Behavioral Cloning

目录

Behavioral Cloning....1
Local environment setup.....2
Training in simulator.....2
Prepare datasets.....2
Construct model.....4
Model training and validation.....5
Save model.....5
Run in autonomous mode 6

Local environment setup.

Laptop PC with GTX1060.
Install Simulator.
Use anaconda navigator ,create term1-gpu environment.(tensorflow with gpu support)

Training in simulator.

First select “TRAINING MODE” menu, then enter into simulator’s training mode.
Press ‘R’ then select the folder where the images and csv file will be stored. In my computer ,I select “mine_data”. Press ‘R’ again, then press the direction keys in keyboard in order to let the car run in the center of the road. Collect 5 rounds clockwise and 5 rounds counter clockwise datas.

Extend training in turning .

Prepare datasets.

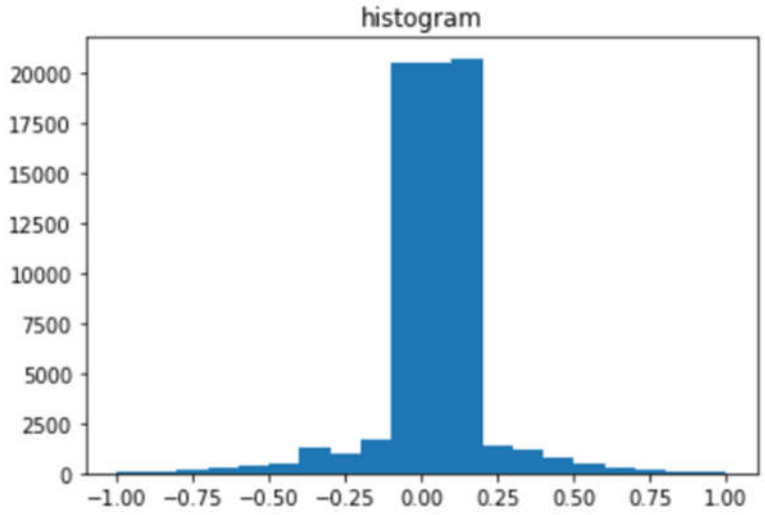
Get all images' data (center, left, right, angle) according to csv file. In order to argument data, append flip image data of all images. The cods clip :

```
lines = []
with open("./mine_data/driving_log.csv") as csv_file:
    reader = csv.reader(csv_file)
    for line in reader:
        lines.append(line)
```

```
images = []
measurements = []
for line in lines:
    correction = 0.1
    measurement = float(line[3])

    # center
    source_path = line[0]
    file_name = source_path.split('\\')[-1]
    current_path = './mine_data/IMG/' + file_name
    image_center = cv2.imread(current_path)
    flip_image_center = cv2.flip(image_center, 1)
    measurement_center = measurement
    flip_measurement_center = measurement_center*-1
```

```
    images.extend([image_center, flip_image_center, image_left, flip_image_left, image_right, flip_image_right])
    measurements.extend([measurement_center, flip_measurement_center, measurement_left, flip_measurement_left, measurement_right, flip_measur
```

```
X_train = np.array(images)
y_train = np.array(measurements)
```



Construct model.

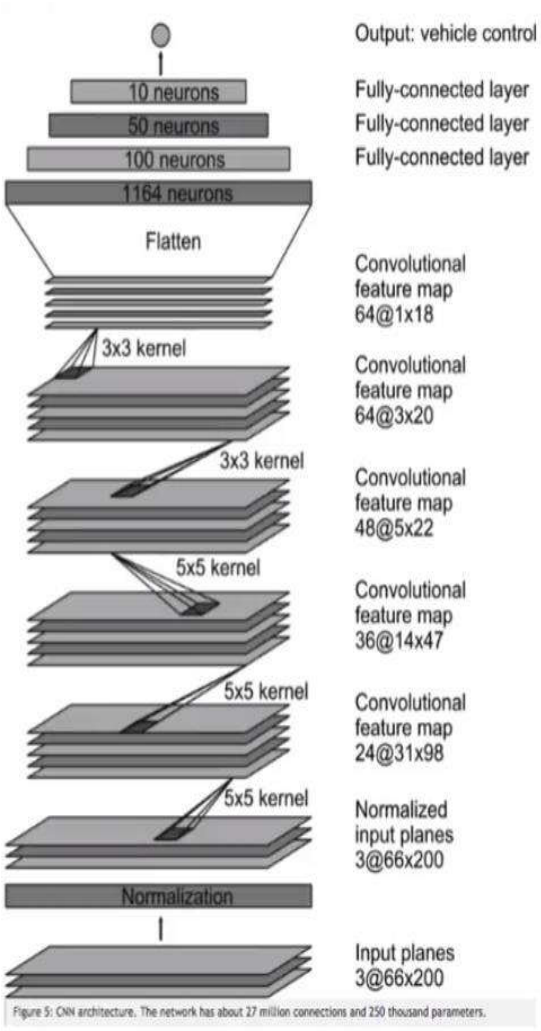


Figure 5: CNN architecture. The network has about 27 million connections and 250 thousand parameters.

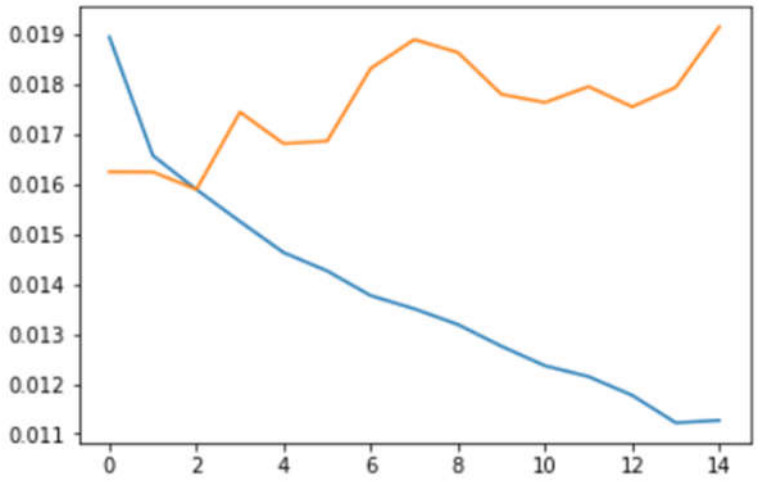
```
model = Sequential()
model.add(Lambda(lambda x: (x / 255.0) - 0.5, input_shape=(160,320,3)))
model.add(Cropping2D(cropping=((50,20), (0,0)),
input_shape=(160,320,3)))


model.add(Conv2D(24,(5,5),strides=(2,2),activation='relu'))
model.add(Conv2D(36,(5,5),strides=(2,2),activation='relu'))
model.add(Conv2D(48,(5,5),strides=(2,2),activation='relu'))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Conv2D(64,(3,3),activation='relu'))

model.add(Flatten())

model.add(Dense(100))
model.add(Dropout(.2))
model.add(Dense(50))
model.add(Dropout(.2))
model.add(Dense(20))
model.add(Dropout(.2))
model.add(Dense(10))
model.add(Dropout(.2))
model.add(Dense(1))


model.compile(loss='mse', optimizer='adam')
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lambda_1 (Lambda)            (None, 160, 320, 3)       0
_________________________________________________________________
cropping2d_1 (Cropping2D)    (None, 90, 320, 3)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 43, 158, 24)       1824
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 20, 77, 36)        21636
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 37, 48)         43248
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 6, 35, 64)         27712
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 4, 33, 64)         36928
_________________________________________________________________
flatten_1 (Flatten)          (None, 8448)              0
_________________________________________________________________
```

```
dense_1 (Dense)              (None, 100)               844900
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dropout_2 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_3 (Dense)              (None, 20)                1020
_________________________________________________________________
dropout_3 (Dropout)          (None, 20)                0
_________________________________________________________________
dense_4 (Dense)              (None, 10)                210
_________________________________________________________________
dropout_4 (Dropout)          (None, 10)                0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 11
=================================================================
Total params: 982,539
Trainable params: 982,539
Non-trainable params: 0
```

Model training and validation.

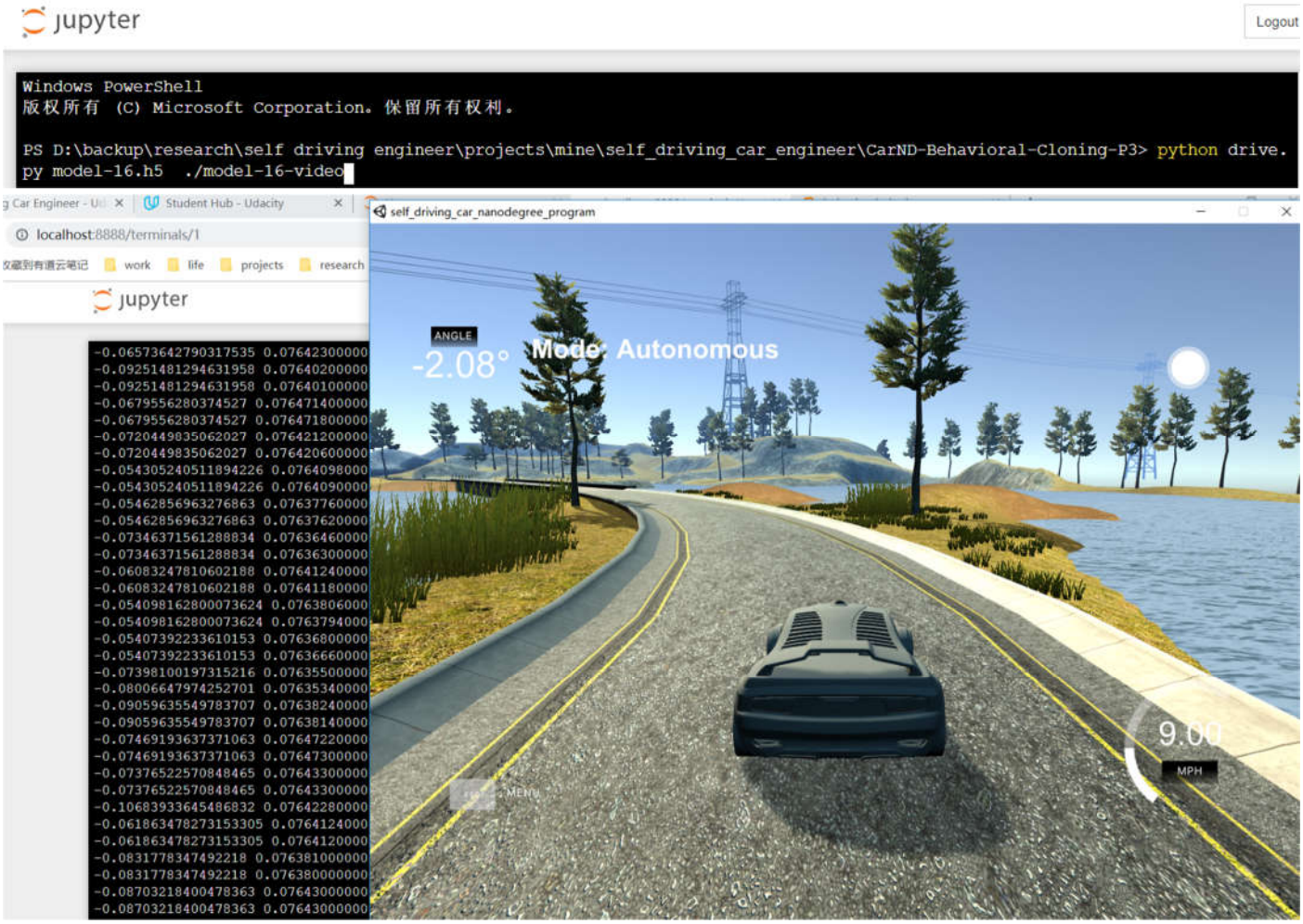
```
history_obj = model.fit(X_train, y_train, validation_split=0.2, shuffle=True, epochs=15)
```



Save model.

```
model.save("track1_model_16.h5")
```

Run in autonomous mode .



Generator video.

```
PS D:\backup\research\self driving engineer\projects\mine\self_driving_car_engineer\CarND-Behavioral-Cloning-P3> python video.
py model-16-video
Creating video model-16-video, FPS=60
[MoviePy] >>>> Building video model-16-video.mp4
[MoviePy] Writing video model-16-video.mp4
  1%|▌                                                                                | 40/6062 [00:00<00:15, 399.31it/s]
  1%|▊                                                                                | 77/6062 [00:00<00:15, 387.98it/s]
  2%|▉                                                                               | 113/6062 [00:00<00:15, 377.14it/s
  2%|▉                                                                               | 151/6062 [00:00<00:15, 377.22it/s
  3%|▉                                                                               | 188/6062 [00:00<00:15, 374.24it/
  4%|▉                                                                               | 225/6062 [00:00<00:15, 370.52it/
  4%|▉                                                                               | 261/6062 [00:00<00:15, 365.97it
  5%|▉                                                                               | 296/6062 [00:00<00:16, 360.26it
  5%|▉                                                                               | 333/6062 [00:00<00:15, 361.87i
  6%|▉                                                                               | 368/6062 [00:01<00:16, 355.85i
  7%|▉                                                                               | 403/6062 [00:01<00:16, 349.14
  7%|▉                                                                               | 438/6062 [00:01<00:17, 325.52
  8%|▉                                                                               | 471/6062 [00:01<00:17, 312.8
  8%|▉                                                                               | 503/6062 [00:01<00:17, 313.8
  9%|▉                                                                               | 537/6062 [00:01<00:17, 319.
  9%|▉                                                                               | 569/6062 [00:01<00:17, 307.
 10%|▉                                                                               | 604/6062 [00:01<00:17, 317
 10%|▉                                                                               | 636/6062 [00:01<00:17, 309
 11%|▉                                                                               | 673/6062 [00:01<00:16, 32
 12%|▉                                                                               | 706/6062 [00:02<00:16, 32
 12%|▉                                                                               | 739/6062 [00:02<00:16, 3
 13%|▉                                                                               | 774/6062 [00:02<00:16, 3
 13%|▉                                                                               | 809/6062 [00:02<00:16,
 14%|▉                                                                               | 845/6062 [00:02<00:15,
 15%|▉                                                                               | 881/6062 [00:02<00:15,
339.99it/s]
```