Behavioral Cloning

Said Zahrai

August 4, 2019

1 Introduction

The goal of this project is to clone the behavior of a driver driving a car in a Unity game. A player needs to play first and record data. While the player is playing, the environment is observed with three cameras mounted in the middle of the car, and in the two left and right corners.

2 Data

2.1 Camera recorded images

2.2 Data augmentation

```
img = plt.imread(image_file_address)
if (abs(batch_sample[1])<0.01):
   r = random.randint(0,400)
   if (r<20):
      images.append(img)
      angles.append(steering_value)
      images.append(np.fliplr(img))
      angles.append(-steering_value)
   elif (data_augmentation):
      r =random.randint(-40,40)
            if (abs(r)<21):
        M = np.float32([[1,0, r],[0,1,0]])
        images.append(cv2.warpAffine(img,M,(cols,rows)))
        angles.append(steering_value+0.01*r)
```

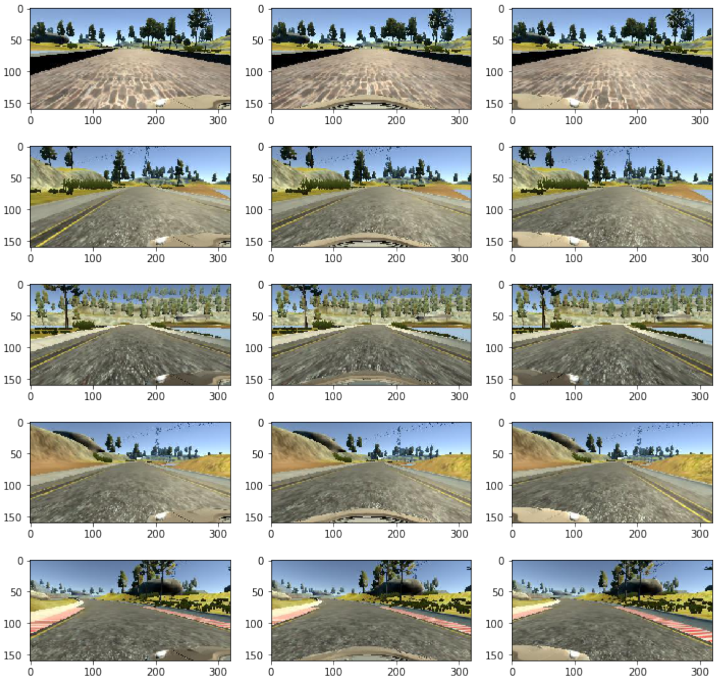

Figure 1: Camera recorded images. Left, seen from left camera, center from center camera and right from right camera.

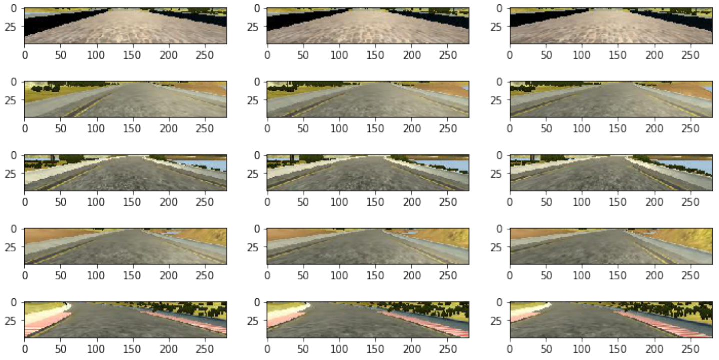

Figure 2: Corped images with focus on interesting region. Left, seen from left camera, center from center camera and right from right camera.

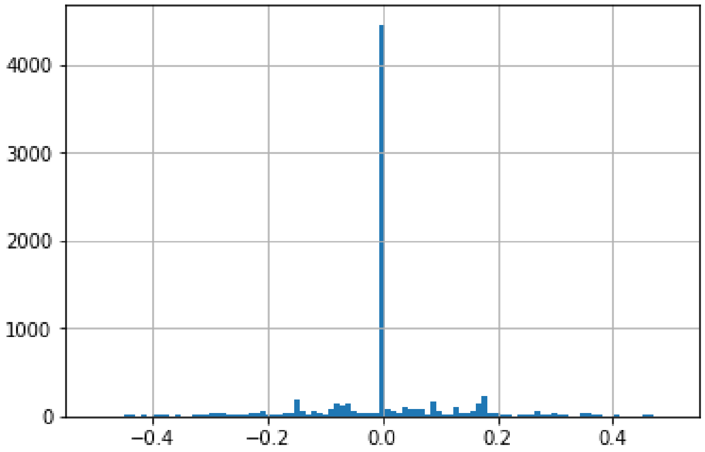

Figure 3: Distribution of steering values.

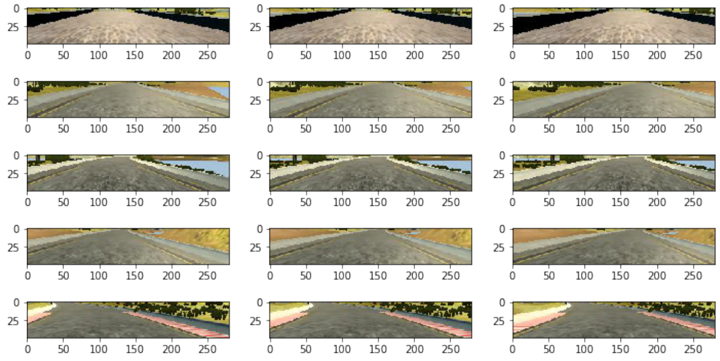

Figure 4: Shifted images: Left, shifted 20 pixels to left, center, original image and right, shifted 20 pixels to right.

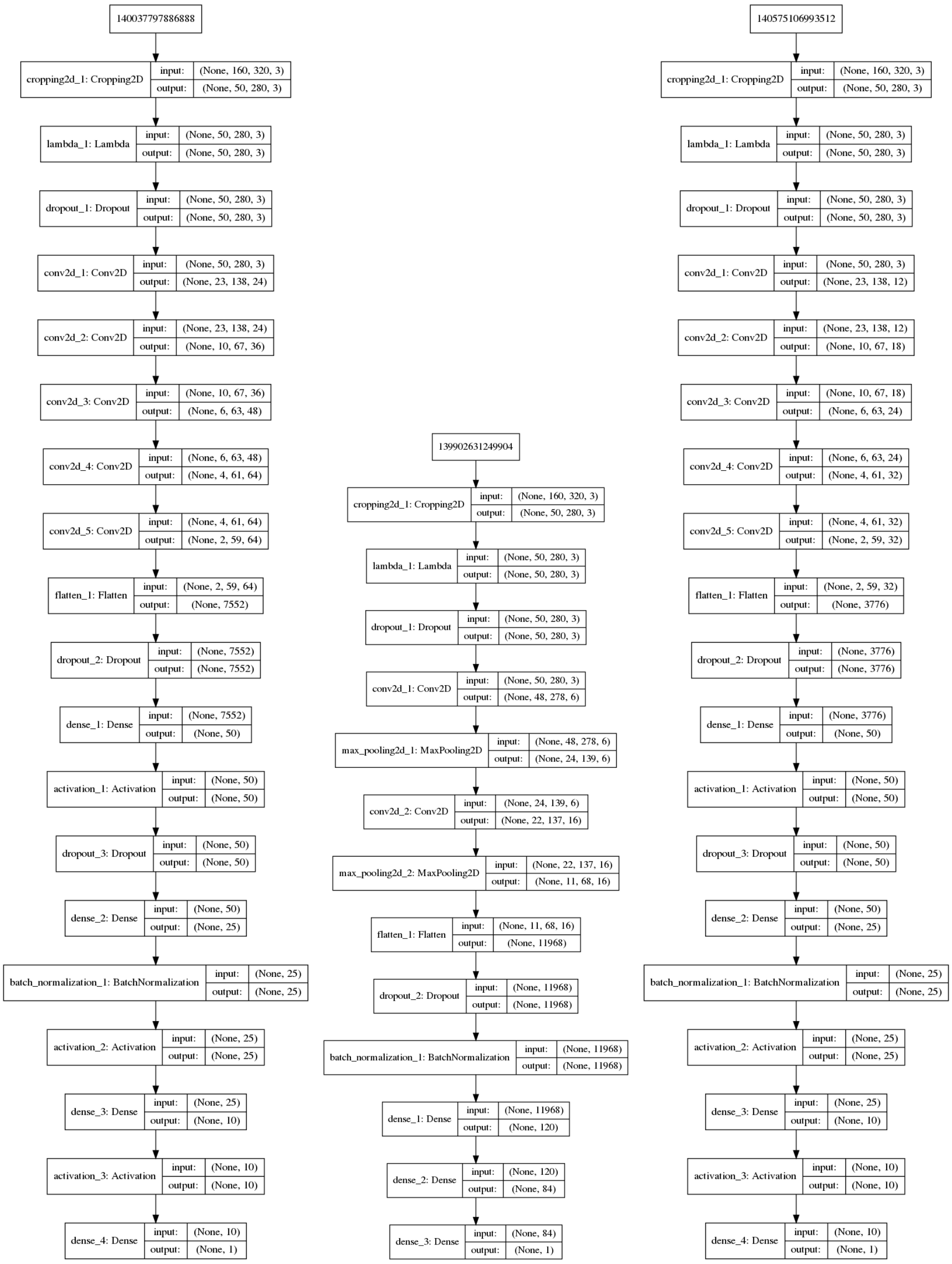
140037797886888
cropping2d_1: Cropping2D input: (None, 160, 320, 3) output: (None, 50, 280, 3)
lambda_1: Lambda input: (None, 50, 280, 3) output: (None, 50, 280, 3)
dropout_1: Dropout input: (None, 50, 280, 3) output: (None, 50, 280, 3)
conv2d_1: Conv2D input: (None, 50, 280, 3) output: (None, 23, 138, 24)
conv2d_2: Conv2D input: (None, 23, 138, 24) output: (None, 10, 67, 36)
conv2d_3: Conv2D input: (None, 10, 67, 36) output: (None, 6, 63, 48)
conv2d_4: Conv2D input: (None, 6, 63, 48) output: (None, 4, 61, 64)
conv2d_5: Conv2D input: (None, 4, 61, 64) output: (None, 2, 59, 64)
flatten_1: Flatten input: (None, 2, 59, 64) output: (None, 7552)
dropout_2: Dropout input: (None, 7552) output: (None, 7552)
dense_1: Dense input: (None, 7552) output: (None, 50)
activation_1: Activation input: (None, 50) output: (None, 50)
dropout_3: Dropout input: (None, 50) output: (None, 50)
dense_2: Dense input: (None, 50) output: (None, 25)
batch_normalization_1: BatchNormalization input: (None, 25) output: (None, 25)
activation_2: Activation input: (None, 25) output: (None, 25)
dense_3: Dense input: (None, 25) output: (None, 10)
activation_3: Activation input: (None, 10) output: (None, 10)
dense_4: Dense input: (None, 10) output: (None, 1)
139902631249904
cropping2d_1: Cropping2D input: (None, 160, 320, 3) output: (None, 50, 280, 3)
lambda_1: Lambda input: (None, 50, 280, 3) output: (None, 50, 280, 3)
dropout_1: Dropout input: (None, 50, 280, 3) output: (None, 50, 280, 3)
conv2d_1: Conv2D input: (None, 50, 280, 3) output: (None, 48, 278, 6)
max_pooling2d_1: MaxPooling2D input: (None, 48, 278, 6) output: (None, 24, 139, 6)
conv2d_2: Conv2D input: (None, 24, 139, 6) output: (None, 22, 137, 16)
max_pooling2d_2: MaxPooling2D input: (None, 22, 137, 16) output: (None, 11, 68, 16)
flatten_1: Flatten input: (None, 11, 68, 16) output: (None, 11968)
dropout_2: Dropout input: (None, 11968) output: (None, 11968)
batch_normalization_1: BatchNormalization input: (None, 11968) output: (None, 11968)
dense_1: Dense input: (None, 11968) output: (None, 120)
dense_2: Dense input: (None, 120) output: (None, 84)
dense_3: Dense input: (None, 84) output: (None, 1)
140575106993512
cropping2d_1: Cropping2D input: (None, 160, 320, 3) output: (None, 50, 280, 3)
lambda_1: Lambda input: (None, 50, 280, 3) output: (None, 50, 280, 3)
dropout_1: Dropout input: (None, 50, 280, 3) output: (None, 50, 280, 3)
conv2d_1: Conv2D input: (None, 50, 280, 3) output: (None, 23, 138, 12)
conv2d_2: Conv2D input: (None, 23, 138, 12) output: (None, 10, 67, 18)
conv2d_3: Conv2D input: (None, 10, 67, 18) output: (None, 6, 63, 24)
conv2d_4: Conv2D input: (None, 6, 63, 24) output: (None, 4, 61, 32)
conv2d_5: Conv2D input: (None, 4, 61, 32) output: (None, 2, 59, 32)
flatten_1: Flatten input: (None, 2, 59, 32) output: (None, 3776)
dropout_2: Dropout input: (None, 3776) output: (None, 3776)
dense_1: Dense input: (None, 3776) output: (None, 50)
activation_1: Activation input: (None, 50) output: (None, 50)
dropout_3: Dropout input: (None, 50) output: (None, 50)
dense_2: Dense input: (None, 50) output: (None, 25)
batch_normalization_1: BatchNormalization input: (None, 25) output: (None, 25)
activation_2: Activation input: (None, 25) output: (None, 25)
dense_3: Dense input: (None, 25) output: (None, 10)
activation_3: Activation input: (None, 10) output: (None, 10)
dense_4: Dense input: (None, 10) output: (None, 1)

Figure 5: Model structure: Left, Nvidia's network, center, Lenet's network and right, reduced version of Nvidia's network.

Figure 6: Convergence history: Left, Nvidia's network, center, Lenet's network and right, reduced version of Nvidia's network.