Summary of
Learning a Deep Neural Network Policy for End-to-End Control of Autonomous Vehicles

I. Introduction

Why? Over 90% of car accidents stem from human error, making self-driving cars a safer alternative. Traditional autonomous car pipelines involve separate tasks like SLAM, perception, path planning, and control. Emerging deep learning (DL) methods raise the question: Can each task or even the entire pipeline be realized with a single DL model?

Which Problem? Rather than engineering each pipeline component separately, this paper [1] proposes a simple end-to-end system using deep learning. The authors focus on encoding a control command for steering a car, using only image frames from a single camera as input. Unlike traditional approaches, tasks such as localization, lane detection, and path planning are not explicitly considered. The controller, a deep neural network, maps camera frames to the steering angle control command.

Contribution. This work demonstrates the feasibility of end-to-end control for autonomous vehicles in simplified settings, offering an overview of implementing a basic deep-learning system. Targeting researchers in classical control techniques, especially those in autonomous vehicle research, this paper provides valuable insights.

Relation to Course Topics. The lecture on "Techniques for Self-Driving Cars" delves into the conventional self-driving car pipeline, tackling tasks separately to generate control commands. The current trend, however, leans towards end-to-end control, employing deep learning models exclusively. This paper introduces this novel approach, departing from the traditional method of individually engineering each pipeline component.

II. Approach

A control policy, which is mapping from observations $\mathbf{z}$ to control actions $\mathbf{u}$, can be encoded using CNNs through a process known as *policy search*. This paper employs *guided policy search*, where human drivers provide guidance based on image frames ($\mathbf{z}$) to generate steering wheel angles ($\mathbf{u}$).

The *CARSIM* simulation environment is utilized for data collection, with the car driving at a constant speed of 60 km/h in a simplified scenario. The simulation is sampled at 12 FPS to balance relevant information and computational efficiency. Raw image frames are RGB with a resolution of 1912x1036 while steering angles range from $-45°$ to $40°$. Normalization of pixel values (0 to 1) and steering angles (-1 to 1) aids faster convergence of the loss function. Frames are scaled down to 190x100 resolution to manage computational demands.

The deep neural network (DNN) search is an iterative process. Chosen CNN architecture has four hidden layers, comprising three convolutional layers and one fully connected layer. The output layer of the model has one neuron and the output is the steering wheel angle.

The activation function is a rectified linear unit (ReLU), and the loss function is a mean squared error (MSE). Dropout regularization is applied after each layer. To optimize the model, multiple optimizers are employed, including SGD, NAG, and Adam. The network is initialized with random weights and biases.

III. Experiments

The model, trained on a 10800-sample dataset using SGD, NAG, and Adam solvers with a batch size of 128, exhibits low converged loss mean after 8000 iterations, indicating successful fitting to training data (Figure 5, Table I). Without validation data for hyperparameter optimization, training ceased after 5000 iterations to prevent overfitting.

When applied to the test data, all of these three models give good results (Figure 6, Table II). Although the model with the Adam solver had the best loss curve on training data, the model with the NAG solver performed best on the test set. This means that the NAG model generalizes better and that the Adam model overfitted training data more.

Figure 7 shows 6 feature maps after each convolutional layer. It can be seen that convolutional layers (i.e. weights) learned important features of the road frames during the training, which is expected since training and test results are good.

This paper presented a simple CNN model for vehicle steering. They showed model performs well, and is able to learn important road features. This shows that hand-engineering features is not required.

In future work, the author aims at recording more data and constructing deeper and more powerful models. This paper is a good indication that end-to-end control is possible to realize with a single DL model.

REFERENCES

[1] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, and J.K. Hedrick. Learning a deep neural net policy for end-to-end control of autonomous vehicles. In *Proc. of the IEEE American Control Conference (ACC)*, 2017.