

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

```

```

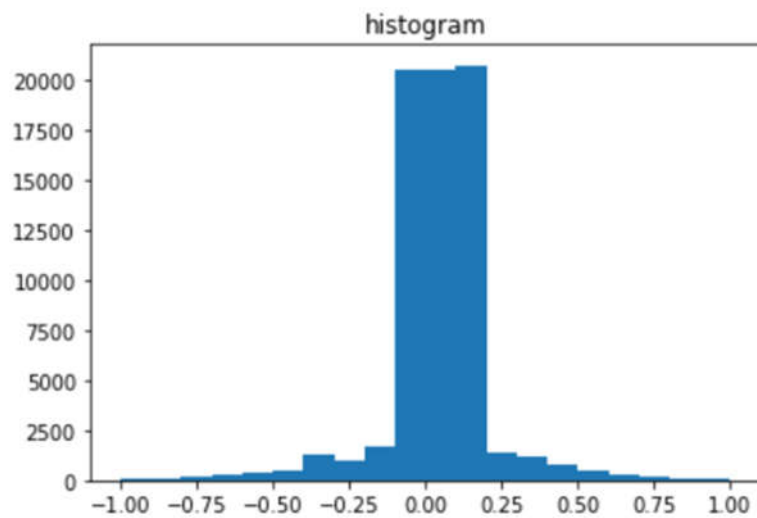
images.extend([image_center, flip_image_center, image_left, flip_image_left, image_right, flip_image_right])
measurements.extend([measurement_center, flip_measurement_center, measurement_left, flip_measurement_left, measurement_right, flip_measurement_right])

```

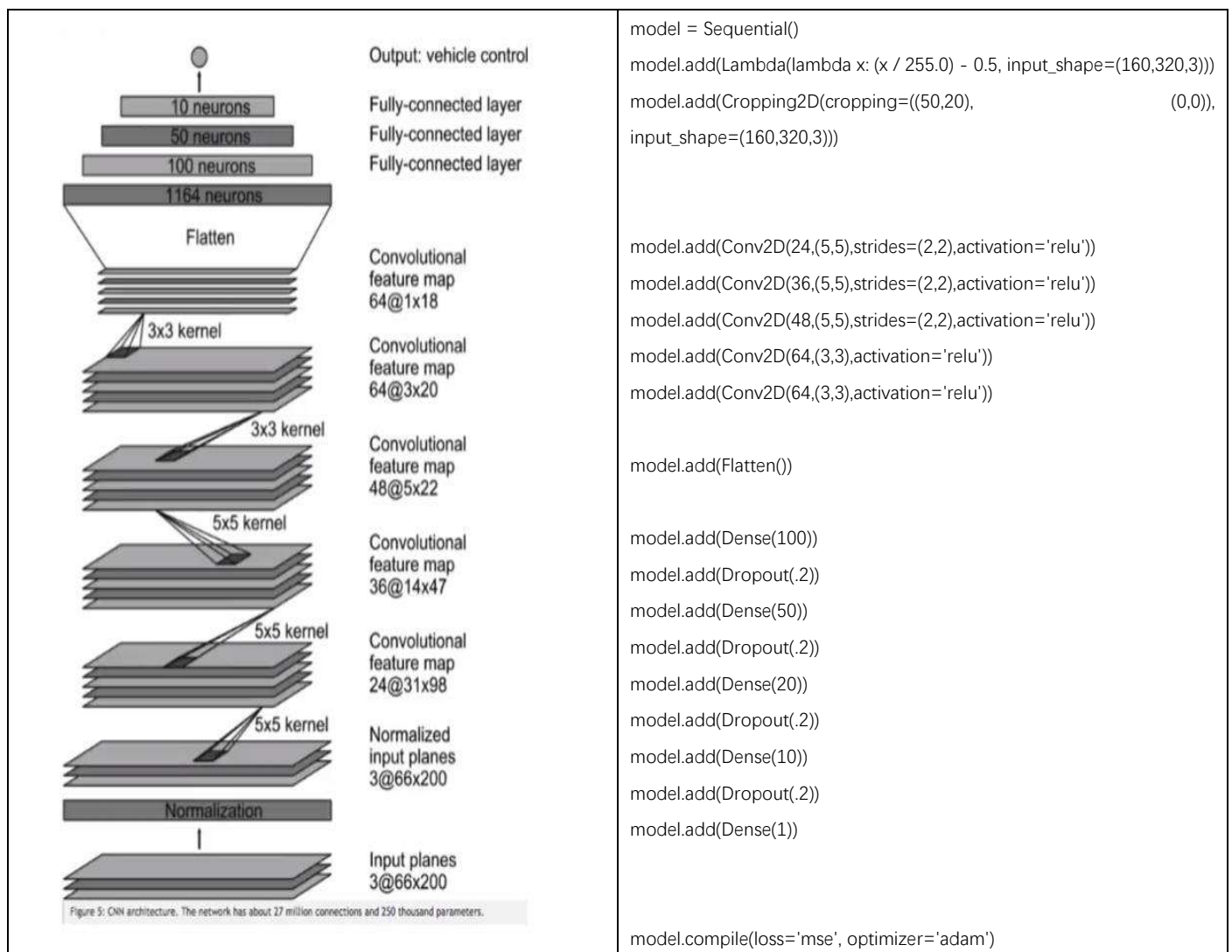
```

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

```



Construct model.



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

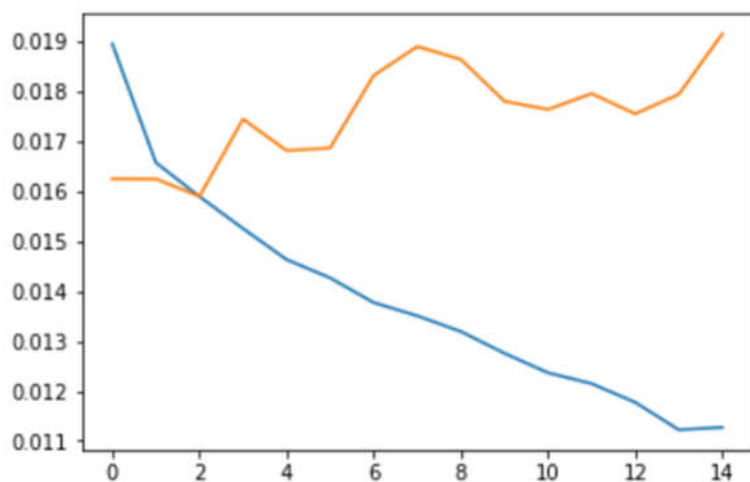
dense_1 (Dense)	(None, 100)	844900
dropout_1 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 50)	5050
dropout_2 (Dropout)	(None, 50)	0
dense_3 (Dense)	(None, 20)	1020
dropout_3 (Dropout)	(None, 20)	0
dense_4 (Dense)	(None, 10)	210
dropout_4 (Dropout)	(None, 10)	0
dense_5 (Dense)	(None, 1)	11

=====

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("model_16.h5")
```

Run in autonomous mode .

jupyter

Logout

```
Windows PowerShell
版权所有 (C) Microsoft Corporation. 保留所有权利。

PS D:\backup\research\self driving engineer\projects\mine\self_driving_car_engineer\CarND-Behavioral-Cloning-P3> python drive.py model-l6.h5 ./model-l6-video
```

localhost:8888/terminals/1

work life projects research

jupyter

self_driving_car_nanodegree_program

Mode: Autonomous

ANGLE -2.08°

9.00 MPH

0.06573642790317535 0.07642300000
-0.09251481294631958 0.07640200000
-0.09251481294631958 0.07640100000
-0.0679556280374527 0.07647140000
-0.0679556280374527 0.07647180000
-0.0720449835062027 0.07642120000
-0.0720449835062027 0.07642060000
-0.054305240511894226 0.07640980000
-0.054305240511894226 0.07640900000
-0.05462856963276863 0.07637760000
-0.05462856963276863 0.07637620000
-0.07346371561288834 0.07636460000
-0.07346371561288834 0.07636300000
-0.06083247810602188 0.07641240000
-0.06083247810602188 0.07641180000
-0.054098162800073624 0.07638060000
-0.054098162800073624 0.07637940000
-0.05407392233610153 0.07636800000
-0.05407392233610153 0.07636660000
-0.07398100197315216 0.07635500000
-0.08006647974252701 0.07635340000
-0.09059635549783707 0.07638240000
-0.09059635549783707 0.07638140000
-0.07469193637371063 0.07647220000
-0.07469193637371063 0.07647300000
-0.07376522570848465 0.07643300000
-0.07376522570848465 0.07643300000
-0.10683933645486832 0.07642280000
-0.061863478273153305 0.07641240000
-0.061863478273153305 0.07641200000
-0.0831778347492218 0.07638100000
-0.0831778347492218 0.07638000000
-0.08703218400478363 0.07643000000
-0.08703218400478363 0.07643000000

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/s]
4% | 225/6062 [00:00<00:15, 370.52it/s]
4% | 261/6062 [00:00<00:15, 365.97it/s]
5% | 296/6062 [00:00<00:16, 360.26it/s]
5% | 333/6062 [00:00<00:15, 361.87it/s]
6% | 368/6062 [00:01<00:16, 355.85it/s]
7% | 403/6062 [00:01<00:16, 349.14it/s]
7% | 438/6062 [00:01<00:17, 325.52it/s]
8% | 471/6062 [00:01<00:17, 312.8it/s]
8% | 503/6062 [00:01<00:17, 313.8it/s]
9% | 537/6062 [00:01<00:17, 319it/s]
9% | 569/6062 [00:01<00:17, 307it/s]
10% | 604/6062 [00:01<00:17, 317it/s]
10% | 636/6062 [00:01<00:17, 309it/s]
11% | 673/6062 [00:01<00:16, 32it/s]
12% | 706/6062 [00:02<00:16, 32it/s]
12% | 739/6062 [00:02<00:16, 3it/s]
13% | 774/6062 [00:02<00:16, 3it/s]
13% | 809/6062 [00:02<00:16, 3it/s]
14% | 845/6062 [00:02<00:15, 3it/s]
15% | 881/6062 [00:02<00:15, 3it/s]
```

