Self_driving_car

March 18, 2019

In [1]:
```
# Credits: https://github.com/SullyChen/Autopilot-TensorFlow
# Research paper: End to End Learning for Self-Driving Cars by Nvidia. [https://arxiv.

# NVidia dataset: 72 hrs of video => 72*60*60*30 = 7,776,000 images
# Nvidia blog: https://devblogs.nvidia.com/deep-learning-self-driving-cars/


# Our Dataset: https://github.com/SullyChen/Autopilot-TensorFlow [https://drive.google
# Size: 25 minutes = 25*60*30 = 45,000 images ~ 2.3 GB


# If you want to try on a slightly large dataset: 70 minutes of data ~ 223GB
# Refer: https://medium.com/udacity/open-sourcing-223gb-of-mountain-view-driving-data-
# Format: Image, latitude, longitude, gear, brake, throttle, steering angles and speed



# Additional Installations:
# pip3 install h5py


# AWS: https://aws.amazon.com/blogs/machine-learning/get-started-with-deep-learning-us

# Youtube:https://www.youtube.com/watch?v=qhUvQiKec2U
# Further reading and extensions: https://medium.com/udacity/teaching-a-machine-to-ste
# More data: https://medium.com/udacity/open-sourcing-223gb-of-mountain-view-driving-d
```

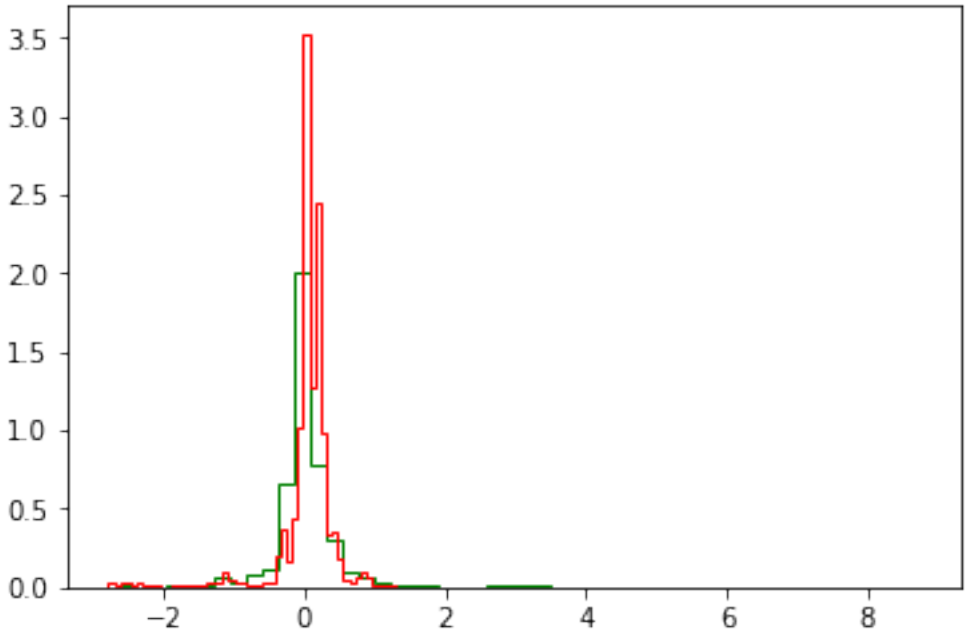
In [6]:
```
# read images and steering angles from driving_dataset folder

from __future__ import division

import os
import numpy as np
import random

from scipy import pi
from itertools import islice
```

```
        DATA_FOLDER = './driving_dataset/' # change this to your folder
        TRAIN_FILE = os.path.join(DATA_FOLDER, 'data.txt')

        LIMIT = None
        split =0.8
        X = []
        y = []
        with open(TRAIN_FILE) as fp:
            for line in islice(fp, LIMIT):
                path, angle = line.strip().split()
                full_path = os.path.join(DATA_FOLDER, path)
                X.append(full_path)

                # converting angle from degrees to radians
                y.append(float(angle) * pi / 180 )


        y = np.array(y)
        print("Completed processing data.txt")

        split_index = int(len(y)*0.8)

        train_y = y[:split_index]
        test_y = y[split_index:]
```

```
Completed processing data.txt
```

```
In [7]: import numpy;

        # PDF of train and test 'y' values.
        import matplotlib.pyplot as plt
        plt.hist(train_y, bins=50, normed=1, color='green', histtype ='step');
        plt.hist(test_y, bins=50, normed=1, color='red', histtype ='step');
        plt.show()
```

```
c:\users\dell\appdata\local\programs\python\python36\lib\site-packages\matplotlib\axes\_axes.py
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' i
  alternative="'density'", removal="3.1")
```



In [8]:
```
#Model 0: Base line Model: y_test_pred = mean(y_train_i)
train_mean_y = np.mean(train_y)

print('Test_MSE(MEAN):%f' % np.mean(np.square(test_y-train_mean_y)) )

print('Test_MSE(ZERO):%f' % np.mean(np.square(test_y-0.0)) )
```

```
Test_MSE(MEAN):0.191127
Test_MSE(ZERO):0.190891
```