Behavioral Cloning

Udacity Self Driving Engineer –Term 1 Project 3

Overview

This project involves building a behavioral cloning model based on the Udacity car driving simulator.

Project files

- model.py containing the script to create and train the model using Udacity provided data.
- successive_training.py contains the script for successive training of the model based on data that I generated.
- drive.py for driving the car in autonomous mode.
- model.json model description in JSON format.
- model.h5 containing a trained convolution neural network.
- README.pdf – project write-up (this file)

Running the model

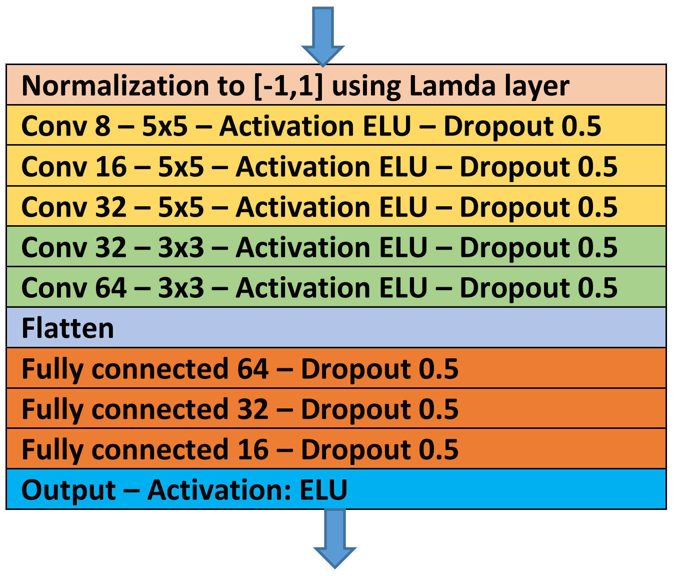
Model can be run to drive the car in autonomous mode on the simulator by the below command:

```
python drive.py model.json
```

Model Architecture and Training Strategy

Initially I conceptualized two different models to experiment, train and test: designed based-on [1] and model inspired by [2]. I found the [2] to perform better upon initial testing and continued to fine tune the model further.

Model architecture visualization is shown below:


Normalization to [-1,1] using Lamda layer
Conv 8 – 5x5 – Activation ELU – Dropout 0.5
Conv 16 – 5x5 – Activation ELU – Dropout 0.5
Conv 32 – 5x5 – Activation ELU – Dropout 0.5
Conv 32 – 3x3 – Activation ELU – Dropout 0.5
Conv 64 – 3x3 – Activation ELU – Dropout 0.5
Flatten
Fully connected 64 – Dropout 0.5
Fully connected 32 – Dropout 0.5
Fully connected 16 – Dropout 0.5
Output – Activation: ELU

The model consists of five convolutional layers – three with filter size of 5x5 with depths 8, 16 and 32 and two with filter size of 3x3 with depths of 32 and 64. Each of the convolutional layer uses ELU as activation layer due its better performance [3]. To reduce over-fitting of the model, dropout layer is following each of the convolutional layers. A high dropout value of 0.5 is used to provide maximum regularization [4].

The convolutional layers are followed by a flatten layer and three fully connected layers of 64, 32 and 16 with dropout of 0.5. Final output layer is followed by ELU activation function.

I used data augmentation extensively using the details as described by Vivek [5]. Random shadow, flipping of the image, horizontal and vertical shifting and brightness changes are used. The illustration is shown in the figure below. The image is pre-processed from original 160x320x3 to 20x80x3 in two steps-

(1) Top 60 pixels and bottom 40 pixels were removed. Top portions of the images contain topographical information not very useful for the car to drive and in fact impairs with training the model.
(2) The cut image from above step is resized to 20x80x3. This reduces the training time for the model. The image resize did not seem to have any noticeable effect on the training results.

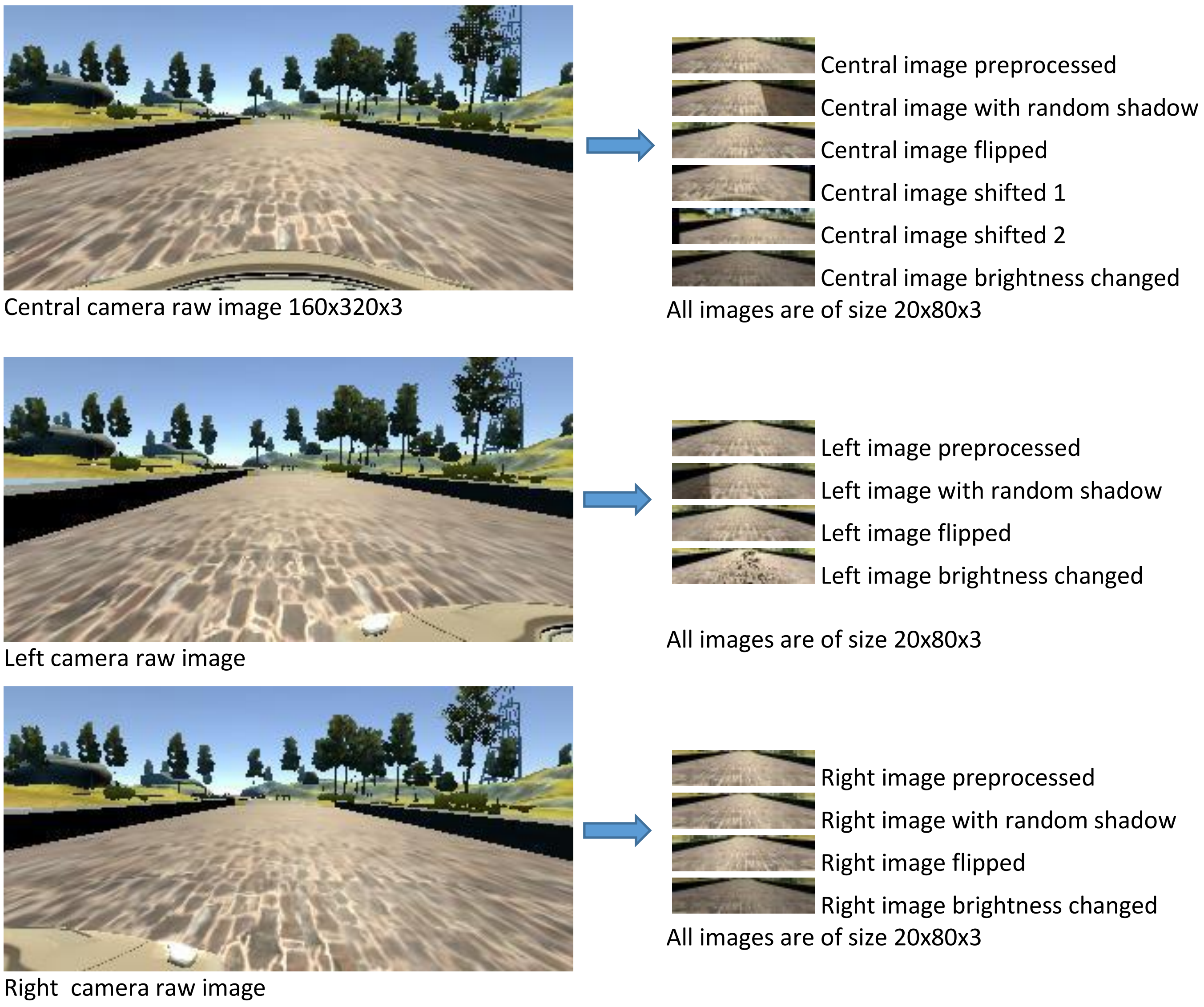

Central image preprocessed
Central image with random shadow
Central image flipped
Central image shifted 1
Central image shifted 2
Central image brightness changed
Central camera raw image 160x320x3
All images are of size 20x80x3
Left image preprocessed
Left image with random shadow
Left image flipped
Left image brightness changed
Left camera raw image
All images are of size 20x80x3
Right image preprocessed
Right image with random shadow
Right image flipped
Right image brightness changed
All images are of size 20x80x3
Right camera raw image

Figure 1 Image preprocessing and data augmentation

For left camera images: The steering angle is taken (steering angle + 0.25)

For right camera images: The steering angle is taken (steering angle - 0.25)

For shifted images: 0.004 steering angle units per pixel shift to the right, and subtracted 0.004 steering angle units per pixel shift to the left. The images are also shifter vertically randomly to simulate up and down the slopes.

For flipped images: the steering angle is taken as negative as that of the initial image.

In order to gauge how well the model was working, I split the udacity provided image and steering angle data(about 24000x12 after data augmentation) into a training – 90% and validation set – 10%. I used 10 epochs with batch size of 64 with shuffling which seemed to work well. The model is training using **Adam optimizer**. The model trained by this data was working well except for certain sharp left turning before the bridge and sharp right turning after the bridge in the simulation.

I further trained the model using the data that I generated using successive_training.py. I generated the data and extracted them in to numpy array to provide as inputs to the model. The generated data contained about 20000 images used without data augmentation. I used training for – 90% of the set and validation s– 10% with 5 epochs with batch size of 64. After this training, the model was working very well on the track 1 without leaving the road.

The video below shows the car completing the track without getting off the road (https://youtu.be/_58fh4GffNU):

Conclusions and further work

After adequate training the model worked as expected. High value of regularization used through drop-out value of 0.5 ensured that model is generalized with effect of extensive training required. The model was not trained on track 2 and did not perform well on the track 2. Additional training might improve the performance of the model on track 2. Python generator was explored in the beginning, but since my laptop was not powerful enough to play the simulation and train simultaneously, I did not used that feature to train further. Additional optimization in the number and depths of layers and number of parameters can be made with further experimentation.

References

[1] c. ai. [Online]. Available: https://github.com/commaai/research/blob/master/train_steering_model.py.

[2] NVIDIA. [Online]. Available: https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf.

[3] [Online]. Available: https://arxiv.org/pdf/1511.07289v1.pdf.

[4] [Online]. Available: http://papers.nips.cc/paper/4878-understanding-dropout.pdf.

[5] [Online]. Available: https://chatbotslife.com/using-augmentation-to-mimic-human-driving-496b569760a9#.4dafs18v3.