Paper: Imitating Driving Behavior with Generative Adversarial Networks

Summary

In this paper, the authors demonstrate the effectiveness of deep imitation learning (IL) as a means of training driver models that perform realistically over long time horizons, while simultaneously capturing microscopic, human-like behavior. They have 2 major contributions. First, they extended Generative Adversarial Imitation Learning to the optimization of recurrent policies. Second, they applied this technique to the creation of a new, intelligent model of highway driving that outperforms the state of the art on several metrics.

Their learned policy must be able to capture human driving behavior which involves non-linearity, high-dimensionality, and stochasticity. To address the first and second points, they represent all learned policies using neural networks. To address the third point, they interpret the network's real-valued outputs as the mean and logarithm of the diagonal covariance of a Gaussian distribution. They represent recurrent policies using Gated Recurrent Unit (GRU) networks due to their comparable performance with fewer parameters than other architectures.

They use Trust Region Policy Optimization (TROP) to learn their human driving policies. TROP updates policy parameters through a constrained optimization procedure that enforces that a policy cannot change too much in a single update and hence limits the damage that can be caused by noisy gradient estimates. Since handcrafting an accurate reward function is often difficult, they used Generative Adversarial Imitation Learning (GAIL). GAIL trains a policy to perform expert-like behavior by rewarding it for deceiving a classifier trained to discriminate between policy and expert state-action pairs. The GAIL objective function is optimized using RL.

For training, they used Next-Generation Simulation (NGSIM) dataset for US highway 100 and Interstate 80. They use GAIL and Behavioral Cloning (BC) to learn policies for two-dimensional highway driving. They used 3 baseline models: static Gaussian (SG) model, mixture regression (MR), and rule-based controller to govern the lateral and longitudinal motion of the ego vehicle. Their validation metrics were root-weighted square error (RWSE), kullback-Leibler (KL) divergence, and emergent behavior.

The root-weighted square error results show that the feedforward BC model has the best short-horizon performance but then begins to accumulate error for longer time horizons. GAIL produces more stable trajectories and its short term predictions perform well. The KL divergence results show very good tracking for SG in everything but jerk. GAIL GRU performs well on the iTTC, speed, and acceleration metrics. It does poorly with respect to turn-rate and jerk. In comparison with GAIL policies, the BC policies are worse with iTTC. The emergent values show that GAIL policies outperform the BC policies. Off-road duration is perhaps the most striking statistic; only GAIL can stay on the road for extended stretches.

The results demonstrate that GAIL-based models capture many desirable properties of both rule-based and machine learning methods while avoiding common pitfalls. Except for the hand-coded controller, GAIL policies achieve the lowest collision and off-road driving rates, considerably outperforming baseline and similarly structured BC models. GAIL also achieves a lane change rate closer to real human driving than any other method against which it is compared.

Strengths

- The paper is nicely organized. Given the length of the paper, they did a good job of explaining their methods and results.
- I like the fact that they also talked about potential future works.
- They even included their code for other people to use.

Weaknesses

- I would have liked to see a little more description of their network architecture.
- I also wised they done some live demos of the simulation.

Discussions

- Why is Generative Adversarial Network right for this problem domain?
- Can we discuss a little more about behavioral cloning and imitation learning? How are they different and what other sectors are they being used in?
- What is Gated Recurrent Unit? How is it different than a normal Recurrent Unit?