Behavioral Cloning Project

The objective of the project is to train a model to drive a car autonomously on a simulated track. The model is trained to predict an appropriate steering angle to an autonomous vehicle.

The project implementation consisted of the following phases:

1) Data Collection
2) Data Pre-Processing
3) Model Training
4) Model Testing

Data Collection

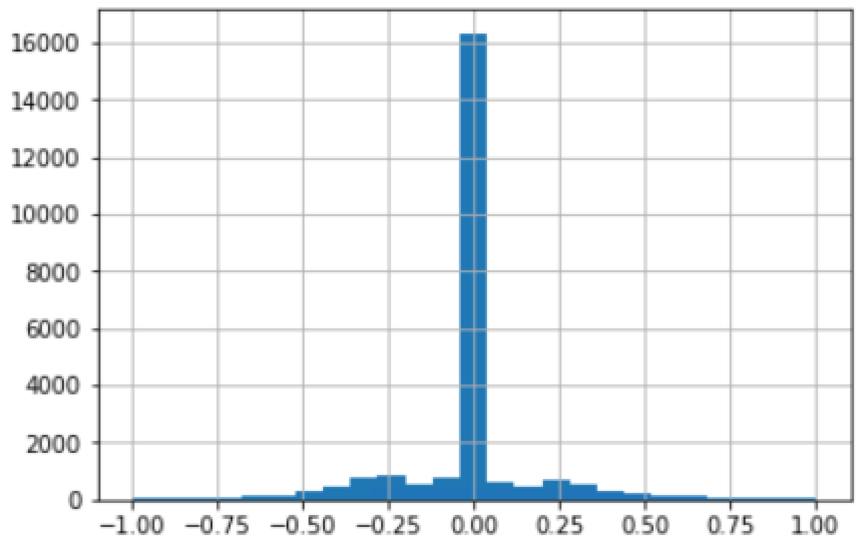
The project repository provided by Udacity contains a dataset of 24K images. The guidelines provided in the project and by prior students had the following recommendation with regards to training data:

- recording own training data (a minimum of 45K images)
- collect data by driving in both the direction on the training track

I collected approx. 72K images driving the track for 5 laps in the forward direction and 3 laps in the reverse direction. The Udacity provided 24K images where kept aside to be used for model testing/validation.

Data Pre-Processing

The histogram of the collected/recorded data was then visualized to understand the distribution of the steering angle to determine any pre-processing/data augmentation steps.



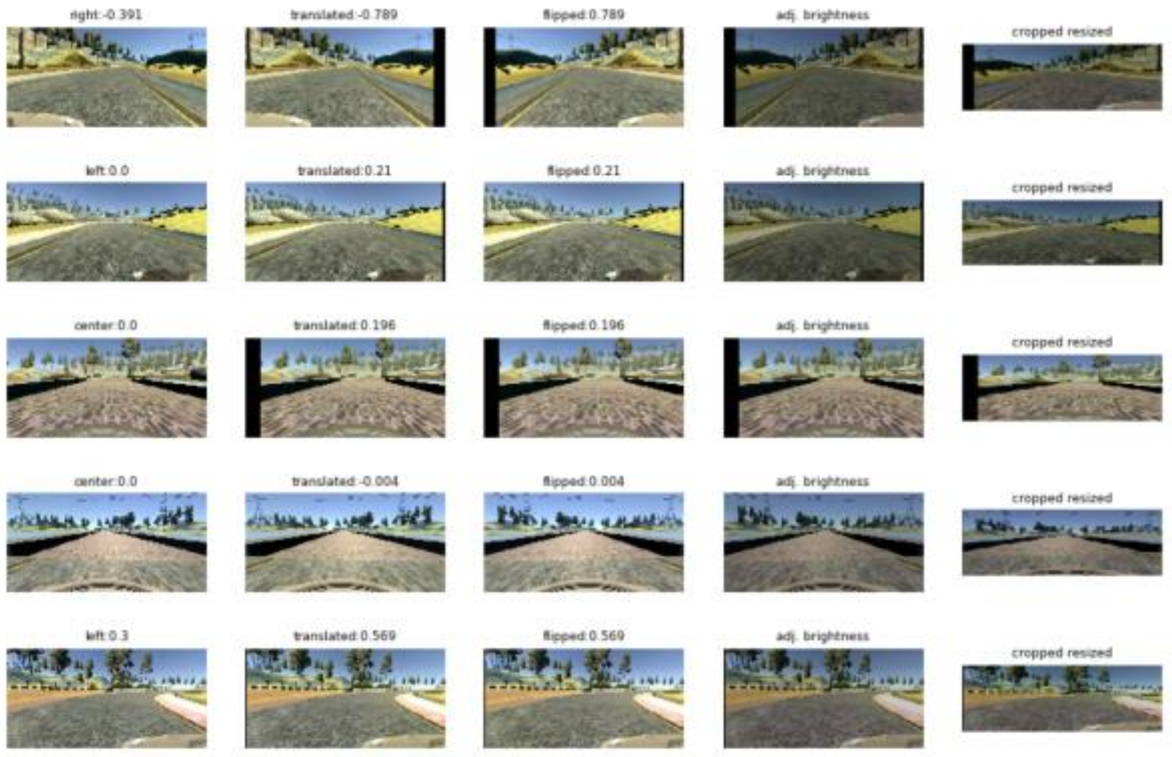
As seen from the above plot a significant majority of the samples are recorded with a steering angle of zero. To balance this, we will shift the images horizontally (randomly to the left or right) to simulate the effect of car being at different positions on the track. Additionally, the simulator records images from the left/center/right cameras. Since we can use only one randomly picked image - we adjust steering angle for the left image by +0.25 and the right image by -0.25, to compensate for the camera offsets

The track used for training has a lot left turns. We balance this to some extent by creating recordings while driving the in the opposite direction. Additionally, we also flip the images about the vertical axis and invert the sign of steering angle for the flipped image.

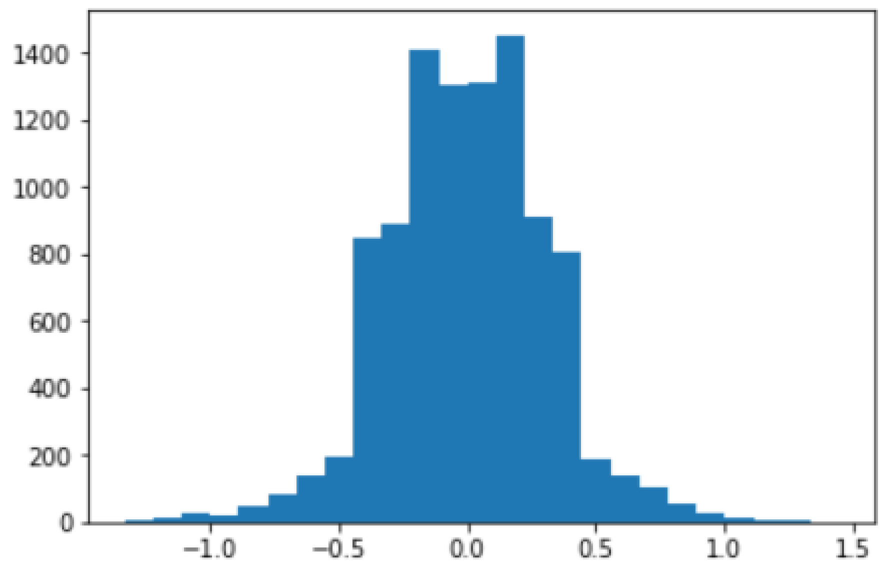
To account for various environmental conditions such as shadows, darker/lighter roads, day/night we generate images with different brightness by converting image to HSV channel and randomly scaling the V channel by a random number between 0.25 - 1.25

Finally, we crop the image to remove the horizon and the car's hood and then resize the cropped image to be 200 x 66 to match the size for the NVIDIA and COMMA.AI networks

The below image show the data pre-processing/augmentation image processing pipeline in action.



The below histogram shows the updated distribution of steering angle post the data pre-processing/image augmentation on the recorded data-set.



Model Training

A total of 3 models where trained on the recorded dataset.

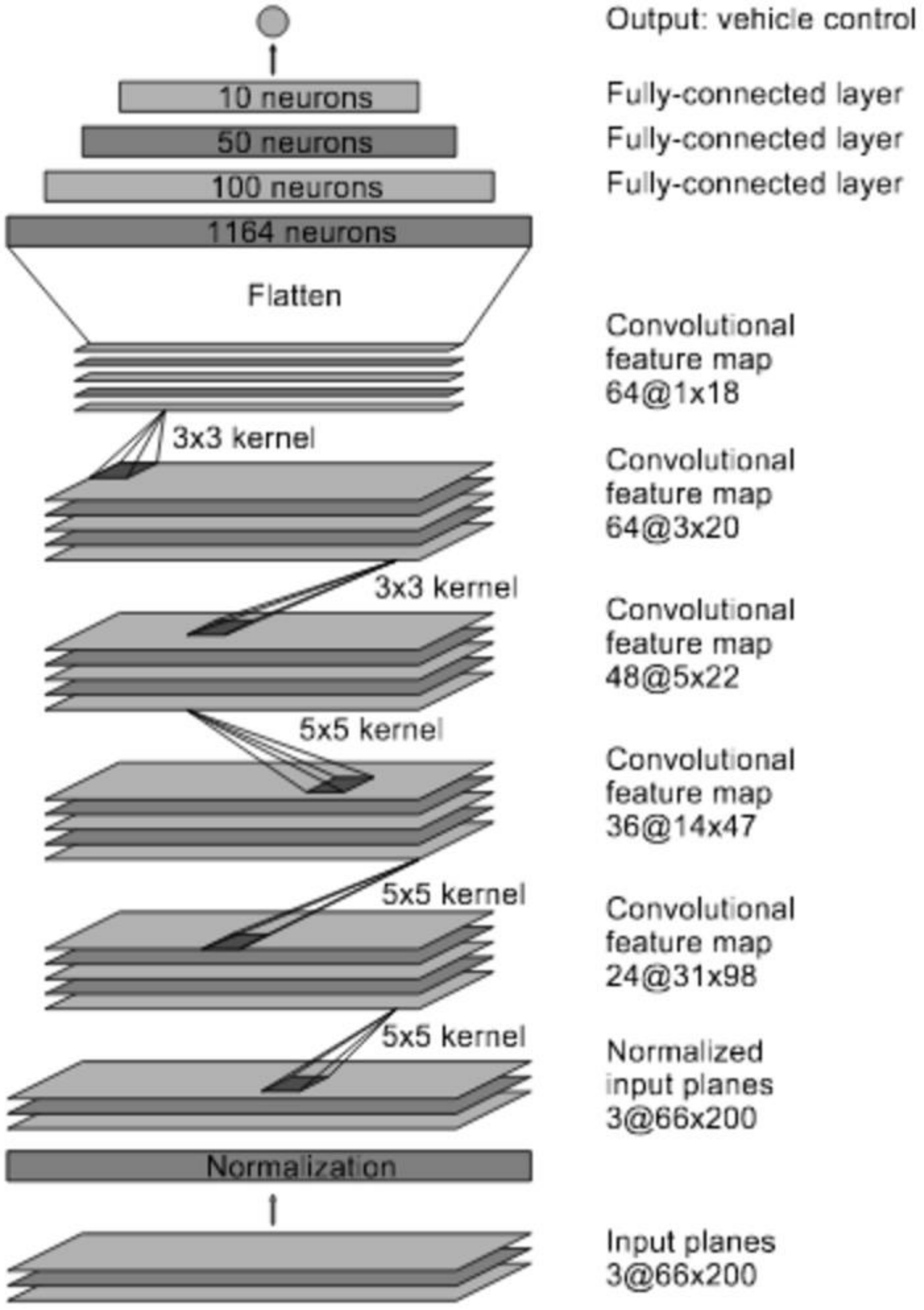
- NVIDIA model with drop-out at each fully connected layer
- COMMA.AI model
- A hybrid model with the basic architecture as an NVIDIA model however using activations and drops similar to the COMMA.AI model

The most consistent and stable results were achieved with the NVIDIA model with drop-out after each fully connected layer and hence was used for the final testing. The final model consisted of 9 layers, including a normalization layer, 5 convolutional layers and 3 fully connected layers with drop-out (using a keep probability of 0.2) after each fully connected layer.

Image was normalized in the first layer. Followed by 3 Convolution layers with 2x2 strides and a 5x5 kernel, and non-strided convolution with 3x3 kernel size in the last two convolutional layers. The convolutional layers were followed by three fully connected layers which then outputs the steering angle. Overfitting was reduced by using aggressive Dropout (0.2) after each fully-connected layers and L2 Regularization (0.001) on the first layer.

An Adam optimizer was used for optimization. In addition, checkpoint and early stop mechanisms were used during training to choose best training model by monitoring the validation loss and stopping the training if the loss does not reduce in three consecutive epochs.

The final model can be visualized as below:



Since I had approx. 72K training images and 24K validation images, to make the training process more tractable a batch generator for both training and validation data. For each epoch, approx. 18K images (randomly selected between left/right/camera for each sample) where used for training – thus all the data was used for training, however at different epochs.

Model Testing

The trained model was tested on Track 1. The car could drive successfully for more than 30+ minutes (after which I stopped the simulator) without leaving the road. The included video.mp4 shows the car driving for approx. 3 laps.

The final video show the car consistently, correcting slightly more the left of the track (after leaving the bridge), correcting slightly to be on the road (but outside the yellow lanes) for a few seconds a then centering. I believe this bias on that section of track is caused due to my recording when driving the car in opposite direction – this part of the track gave me consistent trouble with the car often steering off the center road and me having to correct it due to the awkward nature of the curves when driving in reverse direction.

Future Improvements

As future improvement to make sure that the car stay more consistently within the center of the road, I would like to create a new recording set that demonstrate the good driving behavior by not creating any recordings for reverse direction (which tends to cause more issues by biasing the data with bad driving techniques) and relying the image flipping technique for data augmentation. Additionally I would like to include having random shadows in the data pre-processing/augmentation pipeline which the car encounters on Track 2.

Acknowledgements

While working on the solutions I used the following references:

- NVIDIA paper
- COMMA.AI paper
- Vivek Yadav's blog