Activation Function: Key to Cloning from Human Learning to Deep Learning

Pranit Gopaldas Shah¹ and Hiral Pranit Shah²

¹ Parul Institute of Engineering and Technology

Received: 16 December 2019 Accepted: 5 January 2020 Published: 15 January 2020

Abstract
Maneuvering a steady on-road obstacle at high speed involves taking multiple decisions in split seconds. An inaccurate decision may result in crash. One of the key decision that needs to be taken is can the on-road steady obstacle be surpassed. The model learns to clone the drivers behavior of maneuvering a non-surpass-able obstacle and pass through a surpass-able obstacle. No data with labels of ?surpass-able? and ?non-surpass-able? was provided during training. We have development an array of test cases to verify the robustness of CNN models used in autonomous driving. Experimenting between activation functions and dropouts the model achieves an accuracy of 87.33

Index terms— BC, end-to-end learning, saliency map, computer vision, behavioral cloning, autonomous vehicles, self-driving, obstacle mitigation

1 Introduction
Affordability of powerful computational hardware and advances in deep learning techniques has made vision-based autonomous driving an active research focus within the transport industry. There are considerable drawbacks in the techniques to overcome, even though the research worldwide has already taken giant leap. Foremost downside is the inability to explicitly model each possible scenario. Driving needs responding to a large variety of complex environmental conditions and agent behaviors.

End-to-end method and perception-driven method are the two popular vision-based paradigms for self-driving cars. A perception-based method lacks selflearning ability and all features including task plans are manually hand crafted. This is the major disadvantage of perception-based method.

End-to-end Behavior cloning (Off-policy imitation learning) provides an alternative to traditional modular approach by simultaneously learning both perception and control using deep network.

Maneuvering each and every steady on-road obstacle either surpass able or non-surpass-able is cost intensive. Again maneuvering a steady on-road obstacle at high speed involves taking multiple decisions in split seconds. An inaccurate decision may result in crash. One of the key decision that needs to be taken is can the steady on-road obstacle be surpassed.

Clearly, an autonomous driving vehicle successfully navigating through the streets should be able to follow the roadway as well as maneuver only in cases required. If the autonomous vehicle can surpass a steady on-road obstacle without being unstable it must do so. Therefore, we here in propose an improved convolution neural network model. Overall, this research work makes the following contributions:

? Provides evaluation method for popular learning models by defining test cases to mitigate an onroad obstacle.
? Identify, evaluate and validate the configuration producing optimum results by performing combination of activation function and dropout. ? Improve prediction accuracy by validating statistical data with visual saliency map.

The paper has been organized as follows: Section 2 gives an overview of related work done in past and present. Section 3 describes the methodology used. Section 4 Experimental Setup, Results and Discussion. Finally, Section 5 concludes this paper.
B) END-TO-END LEARNING METHODS

II. Related Work

Perception-driven method and End-to-end method are the two popular vision-based paradigms for self-driving cars. Both the perception and end-to-end methods have been reviewed extensive through literature study and presented here. However, the key aspect of autonomous driving is the problem of object detection itself. Hence in the later part of this section we review our study on object detection in the context of Deep Learning.

a) Perception-driven Learning Methods

Traditional Perception based method has made remarkable achievements, in past decades, in the field of self-driving cars. Several methods of detections have been proposed to generate a description of the local environment. However, the key aspect of autonomous driving is the problem of object detection itself. Hence in the later part of this section we review our study on object detection in the context of Deep Learning.

i. Vision Based Detection

Object location through enclosing boxes has been adopted by researchers as a part of object detection task. This bounding box can be anything from a steady road signs, traffic signs, moving cars, moving bicycles, etc. The model is trained with labeled data of objects. The key factor in case of self-driving car is an ability to identify if there is an obstacle and at what distance. It is immaterial to have the exact location of the bounding box.

Kwang Eun An [10] used Deep Convolutional Neural Network (DCNN) for image classification into obstacle/pothole or non-pothole. The method compared Inception_v4, Inception_ResNet_v2, ResNet_v2_152 and MobileNet_v1 for comparison between color and grayscale images. This method is limited in its efficiency in processing single frame of image.

Canny edge detection or Hough transformation area is implemented to locate lane position through many lane detection methods. There are no specific geometric restrictions required to identify uneven lane boundaries in these methods.

ii. 3D Reconstruction

There are three major approaches for 3D reconstruction of obstacle/pothole each with its own drawback. 1. Chang et al. [19] used a Grid based processing technique. Here a surface receives laser incidents and digitally implements the bounced back pulse to generate a precise. The output was accurate, however this was expensive. Li et al. [20] used infrared laser line projectors, a digital camera and a multi-view coplanar scheme for calibrating the lasers. The method plotted more feature points in the cameras point of view and was much more cost effective. 2. Wang [21] used a series of cameras. This method generated a 3D surface model through a series of 2D images captured. High computation requirement was key backdrop. 3. Xbox Kinect sensor was used by Joubert et al. [22] and Moazzam et al. [23]. The method could not minimize the error and power for computing, even though equipment price was minimized.

iii. Vibration based detection Umang Bhatt [26] combined accelerometer, gyroscope, location and speed data to classify road condition and detect potholes/obstacles. SVM with radial basis function (RBF) kernel was used for road condition classification. SVM and gradient boost were used for pothole/non-pothole classification. Failure attributed includes inability to accurately classify between good and bad road due to insufficient data for all road types. The key drawback was inability to distinguish between a bump, a manhole or a pothole.

One key takeaway from all the above work done in perception based learning provides proactive data to the driver regarding the obstacle. These methods do not provide any method for routing through obstacle. Again, there is not classification if an obstacle can be surpassed or not.

b) End-to-end Learning Methods


In the scenario of autonomous driving one of the key requirement is an ability to identify salient objects. The key differentiator for this architecture was its ability to identify salient objects without the need for hand-coded rules and instead learn by observing. However Dave’s PilotNet model admits that the convolutional layers were chosen empirically and hence the performance was not sufficiently reliable to provide a full alternative to more modular approaches to off-road driving.

Deconvolution Based Matthew [28] presented a method for mid and high-level feature learning like corners, junctions and object parts. Matthew [28] resolves the two fundamental problems found in image descriptors like SIFT, HOG or edge gradient calculators followed by some histogram or pooling operation. First being invariance to orientation and scale. Secondly a CNN models inefficiency in training each model with respect to input. Visualization available is for one activation per layer.
8 iii. Layer-wise Relevance Propagation

Alexander Binder [29] implemented Layer-wise relevance propagation to compute scores for image pixels and image regions denoting the impact of the particular image region on the prediction of the classifier for one particular test image. Alexander Binder [29] demonstrated controlling the noisiness of the heatmap, however an optimal trade between numerical stability and sparsity/meaningfulness of the heatmap was kept as item for future work.

Salient object based methods do identify the key features that impact the steering angles. However these methods have not been explicitly developed/tested for identifying if an obstacle can be surpassed or not.

To overcome the above mentioned limitations, we propose to perform extensive training and testing for a neural network to clone an obstacle mitigation policy. Even though there is a great deal of work and literature on the task of steering angle prediction, our goal is not to propose yet another method for prediction, but rather provide a different perspective on on-road steady obstacles mitigation model. A model to not only detect an on-road steady obstacle but also to predict the obstacle can be surpassed to avoid unnecessary maneuvering.

9 III.

10 Methodology

This section provides the details on CNN models used for validation and steps performed on data for accurate prediction.

11 a) Preprocessing

Network Model continuously predicts the steering angle to clear all test cases with an input of raw pixels incorporating attention in an end-to-end manner. It’s important that our experiments and results are independent of car geometry, hence we represent steering command as inverse turning radius \( ?? \) ? ? ! ( \( r \) is turning radius at time timestamp \( t \)). We use inverse to prevent numerical instability, singularity, and smooth transitions through zero from left turns to right turns. The relation between steering angle \( ?? \) ? ? and inverse turning radius can be given as\( ? ? = \delta \) ?? ? " ????????????? ???????????? ( ? ? ? ? ) = ? ? ? ? ? ? ? ? ? ! ( 1 + ? ? ????????? ?? ?? 2 . (1)\)

Where \( ?? \) ? ? in degree and \( ?? \) ? ? (m/s) is a steering angle and velocity at time \( ?? \), respectively. \( ?? \) ? ? , ?? ??????? and ?? ? ? are vehicle-specific parameters. ?? ? ? is steering ratio between the turn of the steering and the turn of the wheels. ?? ????????? represents the relative motion between the front and rear wheels.

b) Data Bias Removal The general tendency of every driver is to drive as steady as possible without lot of maneuvering. However in case of behavioral cloning if the driver drives steady during training then all the model will learn is to maintain a zero steering angle. Such a training would generate results bias to zero output. In order to avoid output bias towards a -ve/+ve steering angle driving the training is done for complete track in clockwise and then anticlockwise direction over track. In fact additional recovery training tracks are also done where the car is take off the center lane and then recovered to back. The data bias is removed by trimming samples per bin as shown in Figure 3.

12 Experimental Setup, Results and Discussion

This section presents the basic description of experimental setup for data collection, training and testing. We elaborate further on the configuration of hardware and software used. Later we enlist the training cases and test cases for model evaluation and the evaluation criteria. Eventually we enlist the results from our experiment.

Model training and testing is performed in a virtual environment Unity 2017 in the interest of research cost. Other software tools include Visual studio for Unity, Atom, Jupyter Notebook and Anaconda. GitHub for online code repository and Google Colab platform for online code execution. Programming is done in c# and python. Multiple packages including OpenCV, numpy, matplotlib, Keras (model, optimizer, layers), pandas and sklearn are used. Hardware includes my laptop with Intel i5 processor@1.27GHz, 8GB RAM and Intel HD Graphics family with 2GB RAM.

Virtual 3D models of non-surpass-able and surpass-able obstacles are created in unity as shown in Figure 4 below. an array of multiple surpass-able obstacles. The model is supposed to pass through the obstacles with minimum vehicle instability and maintain a steady drive to call it has passes this test case. Starting Bottom Left 1 st : TC06: The model is supposed to maneuver through an array of left and right non-surpass-able obstacles and retain a steady drive to call it has passes this test case. 2 nd : TC07: The track has an array of both multiple left and right non-surpassable and surpass-able obstacles. The model is supposed to pass through surpass-able obstacles and maneuver through the obstacle to call it pass. 3 rd : TC08: In this case an unknown bridge, which has a different track, has to be crossed to call this case passes. 4 th : TC09: In this case an unknown unpaved path has to be avoided by the model. The model has not been trained for this behavior. 5 th : TC10: In model is expected to clear all the obstacles, pass through unknown bridge and unpaved path all in a single stretch.

13 b) Model Accuracy Calculation

Each model is trained in 4 cases and tested for 10 cases. This paper represents two deep learning models tested with a combination of activation function and dropout on same database. Table 2 below shows model used, code
assigned for ease, configurations used and val_loss achieved are 50% and 30% respectively with heavy processing
time of 12000s and 8280s respectively. Model P1 achieves the highest accuracy consuming the processing time
of 4478 seconds.

Model P1, PilotNet with elu and no-dropout, performed a self-recovery in test cases 6 and 7 as listed in Table ?? below

14 Conclusion

In this paper, we presented and compared two most popular autonomous driving methods including DroNet
and PilotNet. We experimented with combinations of different activation functions with and without dropout. The
experiment has demonstrated the PilotNet model P1 is able to learn the entire task of non-surge-able obstacle
maneuvering and passing through a surge-able obstacle. The experiment has provided us with a clear insight
into effect of each activation function and dropout on steering angle prediction. PilotNet model P1 has the
highest prediction accuracy, lowest val_loss and reasonable processing time with best visual saliency map for
obstacles with current dataset. The experiment has clearly concluded that PilotNet, with activation function elu
without dropout, outperforms all other models and configurations.
The system learned to mitigate through an obstacle without the need of explicit surge-able and non-surge-able
obstacle labeling during training.

In the future work, we would like to optimize PilotNet to improve prediction accuracy. We would also like to
introduce a custom-Net that would outperform all current autonomous driving methods.

Figure 1: Fig. 1 :Fig. 2 :
Figure 2: Fig. 3:

![Histogram graph](image)

Figure 3:

![Car and obstacle](image)

Figure 4: Fig. 4:

![Car model](image)
Figure 5: Fig. 5:

Figure 6: Fig. 6:
Figure 7: Fig. 7:

Figure 8: Fig. 8:

Figure 9: Figure 8
Number of test cases model passed: P  
Score for a test cases in which model self-recovered: SR  
Model Accuracy: MA  

\[ SR = \frac{SR_o}{T_o} \]  

<table>
<thead>
<tr>
<th>Model</th>
<th>Input Data after removing bias</th>
<th>Training Samples</th>
<th>Validation Samples</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>DroNet</td>
<td>99308904</td>
<td>3904</td>
<td>977</td>
<td></td>
</tr>
<tr>
<td>PilotNet</td>
<td>99308904</td>
<td>3904</td>
<td>977</td>
<td></td>
</tr>
</tbody>
</table>

below clearly
Figure 14: Table 2:
[Bojarski et al. ()], Mariusz & Bojarski, Davide & Testa, Dworakowski, & Daniel, Bernhard & Firner, Flepp, & Beat, Goyal, & Prasoon, Jackel, & Lawrence, Monfort, & Mathew, Muller, & Urs, Zhang, & Jiakai Zhang, & Xin, Jake & Zhao, Karol Zieba. 2016. (End to End Learning for Self-Driving Cars)


[Li et al. ()] A real-time 3D Scanning System for pavement distortion inspection, measurement science and technology, Q Li, M Yao, Yao, B Xu. 2009. p.


[Srivastava et al. ()] Analysis and Improvements on Current Pothole Detection Techniques. 1-4.10.1109/ICCE.2018.8326142


