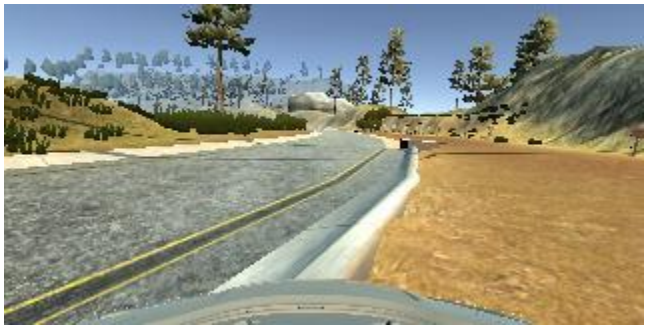
CarND-Behavioral-Cloning-P3

Data Collection:

Track 1

- 2 Laps around the track counterclockwise
  - 
- 1 lap around the track clockwise
  - 
- 2 areas where recovery from going off road:
  - 

Track 2:

- Did some segments clockwise
  - 

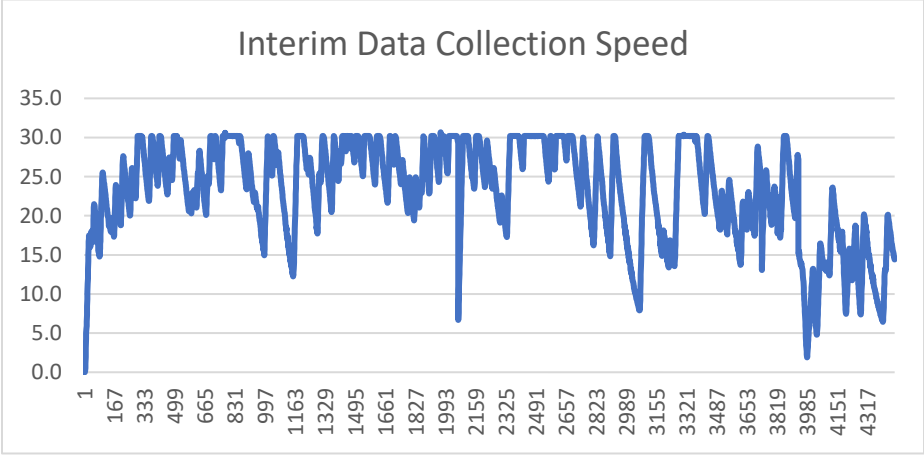
- Did some segments counterclockwise
  - 

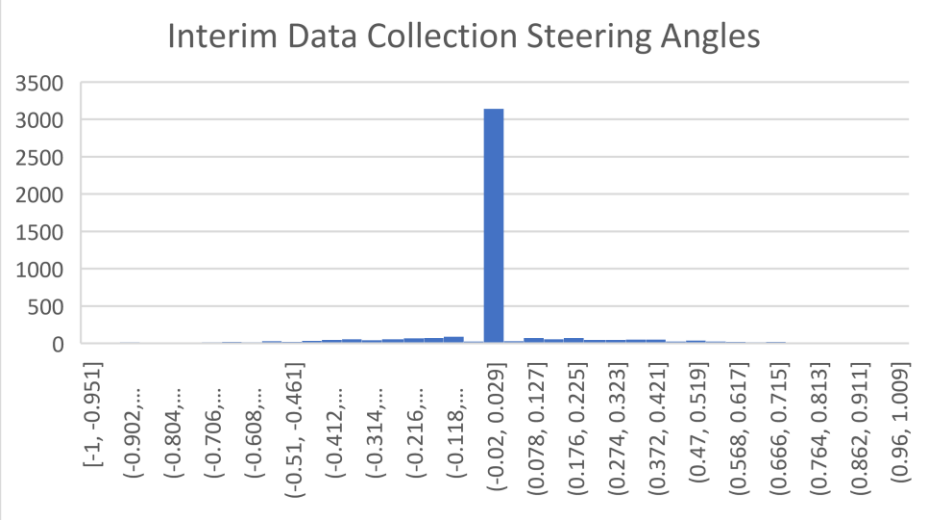
Driving in the simulator was quite different to a video game. Took some time to get used to.

Did multiple data collection rounds to get data somewhat at the center:

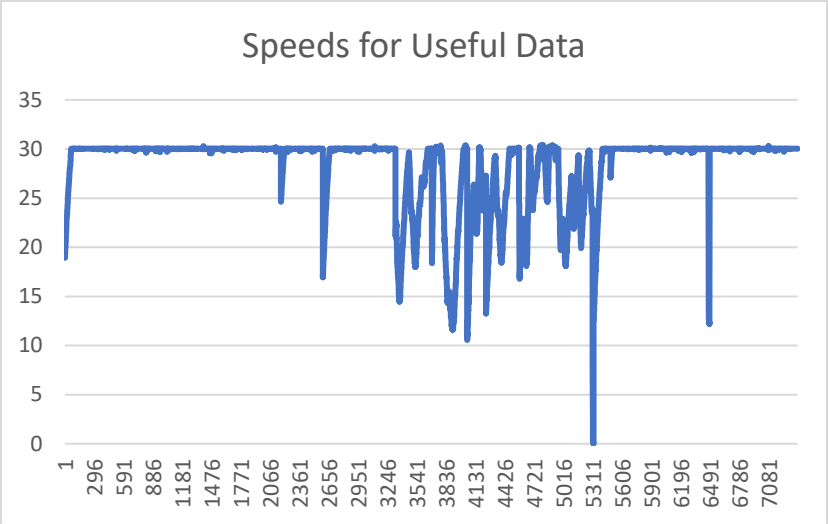
- Determined that cruise control offered the best way to collect data since we are not optimizing the steering angle along with speed.

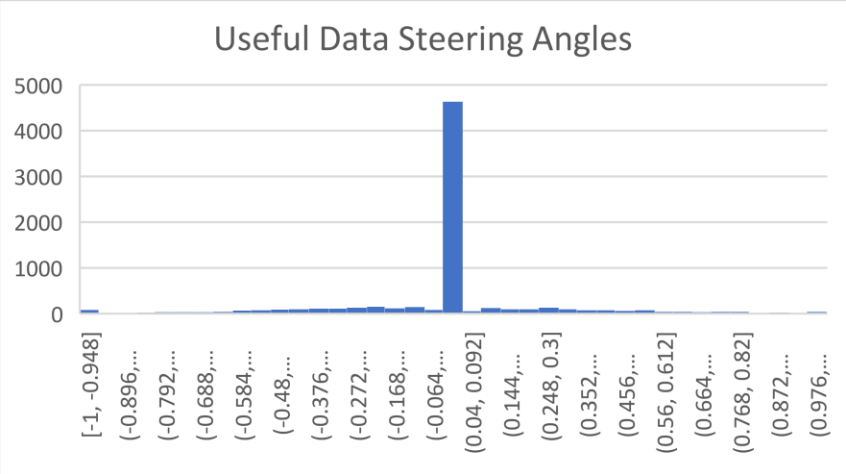
Run with Data that was not useful too much variation in speed and a little erratic on steering:





Run with Data that was useful with mostly constant speed and more controlled steering: 7370 total





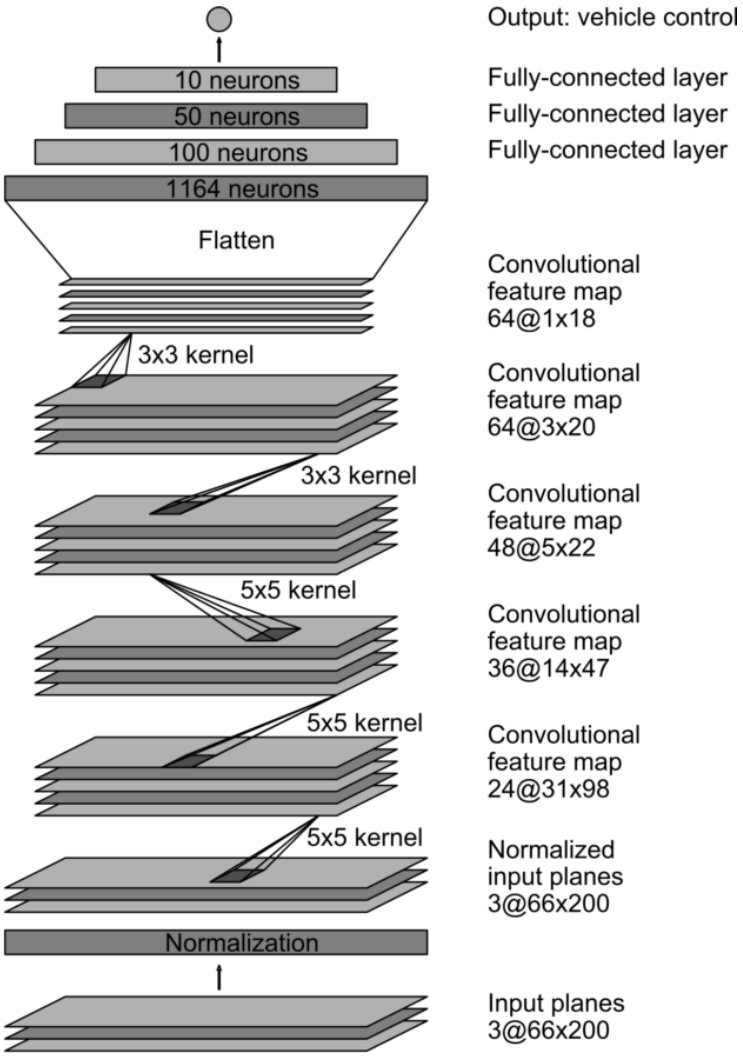
Processing the input images:

- Read the driving log.csv file that is in the workspace under Constant_Data folder
- Extract the image names from the first 3 columns:
    - Center, left and right images
- Extract steering angles from column 4
- Add/ subtract correction factors to the steering angles
    - Tried 0.2, 0.1 and 0.05: ended up selecting 0.05
- Flip images and generate a new set of augmented images:
    - np.fliplr
- Split the image set to training and validation images: 20% validation and 80% training using the test_train_split function
- Read in the images, steering angles and store them as shuffled batched numpy arrays using the generator function:

- Values tried 128,64,32
- Selected 32 for batch size since it gave best results within a few epochs and did not affect the time to train adversely

Model architecture

- Started off with a sample flatten layer to see if the model can execute
- Model could execute and drive on autonomous mode but could not complete the track
- Tried Lenet-5 with Batch size 128 and 25 epochs and batch size 64 with 10 epochs: could not complete track 1
- Tried NVIDIA model with dropouts added for each of the fully connected layers: Car did not drive around the track 1
- Tried nvidia model: https://developer.nvidia.com/blog/deep-learning-self-driving-cars/
    - The validation loss stabilized within 5 epochs and the car could drive around the track 1with few places where it goes towards the edge



- Tuning Parameters and running the model:
    - Batch size: 32 for best results within a few epochs

- Epochs 5: validation loss for the model stabilized within 5 epochs, past that it was pretty much constant
- Used the workspace to run the model training python model.py

Capturing Video:

- Run the following command on the workspace:
  - python drive.py model.h5 run3
- Open the simulator and run in autonomous mode and allow the car to drive around the track once: track 1
- Come back to the terminal, stop the script using CTRL + C
- Install ffmpeg:
  - Sudo apt -get update
  - sudo apt-get install ffmpeg
- Run the command to create video:
  - python video.py run1

Improvements:

- Get more data with the CAR in the center
- Do more switchbacks: left turns followed by immediate right turns
- Try to optimize both speed and steering to adapt better to driving styles
- Improve drive.py to have a second-degree polynomial fit and improve turns and path prediction
- Do additional color transforms for shadows and slopes: most of the first track was uniformly lit