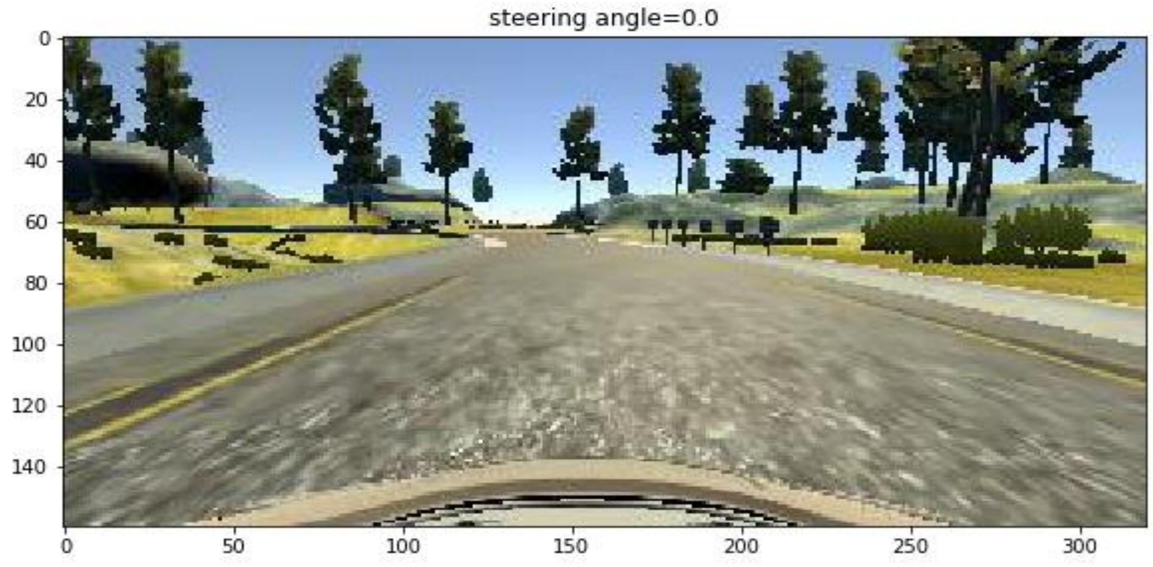
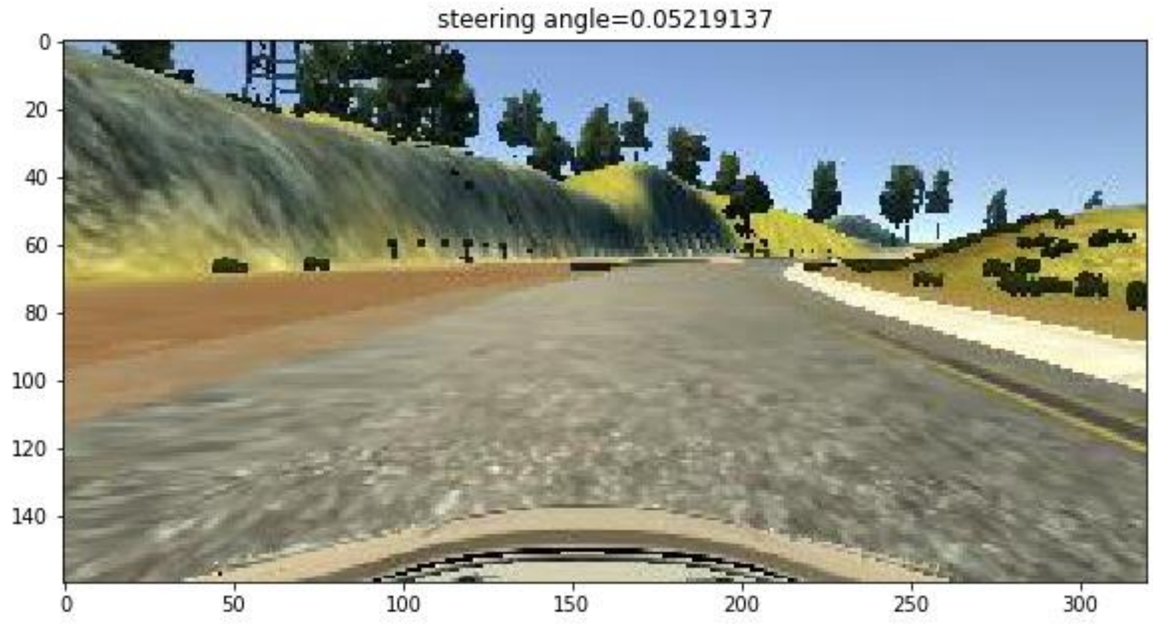
PROJECT 3- Behaviour Cloning

The goals / steps of this project are the following:
1) Use the simulator to collect data of good driving behavior
2) Build, a convolution neural network in Keras that predicts steering angles from images
3) Train and validate the model with a training and validation set
4) Test that the model successfully drives around track one without leaving the road
5) Summarize the results with a written report.

Data for training the simulator

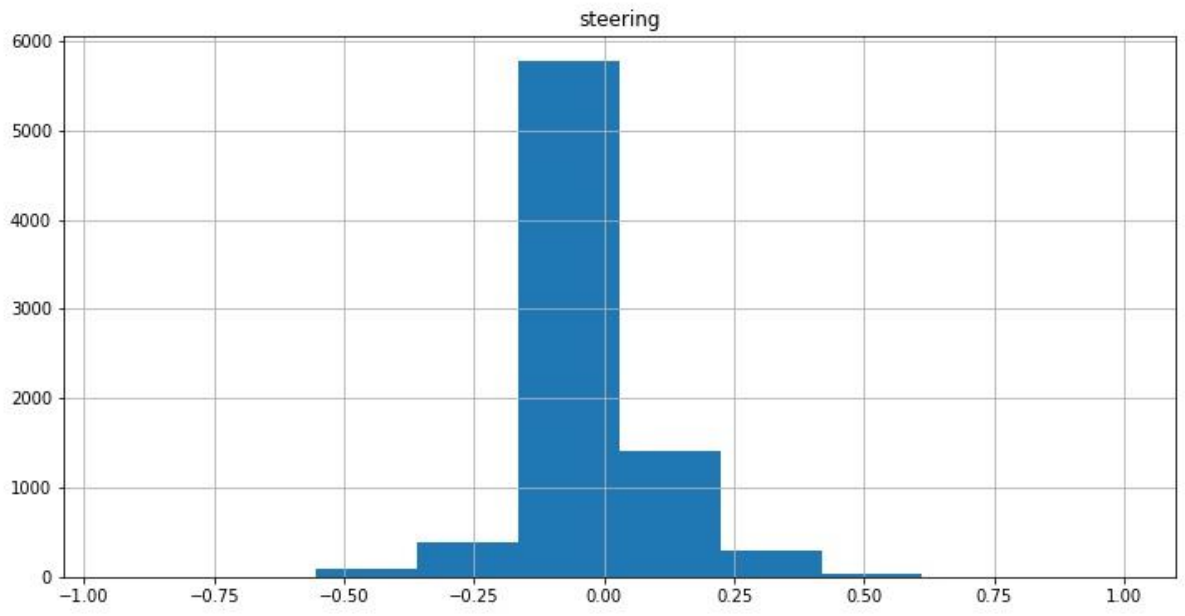
I used the dataset provided by Udacity. Let us visualize some example images with steering angle.





We can observe that the top 30 pixels of the image if scenery and bottom 20 pixels is covered by the hood of the car. Therefore, this can be cropped.

Distribution of Steering Angle:



From 8037 examples, we can observe that the steering angle is skewed to 0 and negative.

Data Augmentation

I use the following data augmentation methods to preprocess for training.

1) Flipping the image and the measurement.
2) Cropping and Reshaping
3) Randomly filtering the data based on steering angle.

Flipping the image and the measurements will make the model robust to both left and right turns if the track is biased to either side.

Cropping and reshaping will help reduce number of inputs and help train faster.

Randomly filtering the data by steering angle can make the model generalized. I allowed 10% of the steering angles whose magnitude is less than 0.05. This made the car go around the track.

These techniques were inspired from the discussion forum, Nvidia paper and it works.

Model Selection

I used the Nvidia End to End architecture which has 5 convolutional layers with relu activation for nonlinearity, The input is normalized by the lambda layer from keras. Then there are 4 full connected layers.

Optimizer used is “Adam” and loss function used is ‘mse’ Mean Squared Error. This way learning rate is not tuned by the user.

Next I fed the CNN with randomly sampled batch sizes of 64 using the generator function.

Model Performance

After training for 17 epochs, I got a good training loss value.

I stopped to test the performance round the track. It appears to successfully go around the track.

Improvements

Other augmentation techniques can be explored. Dropouts can be used to reduce overfitting of the model thereby making it robust.

The next iteration would be to train and test in track 2. The current model does not successfully go around track 2.

References

Nvidia End-End paper, Q&A session, discussion forum.