Project 1 - Imitation Learning

Meier Michael

02. 04. 2025

Introduction

The goal of this project was to train a first build model with imitation learning, after the first evaluation of the model a Dagger algorithm was implemented to gain more data from the expert and improve the model trough further training with the produced data. The task is to train a model that keeps a car on the track by steering either left or right and by speeding up or slowing down. In the first part of this report the used model and the training process will be described. After that the reusults will be presented and the report concludes with a short summary of the results.

Methods

0.1 Model

To solve the task described above, a convolutional neural network (CNN) was implemented. The network consists out of three convolutional layers, followed by two three linear layers. After each convolutional layers a Batch Normalization was applied as also a MaxPooling and then a ReLU activation function. This activation function was also used after each linear layer except for the last one.

0.2 Training with Imitation Learning

The training process was based on the behavioral coloning approach. The model was trained on the provided dataset with a batchsize of 64. An AdamW optimizer was used with a learning rate of 0.00001 and a CrossEntropy loss function was used for the final result. After nine epochs the loss of the model was stagnating at about 0.77 as also the accuracy (0.79).

0.3 Training with Dagger Algorithm

The Dagger algorithm uses the desicion of a expert model to create new data samples. In the implementation a $\beta = 0.9$ was used for the decision if the expert model or the current model should be used for the decision. These datasamples where then appended to the dataset and the model was trained again with the same parameters as in the first training with imitation learning. As better results were achieved here with over 15 epochs, they where increased.

Results

The results of the first trained model (without dagger) shows a mean performace of 690 for the given task. Here the model was tested over 10 episodes where it achieved scores between 366 and 892. This result improves slightly by the further training with the Dagger algorithm. The mean performance of the model was evaluated again over 10 episodes. The results have to be handled with care, as the runs of the model where randomized and the results vary in a range of 100 points.

Conclusion

Imitation learning is a powerful tool to train a model for a given task and it can be improved with the Dagger algorithm. The results show that the model was able to learn the behavior of the expert even if the results are not as good as expected. For further improvements, a hyperaparameter search could help to improve training parameters or the β value could be adjusted depending on the behavior of the model.