# Behavioral Cloning Project Report Write up

Files Submitted & Code Quality

1. Submission includes all required files and can be used to run the simulator in autonomous mode

<u>My project includes the following files:</u>

A. model.py containing the script to create and train the model
B. drive.py for driving the car in autonomous mode
C. model2.h5 containing a trained convolution neural network
D. Behavioral Cloning Project Report Write up.pdf summarizing the results

**<u>Note: I faced lot of issues with my environment while working on this project. I wasn't even able to submit this project from my workspace. Hence I have copied all project files to:</u>**

**<u>https://1drv.ms/u/s!AljMvHVq6qA2gUZGfexbMKpDzHXL?e=AzPlcs</u>**

2. Submission includes functional code

I was really bad driving the car using the simulator. After multiple tries I decided to use the Udacity provided sample data. Using my drive.py file, the car can be driven autonomously around the track by executing:

**python drive.py model.h5**

3. Submission code is usable and readable

The model.py file contains the code for training and saving the convolution neural network. The file shows the pipeline I used for training and validating the model, and it contains comments to explain how the code works.

Model Architecture and Training Strategy

1. An appropriate model architecture has been employed

My model consists of a convolution neural network with 3x3 filter sizes and depths between 32 and 128. In this model to introduce nonlinearity and I have used RELU layers and used Keras lamda layer for data normalized.

2. Attempts to reduce overfitting in the model

In order to reduce overfitting, I have used dropout layers.  I have tested the model by running it through the simulator to make sure that the car could stay on the track. This made so happy because when I driving the simulator I would crash the car with seconds after starting. I couldn't believe that my code could actually complete one full lap without crashing. Considering this as one of my life time achievement ☺.

3. Model parameter tuning

I have used adam optimizer to prevent the learning rate not tuned manually.

4. Appropriate training data

From my miserable crash I have understood the challenges (I hate curves) and hence choose to use a combination of center lane driving and left and right recovery( the last curves which turns left and right immediately is my worst enemy so far).

Model Architecture and Training Strategy

1. Solution Design Approach

My strategy was to driving any car on any terrain.

My first step was to use NVIDIA for convolution neural network to help my car drive through all kinds of terrain. As I progressed coding and testing phase I realized that track complexity was far lesser and hence I reduce the model complexity further by reducing convolutional and dense layers. I was able to achieve a satisfactory result by using 1 convultional layer and followed by 1 dense layers.

In order to gauge how well the model was working, I split my image and steering angle data into a training and validation set. I found that model had a low mean squared error on the training set but a high mean squared error on the validation set. This implied that the model was overfitting.

To combat the overfitting, dropout layer and max pooling layer was used. Which  was followed by convolution layer and I reduced number of epochs to 2.

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 car fell off the track to improve the driving behavior in these cases, I have added left and right side images with offset driving angle, I have cropped the image to include only required details so that noise can be reduced, I flipped images so that additional details are available while training the vehicle.

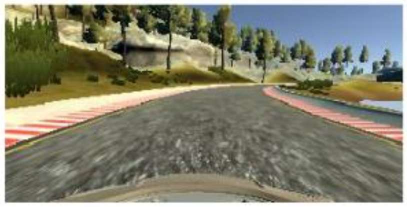
At the end of the process, the vehicle is able to drive autonomously around both the tracks without leaving the road.

2. Final Model Architecture

The final model architecture consisted of a convolution neural network with the following layers and layer sizes 5X5 filter of depth 15


3. Creation of the Training Set & Training Process

To train my model I used dataset provided by Udacity. Some of the samples from original dataset are

| Camera | Image 1 | Image 2 |
| --- | --- | --- |
| Center Camera Images |  |  |
| Left Camera Images |  |  |
| Right Camera Images |  |  |

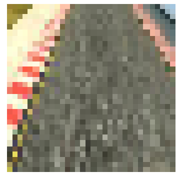
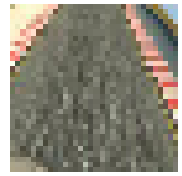
To remove redundant details I cropped image's 80px from top and 20px from bottom for all the images that is center camera images, left camera images and right camera images (right and left camera images are added by adding respective steering angle offset), images after cropping are

| Cropped Center Camera Image | Cropped Left Camera Image | Cropped Right Camera Image |
| --- | --- | --- |
|  |  |  |

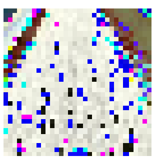
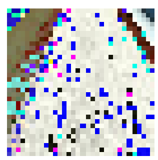
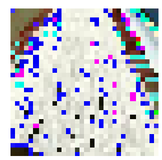
To augment the data set, I flipped center camera images thinking that this would provide more number of samples to be trained. For example, here is an image that has then been flipped:

| Cropped Center Camera Image | Cropped Flipped Center Camera Image |
| --- | --- |
|  |  |

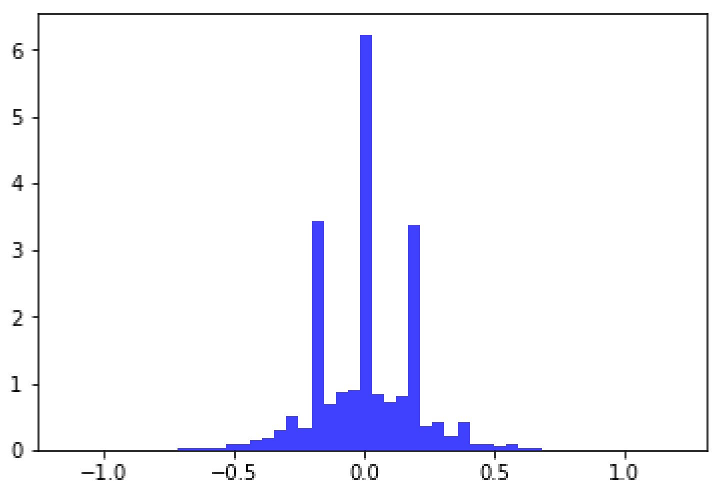
After cropping and flipping images I resized all the images to 32 X 32.

| Resized Center Camera Image | Resized Left Camera Image | Resized Right Camera Image |
|---|---|---|
|  |  |  |

After cropping image I applied normalization on images and these normalized images are passed to the network as input

| Normalized Center Camera Image | Normalized Left Camera Image | Normalized Right Camera Image |
|---|---|---|
|  |  |  |

After the collection process and preprocess, I had 25712 number of training data points. The steering angle distibution for the data points is




I used this training data for training the model. The validation set helped determine if the model was over or under fitting. The ideal number of epochs was 2 as evidenced by running the model several times. I used an adam optimizer so that manually training the learning rate wasn't necessary.

Below is the final video of my car driving:

run2.mp4

Note: I faced lot of issues with my environment while  working on this project. I wasn’t even able to submit this project from my workspace. Hence I have copied all project files to:

https://1drv.ms/u/s!AljMvHVq6qA2gUZGfexbMKpDzHXL?e=AzPIcs