

Self-Driving-Car

January 27, 2022

```
[2]: import os
from itertools import islice
from scipy import pi
import numpy as np
DATA_FOLDER='Documents/Autopilot-TensorFlow-master/driving_dataset/'
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)
        y.append(float(angle)*pi/100)
y=np.array(y)
print("Successful")
```

Successful

```
[3]: #split the dataset
split_index=int(len(y)*0.8)
```

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

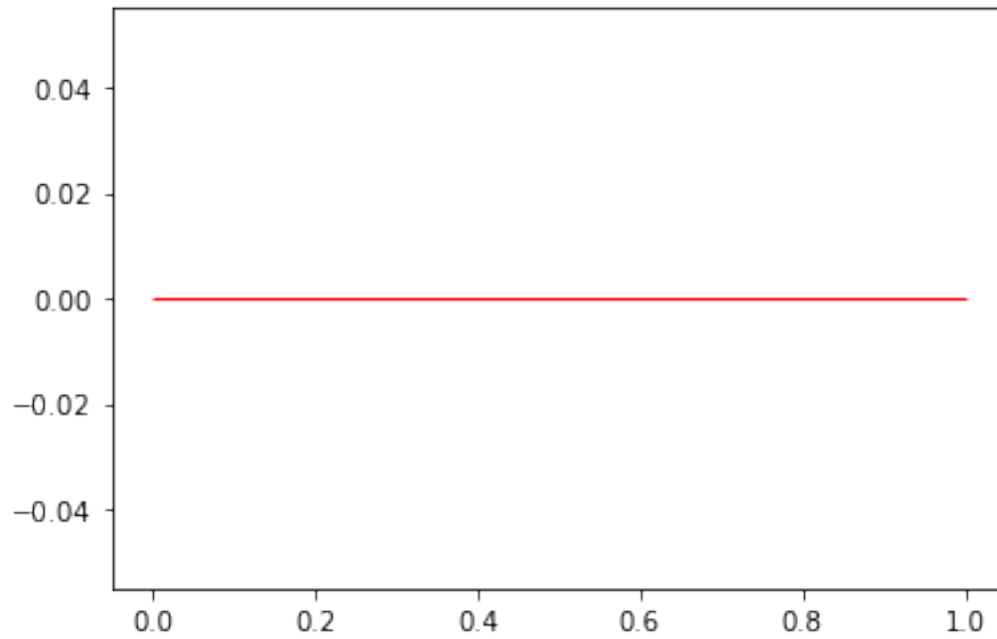
```
[5]: test_y=y[split_index:1]
```

```
[11]: import matplotlib.pyplot as plt
plt.hist(train_y,bins=50,color="blue",histtype='step')
```

```
[11]: (array([5.200e+01, 1.600e+01, 2.900e+01, 1.270e+02, 9.400e+01, 1.770e+02,
4.920e+02, 2.550e+02, 6.840e+02, 9.270e+02, 5.439e+03, 1.660e+04,
6.513e+03, 2.538e+03, 7.510e+02, 5.250e+02, 3.090e+02, 1.640e+02,
1.400e+02, 7.600e+01, 2.400e+01, 2.700e+01, 3.300e+01, 4.300e+01,
5.100e+01, 4.700e+01, 5.400e+01, 3.600e+01, 3.800e+01, 1.400e+01,
8.000e+00, 1.000e+01, 1.000e+00, 2.000e+00, 2.000e+00, 2.000e+00,
2.000e+00, 1.000e+00, 0.000e+00, 2.000e+00, 1.000e+00, 1.000e+00,
```

```
[12]: (array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,  
            0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),  
      array([0. , 0.02, 0.04, 0.06, 0.08, 0.1 , 0.12, 0.14, 0.16, 0.18, 0.2 ,  
            0.22, 0.24, 0.26, 0.28, 0.3 , 0.32, 0.34, 0.36, 0.38, 0.4 , 0.42,  
            0.44, 0.46, 0.48, 0.5 , 0.52, 0.54, 0.56, 0.58, 0.6 , 0.62, 0.64,  
            0.66, 0.68, 0.7 , 0.72, 0.74, 0.76, 0.78, 0.8 , 0.82, 0.84, 0.86,  
            0.88, 0.9 , 0.92, 0.94, 0.96, 0.98, 1. ]),
```

```
[<matplotlib.patches.Polygon at 0x1d57fee52b0>])
```



```
[13]: train_mean_y=np.mean(train_y)
```

```
[16]: np.mean(np.square(test_y-train_mean_y))
```

```
[16]: nan
```

```
[17]: np.mean(np.square(test_y-0.0))
```

```
[17]: nan
```

```
[ ]:
```