

Visualization of End-to-End Autonomous Driving Model Based on Deep Neural Network

1w143137-4 Meng Tianyi

Supervisor: Prof. Tetsuya Ogata

Abstract: Recently, Deep Neural Network(DNN) extensive use for autonomous driving studies, as End-to-End learning from camera to steering commands directly. Because it optimizes the all processing simultaneously, it is hard to understand. To solve this problem, there are some methods of neural network visualization, which can explain the internal procedures of certain convolutional network. However, these methods are quite differently appearing, and have their own strengths and weakness. In this study, we applied several methods on certain pre-trained End-to-End model to analyze the End-to-End model, trying to comprehend and resolve internal procedure of computing steering commands according to input data.

Keywords: autonomous driving, Neural Network visualization, End-to-End learning

1. Introduction

Recently, Deep Neural Network extensive use for autonomous driving studies, as End-to-End learning from camera to steering commands directly. Therefore, it should be comprehensible even though it works as a black box and optimizes all processes simultaneously. In the previous work, some of studies decompose the problem into several sub-problems. We can consider that if there would be a End-to-End model that optimizes all those problems simultaneously. However, this kind of integrated model will also become more complicated because of the feature of multi dimensional of convolutional neural networks.

To solve this problem, we use several methods to analyze the End-to-End model, trying to comprehend and resolve internal procedure of computing steering commands according to input data. In addition, we also did a comparison of certain methods in a mathematically.

At the end of paper, we also discussed the feasibility of End-to-End model that applying to studies of autonomous driving.

2. Purpose of Study

The purpose of this study is comparison of neural network visualization and understanding the procedure of recognition of neural network.

Firstly we generate saliency map (map of sensitivity) with certain pre-trained End-to-End model and input image, and compare to each other through out features of each via observation. In studies of autonomous driving, hence to researchers prefer to know the most activated parts, we need to find out a sharper method.

Moreover, by observing activation in saliency map, we can assume the concentration of neural network. For example, the neural network may focus on traffic lights while passing through one. We can understand the internal strategy and procedure of certain neural network.

3. Methods of visualization

In this study, we applied several methods to visualize the model in order to analyze internally by generating saliency maps.

Vanilla gradients[1] compute saliency map by gradients. It approximate convolutional neural network into a linear function by computing the first-order Taylor expansion.

Guided Backpropagation is considered that filters out values corresponding to negative entries of both top gradient and bottom data. It combines DecovNet and backpropagation together, which can prevent backward flow of negative gradients decrease the activation of the higher layer unit[2].

Integrated gradients[3] combines the Implementation Invariance of Gradients along with the Sensitivity of techniques. Firstly set a baseline image(black image). Integrated gradients are obtained by cumulating these gradients.

4. End-to-End Model

The End-to-End model which for analysis is composed of 9 convolutional layers and 3 fully connected layers. The last fully connected layers output are composed of 12 outputs.

While running, data is inputted into neural network as a speed of 8fps, converted to a 12 dimensional vector as output. System compute a proper steering command. Multi-dimensional output can help driving agent to make proper decision by it self, using these parallel direction control information and probabilities when there are no extra direction chosen instruction from human beings. The procedure of system is shown in Figure 1.

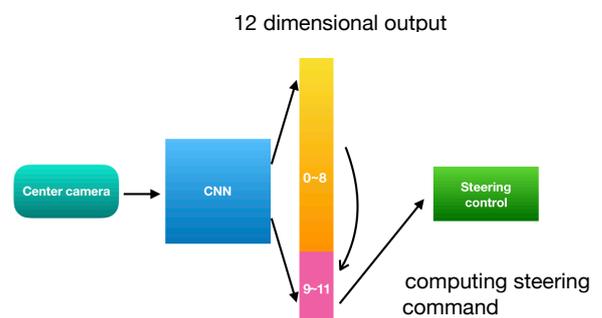


Figure 1 Network generate steering commands.

5. The experiments for comparing methods and analyze the model

In this study, the datasets comprises of a pre-trained End-to-End learning model and a set of steering data. The model is trained in previous work.

In this study, we observe the End-to-End learning model by using certain visualization methods. According to the results, We resolve the internal process of computing steering commands in the model, and compare each methods of visualization.

For comparison of visualization methods, we apply each methods to visualize all data by generating saliency map. Moreover, in order to observe better, we optimize saliency maps by using SmoothGrad[4], which can significantly reduce noise of original saliency maps. The flow of procedure is shown in Figure 2.

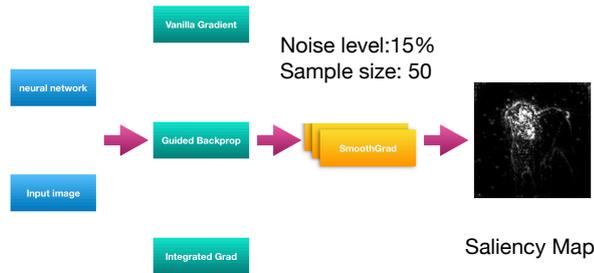


Figure 2. Flow of procedure.

In this study, we estimate each visualization algorithm from distribution of saliency and distribution of exclusive saliency in resulting maps.

6. Result: Flow of Steering

Tendency of variation of concentration also can be observed in such result. The focus is changing following steering command computed by model. The model focused on traffic light after it turns red, shift to background views after it turns green, focus more on views after traffic disappeared from field of vision, and always focus on lanes. The portion is shown in Figure 3.

7. Estimation of results

To estimate results, we generated Saliency Distribution and Peak Map. In case to reduce influence of error, we chose a portion which have features that we concerned important to recognition and responded correctly.



Figure 3. Portion and Saliency Map of Steering

Portions, their Saliency Distribution charts and Peak Maps are shown in Figure 4.

8. Analyze Failure of Steering

In the dataset, system had once ignored a red light with a slight deceleration, and once turned right while driving at a cross without right way.

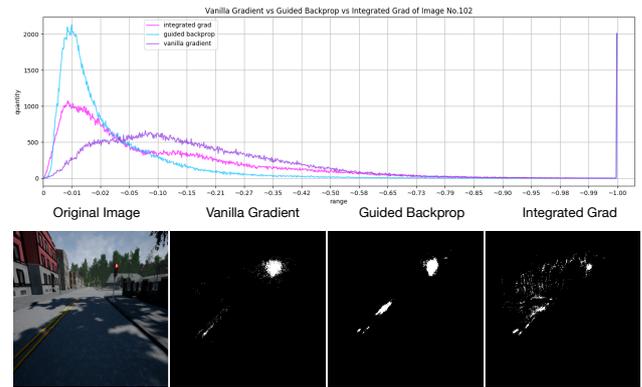


Figure 4.3 Saliency Distribution and Peak Map while system react to traffic lights.

System ignored red light at 58 second in the dataset. Compare screenshot and Saliency Map of “ignoring red lights” to shot of “waiting for red light” and “pass through green light”, we can see that it is closer to green light one.

System turned right at 60 second while driving at a cross without right way in the dataset. Compare screenshot and Saliency Map of incorrect rotation and correct ones, we can see that Saliency Maps are quite similar.

9. Conclusions and Prospections

According to the results of studies, we summarized

	Vanilla Gradient	Guided Backprop	Integrated Grad
Sharpness	△	◎	△
Noise	△	◎	○
Visual effect	△	○	◎
Peak Focused	○	◎	×
Clarity	△	○	○

Table 5.1 Features of visualization methods

strengthens and weakness of each visualization methods (Table 1).

Moreover, we can see that the pre-trained End-to-End model is similarly following a strategy as multiple network model. For example, the model shows a high sensitivity to traffic lights while passing nearby. However, the model also shows a overfitting trend that pay too much attention on background. Besides, the model shows a deficiency of recognition of right side of road.

Furthermore, we applied Saliency Distribution and Peak Map to estimate the results of visualization. Even though these methods help us to compare in many facet, ultimately we evaluated each visualization methods through visual intuition. A quantized standard of evaluation to estimate methods of visualization is anticipated in future’s works.

[1] K. Simonyan, Vedaldi, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034v2, 2014.

[2] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. String For Simplicity: The All Convolutional Net. arXiv preprint arXiv: 1412.6806v3, 2015.

[3] M. Sundararajan, A. Taly and Q. Yan. Axiomatic attribution for deep networks. arXiv preprint arXiv:1703.01365, 2017.

[4] D. Smilkov, N. Thorat, B. Kim, F. Viegas and M. Wattenberg. SmoothGrad: removing noise by adding noise. arXiv preprint arXiv: 1706.03825v1, 2017.