In [1]:

```
# Credits: https://github.com/SullyChen/Autopilot-TensorFlow
# Research paper: End to End Learning for Self-Driving Cars by Nvidia. [https://arxiv.org/pdf/1604.07316.pdf]

# 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.com/file/d/0B-KJCaaF7elleG1RbzVPZWV4Tlk/view]
# 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-f6b5593fbfa5
# 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-using-the-aws-deep-learning-ami/

# Youtube:https://www.youtube.com/watch?v=qhUvQiKec2U
# Further reading and extensions: https://medium.com/udacity/teaching-a-machine-to-steer-a-car-d73217f2492c
# More data: https://medium.com/udacity/open-sourcing-223gb-of-mountain-view-driving-data-f6b5593fbfa5
```

In [2]:

```
import scipy.misc
import random

xs = []
ys = []

#points to the end of the last batch
train_batch_pointer = 0
val_batch_pointer = 0

#read data.txt
with open("driving_dataset/data.txt") as f:
    for line in f:
        xs.append("driving_dataset/" + line.split()[0])
        #the paper by Nvidia uses the inverse of the turning radius,
        #but steering wheel angle is proportional to the inverse of turning radius
        #so the steering wheel angle in radians is used as the output
        ys.append(float(line.split()[1]) * scipy.pi / 180)

#get number of images
num_images = len(xs)


train_xs = xs[:int(len(xs) * 0.7)]
train_ys = ys[:int(len(xs) * 0.7)]

val_xs = xs[-int(len(xs) * 0.3):]
val_ys = ys[-int(len(xs) * 0.3):]

num_train_images = len(train_xs)
num_val_images = len(val_xs)

def LoadTrainBatch(batch_size):
    global train_batch_pointer
    x_out = []
    y_out = []
    for i in range(0, batch_size):
        x_out.append(scipy.misc.imresize(scipy.misc.imread(train_xs[(train_batch_pointer + i) % num_tra
```

```
        x_out.append(scipy.misc.imresize(scipy.misc.imread(train_xs[(train_batch_pointer + i) % num_tra
in_images])[-150:], [66, 200]) / 255.0)
        y_out.append([train_ys[(train_batch_pointer + i) % num_train_images]])
    train_batch_pointer += batch_size
    return x_out, y_out

def LoadValBatch(batch_size):
    global val_batch_pointer
    x_out = []
    y_out = []
    for i in range(0, batch_size):
        x_out.append(scipy.misc.imresize(scipy.misc.imread(val_xs[(val_batch_pointer + i) % num_val_ima
ges])[-150:], [66, 200]) / 255.0)
        y_out.append([val_ys[(val_batch_pointer + i) % num_val_images]])
    val_batch_pointer += batch_size
    return x_out, y_out
```

In [3]:

```
import tensorflow as tf
import scipy

def weight_variable(shape):
  initial = tf.truncated_normal(shape, stddev=0.1)
  return tf.Variable(initial)

def bias_variable(shape):
  initial = tf.constant(0.1, shape=shape)
  return tf.Variable(initial)

def conv2d(x, W, stride):
  return tf.nn.conv2d(x, W, strides=[1, stride, stride, 1], padding='VALID')

x = tf.placeholder(tf.float32, shape=[None, 66, 200, 3])
y_ = tf.placeholder(tf.float32, shape=[None, 1])

x_image = x

#first convolutional layer
W_conv1 = weight_variable([5, 5, 3, 24])
b_conv1 = bias_variable([24])

h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1, 2) + b_conv1)

#second convolutional layer
W_conv2 = weight_variable([5, 5, 24, 36])
b_conv2 = bias_variable([36])

h_conv2 = tf.nn.relu(conv2d(h_conv1, W_conv2, 2) + b_conv2)

#third convolutional layer
W_conv3 = weight_variable([5, 5, 36, 48])
b_conv3 = bias_variable([48])

h_conv3 = tf.nn.relu(conv2d(h_conv2, W_conv3, 2) + b_conv3)

#fourth convolutional layer
W_conv4 = weight_variable([3, 3, 48, 64])
b_conv4 = bias_variable([64])

h_conv4 = tf.nn.relu(conv2d(h_conv3, W_conv4, 1) + b_conv4)

#fifth convolutional layer
W_conv5 = weight_variable([3, 3, 64, 64])
b_conv5 = bias_variable([64])

h_conv5 = tf.nn.relu(conv2d(h_conv4, W_conv5, 1) + b_conv5)

#FCL 1
W_fc1 = weight_variable([1152, 1164])
b_fc1 = bias_variable([1164])

h_conv5_flat = tf.reshape(h_conv5, [-1, 1152])
h_fc1 = tf.nn.relu(tf.matmul(h_conv5_flat, W_fc1) + b_fc1)

keep_prob = tf.placeholder(tf.float32)
```

```
keep_prob = tf.placeholder(tf.float32)
h_fc1_drop = tf.nn.dropout(h_fc1, keep_prob)

#FCL 2
W_fc2 = weight_variable([1164, 100])
b_fc2 = bias_variable([100])

h_fc2 = tf.nn.relu(tf.matmul(h_fc1_drop, W_fc2) + b_fc2)

h_fc2_drop = tf.nn.dropout(h_fc2, keep_prob)

#FCL 3
W_fc3 = weight_variable([100, 50])
b_fc3 = bias_variable([50])

h_fc3 = tf.nn.relu(tf.matmul(h_fc2_drop, W_fc3) + b_fc3)

h_fc3_drop = tf.nn.dropout(h_fc3, keep_prob)

#FCL 3
W_fc4 = weight_variable([50, 10])
b_fc4 = bias_variable([10])

h_fc4 = tf.nn.relu(tf.matmul(h_fc3_drop, W_fc4) + b_fc4)

h_fc4_drop = tf.nn.dropout(h_fc4, keep_prob)

#Output
W_fc5 = weight_variable([10, 1])
b_fc5 = bias_variable([1])

y = tf.matmul(h_fc4_drop, W_fc5) + b_fc5 #scale the atan output
```

```
WARNING:tensorflow:From C:\Users\bolua\AppData\Local\Continuum\anaconda3\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From <ipython-input-3-85648d2ba561>:58: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
```

In [4]:

```
# 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')


split =0.8
X = []
y = []
LIMIT = None
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)*0.7)

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

```
Completed processing data.txt
```

In [5]:

```
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\bolua\AppData\Local\Continuum\anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6571: UserW
arning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

```
<Figure size 640x480 with 1 Axes>
```

In [6]:

```
#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):0.241561
Test_MSE(ZERO):0.241107
```

In [1]:

```
import os
import tensorflow as tf
from tensorflow.core.protobuf import saver_pb2
import driving_data
import model

LOGDIR = './save'

sess = tf.InteractiveSession()

L2NormConst = 0.001

train_vars = tf.trainable_variables()

loss = tf.reduce_mean(tf.square(tf.subtract(model.y_, model.y))) + tf.add_n([tf.nn.l2_loss(v) for v in
train_vars]) * L2NormConst
train_step = tf.train.AdamOptimizer(0.0001).minimize(loss)
sess.run(tf.initialize_all_variables())

# create a summary to monitor cost tensor
tf.summary.scalar("loss", loss)
# merge all summaries into a single op
merged_summary_op =  tf.summary.merge_all()

saver = tf.train.Saver(write_version = saver_pb2.SaverDef.V2)

# op to write logs to Tensorboard
logs_path = './logs'
```

```
logs_path = './logs'
summary_writer = tf.summary.FileWriter(logs_path, graph=tf.get_default_graph())

epochs = 30
batch_size = 1000

# train over the dataset about 30 times
for epoch in range(epochs):
  for i in range(int(driving_data.num_images/batch_size)):
    xs, ys = driving_data.LoadTrainBatch(batch_size)
    train_step.run(feed_dict={model.x: xs, model.y_: ys, model.keep_prob: 0.5})
    if i % 10 == 0:
      xs, ys = driving_data.LoadValBatch(batch_size)
      loss_value = loss.eval(feed_dict={model.x:xs, model.y_: ys, model.keep_prob: 0.5})
      print("Epoch: %d, Step: %d, Loss: %g" % (epoch, epoch * batch_size + i, loss_value))

    # write logs at every iteration
    summary = merged_summary_op.eval(feed_dict={model.x:xs, model.y_: ys, model.keep_prob: 0.5})
    summary_writer.add_summary(summary, epoch * driving_data.num_images/batch_size + i)

    if i % batch_size == 0:
      if not os.path.exists(LOGDIR):
        os.makedirs(LOGDIR)
      checkpoint_path = os.path.join(LOGDIR, "model.ckpt")
      filename = saver.save(sess, checkpoint_path)
  print("Model saved in file: %s" % filename)
```

```
WARNING:tensorflow:From C:\Users\bolua\AppData\Local\Continuum\anaconda3\lib\site-packages\tensorflow\p
ython\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is deprecat
ed and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\bolua\ML\26 Selfdriving car\Autopilot-TensorFlow-master\model.py:58: c
alling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in
a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
WARNING:tensorflow:From C:\Users\bolua\AppData\Local\Continuum\anaconda3\lib\site-packages\tensorflow\p
ython\util\tf_should_use.py:193: initialize_all_variables (from tensorflow.python.ops.variables) is dep
recated and will be removed after 2017-03-02.
Instructions for updating:
Use `tf.global_variables_initializer` instead.
Epoch: 0, Step: 0, Loss: 7.34966
Epoch: 0, Step: 10, Loss: 6.28368
Epoch: 0, Step: 20, Loss: 6.69974
Epoch: 0, Step: 30, Loss: 6.00156
Epoch: 0, Step: 40, Loss: 5.93357
Model saved in file: ./save\model.ckpt
Epoch: 1, Step: 1000, Loss: 6.49015
Epoch: 1, Step: 1010, Loss: 5.86369
Epoch: 1, Step: 1020, Loss: 5.80238
Epoch: 1, Step: 1030, Loss: 5.69918
Epoch: 1, Step: 1040, Loss: 6.00694
Model saved in file: ./save\model.ckpt
Epoch: 2, Step: 2000, Loss: 5.63508
Epoch: 2, Step: 2010, Loss: 6.15848
Epoch: 2, Step: 2020, Loss: 5.46125
Epoch: 2, Step: 2030, Loss: 5.39945
Epoch: 2, Step: 2040, Loss: 5.50207
Model saved in file: ./save\model.ckpt
Epoch: 3, Step: 3000, Loss: 5.74006
Epoch: 3, Step: 3010, Loss: 5.28472
Epoch: 3, Step: 3020, Loss: 5.17879
Epoch: 3, Step: 3030, Loss: 5.39833
Epoch: 3, Step: 3040, Loss: 5.19149
Model saved in file: ./save\model.ckpt
Epoch: 4, Step: 4000, Loss: 5.65396
Epoch: 4, Step: 4010, Loss: 4.96888
Epoch: 4, Step: 4020, Loss: 4.90935
Epoch: 4, Step: 4030, Loss: 4.8551
Epoch: 4, Step: 4040, Loss: 5.38571
Model saved in file: ./save\model.ckpt
Epoch: 5, Step: 5000, Loss: 4.80764
Epoch: 5, Step: 5010, Loss: 4.70511
Epoch: 5, Step: 5020, Loss: 4.81079
Epoch: 5, Step: 5030, Loss: 4.84453
Epoch: 5, Step: 5040, Loss: 4.76088
```

```
Epoch: 5, Step: 5040, Loss: 4.76088
Model saved in file: ./save\model.ckpt
Epoch: 6, Step: 6000, Loss: 4.9173
Epoch: 6, Step: 6010, Loss: 4.45698
Epoch: 6, Step: 6020, Loss: 4.40394
Epoch: 6, Step: 6030, Loss: 4.95235
Epoch: 6, Step: 6040, Loss: 4.34234
Model saved in file: ./save\model.ckpt
Epoch: 7, Step: 7000, Loss: 4.27353
Epoch: 7, Step: 7010, Loss: 4.35137
Epoch: 7, Step: 7020, Loss: 4.45729
Epoch: 7, Step: 7030, Loss: 4.18379
Epoch: 7, Step: 7040, Loss: 4.63299
Model saved in file: ./save\model.ckpt
Epoch: 8, Step: 8000, Loss: 4.04582
Epoch: 8, Step: 8010, Loss: 3.99293
Epoch: 8, Step: 8020, Loss: 4.54105
Epoch: 8, Step: 8030, Loss: 3.94988
Epoch: 8, Step: 8040, Loss: 3.85601
Model saved in file: ./save\model.ckpt
Epoch: 9, Step: 9000, Loss: 3.94881
Epoch: 9, Step: 9010, Loss: 4.07057
Epoch: 9, Step: 9020, Loss: 3.80924
Epoch: 9, Step: 9030, Loss: 4.25113
Epoch: 9, Step: 9040, Loss: 3.64898
Model saved in file: ./save\model.ckpt
Epoch: 10, Step: 10000, Loss: 3.62266
Epoch: 10, Step: 10010, Loss: 4.16344
Epoch: 10, Step: 10020, Loss: 3.59173
Epoch: 10, Step: 10030, Loss: 3.50212
Epoch: 10, Step: 10040, Loss: 3.57334
Model saved in file: ./save\model.ckpt
Epoch: 11, Step: 11000, Loss: 3.70725
Epoch: 11, Step: 11010, Loss: 3.46774
Epoch: 11, Step: 11020, Loss: 3.91141
Epoch: 11, Step: 11030, Loss: 3.3105
Epoch: 11, Step: 11040, Loss: 3.26607
Model saved in file: ./save\model.ckpt
Epoch: 12, Step: 12000, Loss: 3.83126
Epoch: 12, Step: 12010, Loss: 3.25271
Epoch: 12, Step: 12020, Loss: 3.18903
Epoch: 12, Step: 12030, Loss: 3.25146
Epoch: 12, Step: 12040, Loss: 3.35348
Model saved in file: ./save\model.ckpt
Epoch: 13, Step: 13000, Loss: 3.16053
Epoch: 13, Step: 13010, Loss: 3.60229
Epoch: 13, Step: 13020, Loss: 3.00658
Epoch: 13, Step: 13030, Loss: 2.96534
Epoch: 13, Step: 13040, Loss: 3.51952
Model saved in file: ./save\model.ckpt
Epoch: 14, Step: 14000, Loss: 2.9527
Epoch: 14, Step: 14010, Loss: 2.9036
Epoch: 14, Step: 14020, Loss: 2.96212
Epoch: 14, Step: 14030, Loss: 3.05615
Epoch: 14, Step: 14040, Loss: 2.86088
Model saved in file: ./save\model.ckpt
Epoch: 15, Step: 15000, Loss: 3.32235
Epoch: 15, Step: 15010, Loss: 2.73061
Epoch: 15, Step: 15020, Loss: 2.69538
Epoch: 15, Step: 15030, Loss: 3.28219
Epoch: 15, Step: 15040, Loss: 2.66603
Model saved in file: ./save\model.ckpt
Epoch: 16, Step: 16000, Loss: 2.6402
Epoch: 16, Step: 16010, Loss: 2.68621
Epoch: 16, Step: 16020, Loss: 2.79336
Epoch: 16, Step: 16030, Loss: 2.6065
Epoch: 16, Step: 16040, Loss: 3.04542
Model saved in file: ./save\model.ckpt
Epoch: 17, Step: 17000, Loss: 2.4875
Epoch: 17, Step: 17010, Loss: 2.45869
Epoch: 17, Step: 17020, Loss: 3.01831
Epoch: 17, Step: 17030, Loss: 2.42708
Epoch: 17, Step: 17040, Loss: 2.39714
Model saved in file: ./save\model.ckpt
Epoch: 18, Step: 18000, Loss: 2.42567
Epoch: 18, Step: 18010, Loss: 2.58772
Epoch: 18, Step: 18020, Loss: 2.38387
Epoch: 18, Step: 18030, Loss: 2.84162
```

```
Epoch: 18, Step: 18030, Loss: 2.84162
Epoch: 18, Step: 18040, Loss: 2.26204
Model saved in file: ./save\model.ckpt
Epoch: 19, Step: 19000, Loss: 2.23275
Epoch: 19, Step: 19010, Loss: 2.79713
Epoch: 19, Step: 19020, Loss: 2.22152
Epoch: 19, Step: 19030, Loss: 2.18529
Epoch: 19, Step: 19040, Loss: 2.16178
Model saved in file: ./save\model.ckpt
Epoch: 20, Step: 20000, Loss: 2.40858
Epoch: 20, Step: 20010, Loss: 2.17216
Epoch: 20, Step: 20020, Loss: 2.67614
Epoch: 20, Step: 20030, Loss: 2.05627
Epoch: 20, Step: 20040, Loss: 2.02667
Model saved in file: ./save\model.ckpt
Epoch: 21, Step: 21000, Loss: 2.63607
Epoch: 21, Step: 21010, Loss: 2.03193
Epoch: 21, Step: 21020, Loss: 1.99634
Epoch: 21, Step: 21030, Loss: 1.95087
Epoch: 21, Step: 21040, Loss: 2.20487
Model saved in file: ./save\model.ckpt
Epoch: 22, Step: 22000, Loss: 1.98206
Epoch: 22, Step: 22010, Loss: 2.47041
Epoch: 22, Step: 22020, Loss: 1.87867
Epoch: 22, Step: 22030, Loss: 1.8693
Epoch: 22, Step: 22040, Loss: 2.49177
Model saved in file: ./save\model.ckpt
Epoch: 23, Step: 23000, Loss: 1.84405
Epoch: 23, Step: 23010, Loss: 1.82787
Epoch: 23, Step: 23020, Loss: 1.78735
Epoch: 23, Step: 23030, Loss: 2.02904
Epoch: 23, Step: 23040, Loss: 1.76283
Model saved in file: ./save\model.ckpt
Epoch: 24, Step: 24000, Loss: 2.33225
Epoch: 24, Step: 24010, Loss: 1.73553
Epoch: 24, Step: 24020, Loss: 1.73052
Epoch: 24, Step: 24030, Loss: 2.01364
Epoch: 24, Step: 24040, Loss: 1.94331
Model saved in file: ./save\model.ckpt
Epoch: 25, Step: 25000, Loss: 1.69571
Epoch: 25, Step: 25010, Loss: 1.66287
Epoch: 25, Step: 25020, Loss: 1.84235
Epoch: 25, Step: 25030, Loss: 1.64742
Epoch: 25, Step: 25040, Loss: 2.21502
Model saved in file: ./save\model.ckpt
Epoch: 26, Step: 26000, Loss: 1.63816
Epoch: 26, Step: 26010, Loss: 1.56753
Epoch: 26, Step: 26020, Loss: 1.54655
Epoch: 26, Step: 26030, Loss: 2.19994
Epoch: 26, Step: 26040, Loss: 1.54405
Model saved in file: ./save\model.ckpt
Epoch: 27, Step: 27000, Loss: 1.50049
Epoch: 27, Step: 27010, Loss: 1.62668
Epoch: 27, Step: 27020, Loss: 1.57588
Epoch: 27, Step: 27030, Loss: 2.00266
Epoch: 27, Step: 27040, Loss: 1.60818
Model saved in file: ./save\model.ckpt
Epoch: 28, Step: 28000, Loss: 1.42605
Epoch: 28, Step: 28010, Loss: 1.42289
Epoch: 28, Step: 28020, Loss: 2.1344
Epoch: 28, Step: 28030, Loss: 1.42582
Epoch: 28, Step: 28040, Loss: 1.36689
Model saved in file: ./save\model.ckpt
Epoch: 29, Step: 29000, Loss: 1.45798
Epoch: 29, Step: 29010, Loss: 1.50546
Epoch: 29, Step: 29020, Loss: 1.42198
Epoch: 29, Step: 29030, Loss: 1.89641
Epoch: 29, Step: 29040, Loss: 1.3587
Model saved in file: ./save\model.ckpt
```

In [ ]:

```
```