Self Driving Car Case Study

In [1]:
```
import numpy as np
import scipy.misc
import random
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

import scipy
import os
import numpy as np
import random

from scipy import pi
from itertools import islice
import scipy.misc
import cv2
from subprocess import call
```

```
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type,
(1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type,
(1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type,
(1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type,
(1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type,
(1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing
(type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type,
(1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

```
WARNING:tensorflow:From C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\compat\compat.py:175: di
sable_resource_variables (from tensorflow.python.ops.variable_scope) is deprecated and will be removed in a future vers
ion.
Instructions for updating:
non-resource variables are not supported in the long term
```

Reading images

In [2]:
```
# 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 =0.7
X = []
y = []
with open(TRAIN_FILE) as fp:
    for line in fp:
        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:]
```

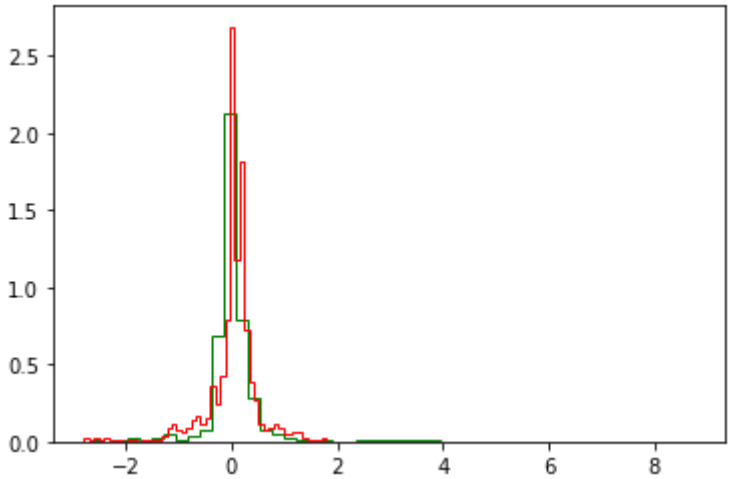
```
Completed processing data.txt
```

In [3]:
```
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_train_images])[-150:
        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_images])[-150:], [66
        y_out.append([val_ys[(val_batch_pointer + i) % num_val_images]])
    val_batch_pointer += batch_size
    return x_out, y_out
```

In [4]:
```
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\hims1\.conda\envs\gputest\lib\site-packages\ipykernel_launcher.py:5: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
  """
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\ipykernel_launcher.py:6: MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
```



In [5]:
```
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_in = tf.placeholder(tf.float32, shape=[None, 66, 200, 3])
y_tr = tf.placeholder(tf.float32, shape=[None, 1])

x_image = x_in

#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)
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_pred = tf.matmul(h_fc4_drop, W_fc5) + b_fc5
```

```
WARNING:tensorflow:From C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\framework\op_def_librar
y.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-5-160811d01183>:55: calling dropout (from tensorflow.python.ops.nn_ops) with kee
p_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`.
```

Training Model

In [6]:
```
!pip install -U scikit-image
```

```
Requirement already up-to-date: scikit-image in c:\users\hims1\.conda\envs\gputest\lib\site-packages (0.16.2)
Requirement already satisfied, skipping upgrade: networkx>=2.0 in c:\users\hims1\.conda\envs\gputest\lib\site-packages
(from scikit-image) (2.4)
Requirement already satisfied, skipping upgrade: PyWavelets>=0.4.0 in c:\users\hims1\.conda\envs\gputest\lib\site-packa
ges (from scikit-image) (1.1.1)
Requirement already satisfied, skipping upgrade: scipy>=0.19.0 in c:\users\hims1\.conda\envs\gputest\lib\site-packages
(from scikit-image) (1.1.0)
Requirement already satisfied, skipping upgrade: imageio>=2.3.0 in c:\users\hims1\.conda\envs\gputest\lib\site-packages
(from scikit-image) (2.6.1)
Requirement already satisfied, skipping upgrade: pillow>=4.3.0 in c:\users\hims1\.conda\envs\gputest\lib\site-packages
(from scikit-image) (5.2.0)
Requirement already satisfied, skipping upgrade: matplotlib!=3.0.0,>=2.0.0 in c:\users\hims1\.conda\envs\gputest\lib\si
te-packages (from scikit-image) (3.1.3)
Requirement already satisfied, skipping upgrade: decorator>=4.3.0 in c:\users\hims1\.conda\envs\gputest\lib\site-packag
es (from networkx>=2.0->scikit-image) (4.4.1)
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in c:\users\hims1\.conda\envs\gputest\lib\site-packages
(from PyWavelets>=0.4.0->scikit-image) (1.18.1)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in c:\users\hims1\.conda\envs\gputest\lib\site-pa
ckages (from matplotlib!=3.0.0,>=2.0.0->scikit-image) (2.8.1)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in c:\users\hims1\.conda\envs\gputest\lib\site-packa
ges (from matplotlib!=3.0.0,>=2.0.0->scikit-image) (1.1.0)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in c:\users\hims1\.conda\envs\gputest\lib\site-packages
(from matplotlib!=3.0.0,>=2.0.0->scikit-image) (0.10.0)
Requirement already satisfied, skipping upgrade: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\hims1\.conda\envs
\gputest\lib\site-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image) (2.4.6)
Requirement already satisfied, skipping upgrade: six>=1.5 in c:\users\hims1\.conda\envs\gputest\lib\site-packages (from
python-dateutil>=2.1->matplotlib!=3.0.0,>=2.0.0->scikit-image) (1.14.0)
Requirement already satisfied, skipping upgrade: setuptools in c:\users\hims1\.conda\envs\gputest\lib\site-packages (fr
om kiwisolver>=1.0.1->matplotlib!=3.0.0,>=2.0.0->scikit-image) (45.2.0.post20200210)
```

In [7]:
```
LOGDIR = './save'

sess = tf.InteractiveSession()

L2NormConst = 0.001

train_vars = tf.trainable_variables()

loss = tf.reduce_mean(tf.square(tf.subtract(y_tr, y_pred))) + tf.add_n([tf.nn.l2_loss(v) for v in train_vars]) * L2NormC
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()

# op to write logs to Tensorboard
logs_path = './logs'
summary_writer = tf.summary.FileWriter(logs_path, graph=tf.get_default_graph())

epochs = 30
batch_size = 100

# train over the dataset about 30 times
for epoch in range(epochs):
    for i in range(int(num_images/batch_size)):
        xs, ys = LoadTrainBatch(batch_size)
        train_step.run(feed_dict={x_in: xs, y_tr: ys, keep_prob: 0.5})
        if i % 10 == 0:
            xs, ys = LoadValBatch(batch_size)
            loss_value = loss.eval(feed_dict={x_in:xs, y_tr: ys, 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={x_in:xs, y_tr: ys, keep_prob: 0.5})
    summary_writer.add_summary(summary, epoch * 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)

    if (i+1) % batch_size == 0:
        if not os.path.exists(SAVEDIR):
            os.makedirs(SAVEDIR)
        save_path = os.path.join(SAVEDIR, "model.ckpt")
        saver.save(sess = sess, save_path = save_path)
        print("Model saved at location {} at epoch {}".format(save_path, epoch + 1))

print("Run the command line:\n" \
          "--> tensorboard --logdir=./logs " \
          "\nThen open http://0.0.0.0:6006/ into your web browser")
```

```
WARNING:tensorflow:From C:\Users\hims1\.conda\envs\gputest\lib\site-packages\tensorflow\python\util\tf_should_use.py:
193: initialize_all_variables (from tensorflow.python.ops.variables) is deprecated and will be removed after 2017-03-
02.
Instructions for updating:
Use `tf.global_variables_initializer` instead.

C:\Users\hims1\.conda\envs\gputest\lib\site-packages\ipykernel_launcher.py:38: DeprecationWarning: `imread` is deprec
ated!
`imread` is deprecated in SciPy 1.0.0, and will be removed in 1.2.0.
Use ``imageio.imread`` instead.
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\ipykernel_launcher.py:38: DeprecationWarning: `imresize` is depr
ecated!
`imresize` is deprecated in SciPy 1.0.0, and will be removed in 1.2.0.
Use ``skimage.transform.resize`` instead.
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\ipykernel_launcher.py:48: DeprecationWarning: `imread` is deprec
ated!
`imread` is deprecated in SciPy 1.0.0, and will be removed in 1.2.0.
Use ``imageio.imread`` instead.
C:\Users\hims1\.conda\envs\gputest\lib\site-packages\ipykernel_launcher.py:48: DeprecationWarning: `imresize` is depr
ecated!
```

In [9]:
```
import os
os.getcwd()
```

Out[9]:
```
'C:\\Users\\hims1\\Autopilot-TensorFlow-master\\Autopilot-TensorFlow-master'
```

In [ ]:
```
import tensorflow as tf
import scipy.misc
import model
import cv2
from subprocess import call

sess = tf.InteractiveSession()
saver = tf.train.Saver()
saver.restore(sess, "save/model.ckpt")

img = cv2.imread('steering_wheel_image.jpg',0)
rows,cols = img.shape

smoothed_angle = 0

cap = cv2.VideoCapture(0)
while(cv2.waitKey(10) != ord('q')):
    ret, frame = cap.read()
    image = scipy.misc.imresize(frame, [66, 200]) / 255.0
    degrees = model.y.eval(feed_dict={model.x: [image], model.keep_prob: 1.0})[0][0] * 180 / scipy.pi
    call("clear")
    print("Predicted steering angle: " + str(degrees) + " degrees")
    cv2.imshow('frame', frame)
    #make smooth angle transitions by turning the steering wheel based on the difference of the current angle
    #and the predicted angle
    smoothed_angle += 0.2 * pow(abs((degrees - smoothed_angle)), 2.0 / 3.0) * (degrees - smoothed_angle) / abs(degrees -
    M = cv2.getRotationMatrix2D((cols/2,rows/2),-smoothed_angle,1)
    dst = cv2.warpAffine(img,M,(cols,rows))
    cv2.imshow("steering wheel", dst)

cap.release()
cv2.destroyAllWindows()
```

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-KJCaaF7elleG1RbzVPZ
# 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
```