Indian Institute of Technology Kanpur

Department of Computer Science and Engineering

Visual Recognition : CS698O

Steering Wheel Angle Prediction

Prepared By:

Dipendra Singh 14223
Sagar Chand 14579

Supervisor:

Professor Vinay Namboodiri

1 Problem Statement

We are working on the problem of predicting steering angle of a car that would be produced by human driver. Given an image, our task is to predict the steering angle that would be taken at that time. The problem can be viewed as a regression problem, with the steering angle being the continuous quantity. This would partially help solving the problem of autonomous car driving.

2 Related Work

Pomerleau (1989) [1] built the Autonomous Land Vehicle in a Neural Network (ALVINN) system. Their model consisted of a fully connected layer. The network predicted actions from pixel inputs applied to simple driving scenarios with few obstacles. However, it demonstrated the potential of neural networks for end-to-end autonomous navigation.
Udacity launched a self-driving car challenge [2] in which the task was to predict steering angle using camera images. Udacity has open-sourced its driving dataset collected for this challenge. Many solutions by teams participating in this challenge used CNN architectures. They report that deeper architectures (such as ResNet, VGGNet) perform worse for this regression problem than relatively shallower CNN architecture. The final leaderboard and achieved validation accuracies are available at [3]. One of the teams [4] used an interesting approach. They proposed using CNNs but instead of using raw images as input, they pass in the delta between consecutive images in the dataset. This is based on the intuition that the displacement of image features is more valuable than the image itself.
In 2016, NVIDIA released a paper [5] in which they used CNN architecture to extract features from the driving frames. The layout of the architecture can be seen in Figure 1. This framework was successful in real-world scenarios, such as highway lane-following and driving in flat, obstacle-free courses. They also used YUV color space instead of conventional RGB. In order to make the system independent of the car geometry, they represented the steering command as $\frac{1}{r}$ where r is the turning radius in meters. They used $\frac{1}{r}$ instead of r to prevent a singularity when driving straight (the turning radius for driving straight is infinity). $\frac{1}{r}$ smoothly transitions through zero from left turns (negative values) to right turns (positive values). Training data contains single images sampled from the video, paired with the corresponding steering command ($\frac{1}{r}$).The training data is therefore augmented with additional images that show the car in different shifts from the center of the lane and rotations from the direction of the road.
Code for the NVIDIA paper was not available. We have tried to implement it in Pytorch environment in our project.

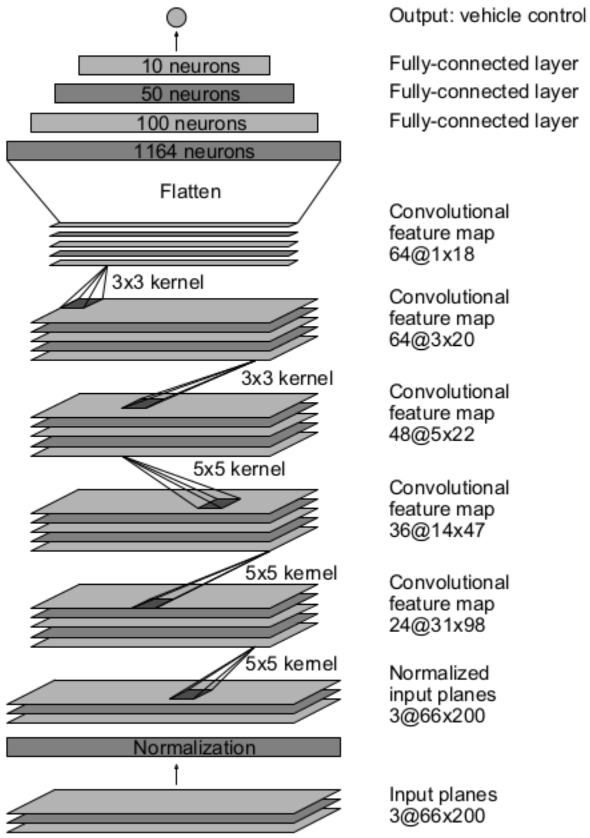

Figure 1: CNN architecture used in [1]. The network contains approximately 27 million connections and 250 thousand parameters.

3 Method

Our method is similar to the one implemented by NVIDIA. We have used their core architecture in our project. This this a 9 layer architecture. The network has about 27 million connections and 250 thousand parameters. Images have been rescaled to 3x66x200 pixels so that they can be fed to our network. The images are first normalized using either batch normalization or instance normalization. Then this is fed to series of convolution layers, which is finally connected to fully connected layers. Learnable affine parameter is also defined as being true or false.
We have used this model and by changing hyper parameters, we have obtained results of 216 networks. Here we have taken error of last 20 batches in case of training error and running rms loss in case of test error.

4 Dataset

We have used the dataset which is provided by Udacity, which is generated by NVIDIAs DAVE-2 System. Specifically, cameras are mounted behind the windshield of the data-acquisition car. Time-stamped video from the cameras is captured simultaneously with the steering angle applied by the human driver. Training data contains single images sampled from the video, paired with the corresponding steering command. Training data set contains 101397 frames and corresponding labels including steering angle. We split this data set into training and test in a 90/10 fashion. The original resolution of the image is 640x480.

5 Experimental Result

This is for RGB color space

Learning rate	Batch Size	Epochs	Normalization	Affine	Training Error	Test Error
0.001	1	1	Batch	False	Bad	Bad
0.001	1	1	Batch	True	0.643	0.026
0.001	1	1	Instance	False	Bad	Bad
0.001	1	1	Instance	True	0.251	0.652
0.001	1	2	Batch	False	Bad	Bad
0.001	1	2	Batch	True	0.144	0.010
0.001	1	2	Instance	False	Bad	Bad
0.001	1	2	Instance	True	0.083	0.005
0.001	1	3	Batch	False	Bad	Bad
0.001	1	3	Batch	True	0.112	0.036
0.001	1	3	Instance	False	Bad	Bad
0.001	1	3	Instance	True	0.074	0.005
0.001	5	1	Batch	False	0.116	0.005
0.001	5	1	Batch	True	0.077	0.005
0.001	5	1	Instance	False	0.582	0.607
0.001	5	1	Instance	True	0.096	0.028
0.001	5	2	Batch	False	0.146	0.021
0.001	5	2	Batch	True	0.071	0.005
0.001	5	2	Instance	False	0.064	0.006
0.001	5	2	Instance	True	0.074	0.033
0.001	5	3	Batch	False	0.039	0.006
0.001	5	3	Batch	True	0.078	0.005
0.001	5	3	Instance	False	0.101	0.004
0.001	5	3	Instance	True	0.075	0.009
0.001	10	1	Batch	False	0.246	0.104
0.001	10	1	Batch	True	0.061	0.011
0.001	10	1	Instance	False	0.095	0.008
0.001	10	1	Instance	True	0.072	0.005
0.001	10	2	Batch	False	0.162	0.044
0.001	10	2	Batch	True	Bad	Bad
0.001	10	2	Instance	False	0.137	0.079
0.001	10	2	Instance	True	0.120	0.006
0.001	10	3	Batch	False	0.046	0.005
0.001	10	3	Batch	True	0.056	0.007
0.001	10	3	Instance	False	0.120	0.008
0.001	10	3	Instance	True	0.068	0.003

Table 1: F1 Scores for various methods

Learning rate	**Batch Size**	**Epochs**	**Normalization**	**Affine**	**Training Error**	**Test Error**
0.002	1	1	Batch	False	Bad	Bad
0.002	1	1	Batch	True	Bad	Bad
0.002	1	1	Instance	False	Bad	Bad
0.002	1	1	Instance	True	0.428	0.733
0.002	1	2	Batch	False	Bad	Bad
0.002	1	2	Batch	True	Bad	Bad
0.002	1	2	Instance	False	Bad	Bad
0.002	1	2	Instance	True	Bad	Bad
0.002	1	3	Batch	False	Bad	Bad
0.002	1	3	Batch	True	Bad	Bad
0.002	1	3	Instance	False	Bad	Bad
0.002	1	3	Instance	True	Bad	Bad
0.002	5	1	Batch	False	0.064	0.004
0.002	5	1	Batch	True	Bad	Bad
0.002	5	1	Instance	False	0.086	0.009
0.002	5	1	Instance	True	Bad	Bad
0.002	5	2	Batch	False	Bad	Bad
0.002	5	2	Batch	True	Bad	0.445
0.002	5	2	Instance	False	Bad	Bad
0.002	5	2	Instance	True	0.043	0.010
0.002	5	3	Batch	False	Bad	Bad
0.002	5	3	Batch	True	0.183	0.246
0.002	5	3	Instance	False	Bad	Bad
0.002	5	3	Instance	True	Bad	Bad
0.002	10	1	Batch	False	Bad	Bad
0.002	10	1	Batch	True	Bad	Bad
0.002	10	1	Instance	False	0.073	0.013
0.002	10	1	Instance	True	0.059	0.007
0.002	10	2	Batch	False	Bad	Bad
0.002	10	2	Batch	True	0.656	0.347
0.002	10	2	Instance	False	Bad	Bad
0.002	10	2	Instance	True	Bad	Bad
0.002	10	3	Batch	False	Bad	Bad
0.002	10	3	Batch	True	Bad	Bad
0.002	10	3	Instance	False	Bad	Bad
0.002	10	3	Instance	True	Bad	Bad

Table 2: Training and test error for various parameters

Learning rate	Batch Size	Epochs	Normalization	Affine	Training Error	Test Error
0.003	1	1	Batch	False	Bad	Bad
0.003	1	1	Batch	True	Bad	Bad
0.003	1	1	Instance	False	Bad	Bad
0.003	1	1	Instance	True	Bad	Bad
0.003	1	2	Batch	False	Bad	Bad
0.003	1	2	Batch	True	Bad	Bad
0.003	1	2	Instance	False	Bad	Bad
0.003	1	2	Instance	True	Bad	Bad
0.003	1	3	Batch	False	Bad	Bad
0.003	1	3	Batch	True	Bad	Bad
0.003	1	3	Instance	False	Bad	Bad
0.003	1	3	Instance	True	Bad	Bad
0.003	5	1	Batch	False	0.054	0.009
0.003	5	1	Batch	True	0.047	0.004
0.003	5	1	Instance	False	0.060	0.004
0.003	5	1	Instance	True	0.115	0.004
0.003	5	2	Batch	False	0.050	0.005
0.003	5	2	Batch	True	Bad	Bad
0.003	5	2	Instance	False	0.043	0.005
0.003	5	2	Instance	True	0.167	0.015
0.003	5	3	Batch	False	Bad	Bad
0.003	5	3	Batch	True	Bad	Bad
0.003	5	3	Instance	False	Bad	Bad
0.003	5	3	Instance	True	Bad	Bad
0.003	10	1	Batch	False	0.075	0.004
0.003	10	1	Batch	True	0.100	0.089
0.003	10	1	Instance	False	0.112	0.055
0.003	10	1	Instance	True	0.119	0.006
0.003	10	2	Batch	False	0.091	0.004
0.003	10	2	Batch	True	0.071	0.004
0.003	10	2	Instance	False	0.120	0.007
0.003	10	2	Instance	True	Bad	Bad
0.003	10	3	Batch	False	0.074	0.005
0.003	10	3	Batch	True	Bad	Bad
0.003	10	3	Instance	False	0.068	0.006
0.003	10	3	Instance	True	Bad	Bad

Table 3: Training and test error for various parameters

Color space changed to YCbCr

Learning rate	Batch Size	Epochs	Normalization	Affine	Training Error	Test Error
0.001	1	1	Batch	False	Bad	0.408
0.001	1	1	Batch	True	0.070	0.007
0.001	1	1	Instance	False	Bad	Bad
0.001	1	1	Instance	True	0.171	0.034
0.001	1	2	Batch	False	Bad	Bad
0.001	1	2	Batch	True	0.092	0.031
0.001	1	2	Instance	False	Bad	Bad
0.001	1	2	Instance	True	0.093	0.126
0.001	1	3	Batch	False	0.116	0.070
0.001	1	3	Batch	True	0.024	0.007
0.001	1	3	Instance	False	0.076	0.007
0.001	1	3	Instance	True	0.042	0.006
0.001	5	1	Batch	False	0.223	0.033
0.001	5	1	Batch	True	Bad	0.743
0.001	5	1	Instance	False	Bad	Bad
0.001	5	1	Instance	True	0.045	0.005
0.001	5	2	Batch	False	0.063	0.009
0.001	5	2	Batch	True	0.058	0.010
0.001	5	2	Instance	False	0.055	0.004
0.001	5	2	Instance	True	0.087	0.012
0.001	5	3	Batch	False	Bad	Bad
0.001	5	3	Batch	True	0.093	0.006
0.001	5	3	Instance	False	0.075	0.005
0.001	5	3	Instance	True	0.962	0.603
0.001	10	1	Batch	False	0.485	0.198
0.001	10	1	Batch	True	0.064	0.008
0.001	10	1	Instance	False	0.133	0.033
0.001	10	1	Instance	True	0.051	0.010
0.001	10	2	Batch	False	0.199	0.184
0.001	10	2	Batch	True	0.062	0.009
0.001	10	2	Instance	False	0.074	0.012
0.001	10	2	Instance	True	0.080	0.013
0.001	10	3	Batch	False	Bad	Bad
0.001	10	3	Batch	True	0.062	0.006
0.001	10	3	Instance	False	0.077	0.015
0.001	10	3	Instance	True	0.064	0.008

Table 4: Training and test error for various parameters

Learning rate	**Batch Size**	**Epochs**	**Normalization**	**Affine**	**Training Error**	**Test Error**
0.002	1	1	Batch	False	Bad	Bad
0.002	1	1	Batch	True	Bad	Bad
0.002	1	1	Instance	False	Bad	Bad
0.002	1	1	Instance	True	Bad	Bad
0.002	1	2	Batch	False	Bad	Bad
0.002	1	2	Batch	True	Bad	Bad
0.002	1	2	Instance	False	Bad	Bad
0.002	1	2	Instance	True	Bad	Bad
0.002	1	3	Batch	False	Bad	Bad
0.002	1	3	Batch	True	Bad	Bad
0.002	1	3	Instance	False	Bad	Bad
0.002	1	3	Instance	True	Bad	Bad
0.002	5	1	Batch	False	0.076	0.006
0.002	5	1	Batch	True	Bad	Bad
0.002	5	1	Instance	False	Bad	Bad
0.002	5	1	Instance	True	0.425	0.172
0.002	5	2	Batch	False	Bad	Bad
0.002	5	2	Batch	True	Bad	Bad
0.002	5	2	Instance	False	Bad	Bad
0.002	5	2	Instance	True	Bad	Bad
0.002	5	3	Batch	False	Bad	Bad
0.002	5	3	Batch	True	Bad	Bad
0.002	5	3	Instance	False	Bad	Bad
0.002	5	3	Instance	True	Bad	Bad
0.002	10	1	Batch	False	0.060	0.007
0.002	10	1	Batch	True	0.084	0.025
0.002	10	1	Instance	False	Bad	Bad
0.002	10	1	Instance	True	Bad	Bad
0.002	10	2	Batch	False	0.078	0.006
0.002	10	2	Batch	True	0.044	0.005
0.002	10	2	Instance	False	Bad	Bad
0.002	10	2	Instance	True	Bad	Bad
0.002	10	3	Batch	False	0.110	0.059
0.002	10	3	Batch	True	0.152	0.071
0.002	10	3	Instance	False	Bad	0.700
0.002	10	3	Instance	True	Bad	Bad

Table 5: Training and test error for various parameters

Learning rate	Batch Size	Epochs	Normalization	Affine	Training Error	Test Error
0.003	1	1	Batch	False	Bad	Bad
0.003	1	1	Batch	True	Bad	Bad
0.003	1	1	Instance	False	Bad	Bad
0.003	1	1	Instance	True	Bad	Bad
0.003	1	2	Batch	False	Bad	Bad
0.003	1	2	Batch	True	Bad	Bad
0.003	1	2	Instance	False	Bad	Bad
0.003	1	2	Instance	True	Bad	Bad
0.003	1	3	Batch	False	Bad	Bad
0.003	1	3	Batch	True	Bad	Bad
0.003	1	3	Instance	False	Bad	Bad
0.003	1	3	Instance	True	Bad	Bad
0.003	5	1	Batch	False	Bad	Bad
0.003	5	1	Batch	True	Bad	Bad
0.003	5	1	Instance	False	0.037	0.003
0.003	5	1	Instance	True	Bad	Bad
0.003	5	2	Batch	False	Bad	Bad
0.003	5	2	Batch	True	0.071	0.005
0.003	5	2	Instance	False	Bad	Bad
0.003	5	2	Instance	True	Bad	Bad
0.003	5	3	Batch	False	0.086	0.010
0.003	5	3	Batch	True	Bad	Bad
0.003	5	3	Instance	False	Bad	Bad
0.003	5	3	Instance	True	Bad	Bad
0.003	10	1	Batch	False	0.051	0.007
0.003	10	1	Batch	True	0.099	0.016
0.003	10	1	Instance	False	0.037	0.008
0.003	10	1	Instance	True	0.053	0.027
0.003	10	2	Batch	False	0.060	0.007
0.003	10	2	Batch	True	0.067	0.013
0.003	10	2	Instance	False	Bad	Bad
0.003	10	2	Instance	True	0.080	0.008
0.003	10	3	Batch	False	0.100	0.005
0.003	10	3	Batch	True	Bad	Bad
0.003	10	3	Instance	False	0.078	0.008
0.003	10	3	Instance	True	Bad	Bad

Table 6: Training and test error for various parameters

6 Inference

By looking at the result it can be inferred that as the learning rate increases, our errors(both training and test) become very high. Thus for this model, higher learning rate should be avoided for both the color spaces. It is also observed that error is lower when learn-able affine parameters is true. Error in YCbCr color space was found to be lower than RGB color space. Though no significant difference was seen with the normalization, in few places Batch normalization was found to show lower error than Instance normalization. In general, error showed a decreasing trend with increase in number of epochs. Again no general trend could be inferred from change in batch sizes, but it was seen that model with batch sizes 5 or 10 showed better results than models with batch size of 1.

References

[1] D. A. Pomerleau. Alvinn, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, Computer Science Department, 1989.

[2] Udacity Self-Driving Car Challenge#2:https://medium.com/udacity/challenge-2-using-deep-learnin g-to-predict-steering-angles-f42004a36ff3

[3] Udacity Challenge Final Ranking and Leaderboard https://github.com/udacity/self-driving-car/tree/master/challenges/challenge-2

[4] 2nd Rank in Udacity Challenge, Team rambo. https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/rambo

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars.arXiv preprint arXiv:1604.07316, 2016.