linearity is introduced with Relu Activations and the normalization is carried out with the Lambda layer to normalize the pixel densities in the image.The below figure gives the architecture layout:



Has an attempt been made to reduce overfitting of the model?

The 20% of the total sample size has been split as the validation data and the data has been shuffled. Finally there are about 20500 training images and 5000 validation images.

I used dropout in the second layer of the neural network about 50% keep probability to reduce the overfitting of data.

Have the model parameters been tuned appropriately?

I employed Adam optimizer, so there is no necessity for tuning the learning rate manually.

Is the training data chosen appropriately?

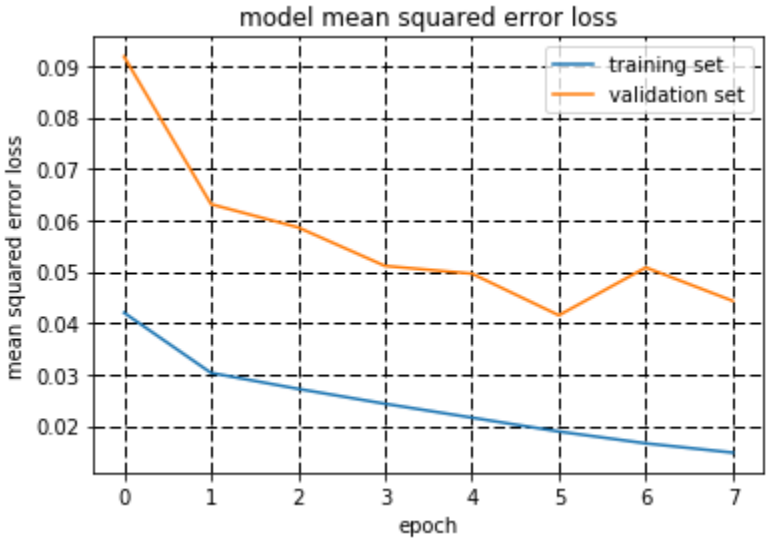
The training data was done by doing a center lane driving in two forward and two reverse laps and further recovery data is taken at almost at each of the individual turns. So, the data almost had a good distribution of steering angles for center lane, left and right turns.

Architecture and Training Documentation

Is the solution design documented?

I initially tried with a simpler LeNet architecture and there is no successful attempt in running the car. I turned towards the medium to check NVIDIA architecture seemed successful for almost everyone and I tried a NVIDIA network architecture and found relatively better results. After some attempts and suggestions ,I varied the strides to 2 and stride=1 for last two convolutions for a quicker and better feature extraction. This worked providing a better result. The chosen architecture is described in detail in the previous section about the neural network.

The neural network performance shown by the Mean Squared Error plot



The network performance was a good one, but if had try this network for the second track , the training has to be adjusted and it might not be robust with this progression.

Is the model architecture documented?

The below is the representation of the detailed neural network model architecture trained:

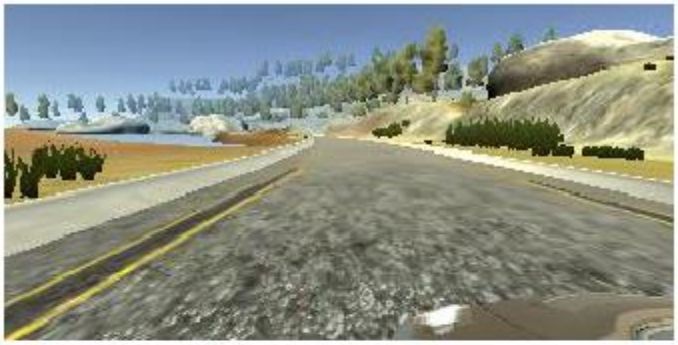
Lambda Layer	Image normalization
Cropping Layer	Cropping the region of interest between 50 x 20 pixels
Convolution 1	Kernal=5x5, Depth= 24, Stride/Subsample=2
Relu Activation	
Convolution 2	Kernal=5x5, Depth= 36, Stride/Subsample=2
Relu Activation	
Convolution 3	Kernal=5x5, Depth= 48, Stride/Subsample=2
Relu Activation	
Convolution 4	Kernal=3x3, Depth= 64, Stride/Subsample=1
Relu Activation	
Convolution 5	Kernal=3x3, Depth= 64, Stride/Subsample=1
Relu Activation	
Flatten	
Fully Connected Input Layer	1064 neurons
Fully Connected Layer1	100 neurons
Dropout	
Fully Connected Layer2	50 neurons
Fully Connected layer3	10 neurons
Output Layer	

Is the creation of the training dataset and training process documented?

The dataset was created with center lane driving for two forward and two reverse laps and made some recovery data for each turn in the track 1. The total sample size of the data is about 25,600. The corresponding steering angle correction for the left and right camera images is tuned from 0.2 to 0.25 and finally the corresponding steering angles are appended. The images are cropped between 50 x 20 pixel to extract the region of interest for feature extraction by convolution. The representation of a sample image from each camera are shown below

Center

Left camera



Right camera