Autonomous R/C Car Behavioral Cloning Optimization

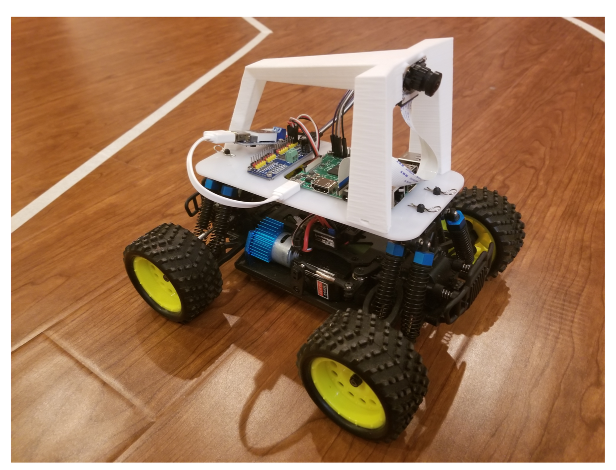
Joon Jung (joonjung@stanford.edu)

Stanford University

Abstract

Behavioral cloning is relatively simple to implement but yields optimal result efficiently. We have used behavioral cloning to train a CNN based autopilot based on an open source platform. The goal of the project was to model and optimize the autopilot in a real world setting, other than a simulated one, trying to gain valuable insights to launch a real world machine learning agent. For the performance optimization, we have employed Data Aggregation[3] to augment the training process.

Agent & Data



Donkey Car [4]

Training

- Input1: 120 x 160 RGB image captured from frontal mono wide angle camera
- Input2: Steering angle from a human driver or an autopilot
- Input3: Motor throttling value form a human driver or an autopilot

Auto Piloting

- Input: 120 x 160 RGB image captured from frontal mono wide angle camera
- Output1: Steering angle
- Output2: Motor throttling value

Behavioral Cloning [2]

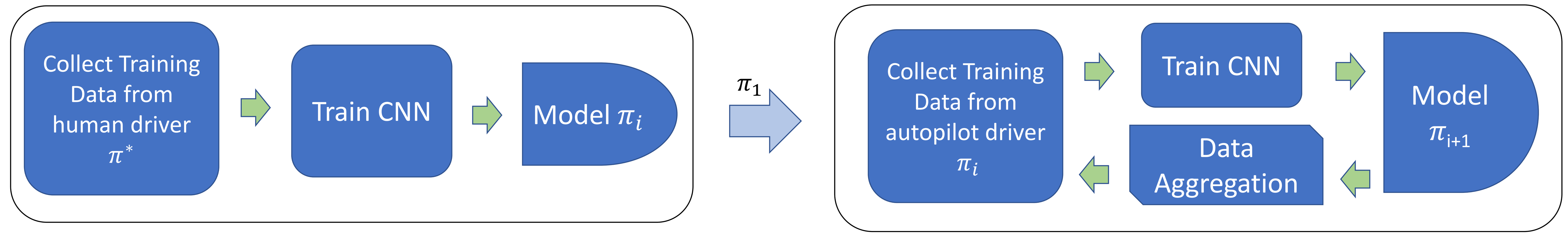
State s: s1 = front direction camera image,
s2 = steering angle, s3 = throttling value
Actions a: a1 = steering angle, a2 = throttling value

Training set: D = {τ:=(**s**,**a**)} from π*
s: sequence of s **a**: sequence of a

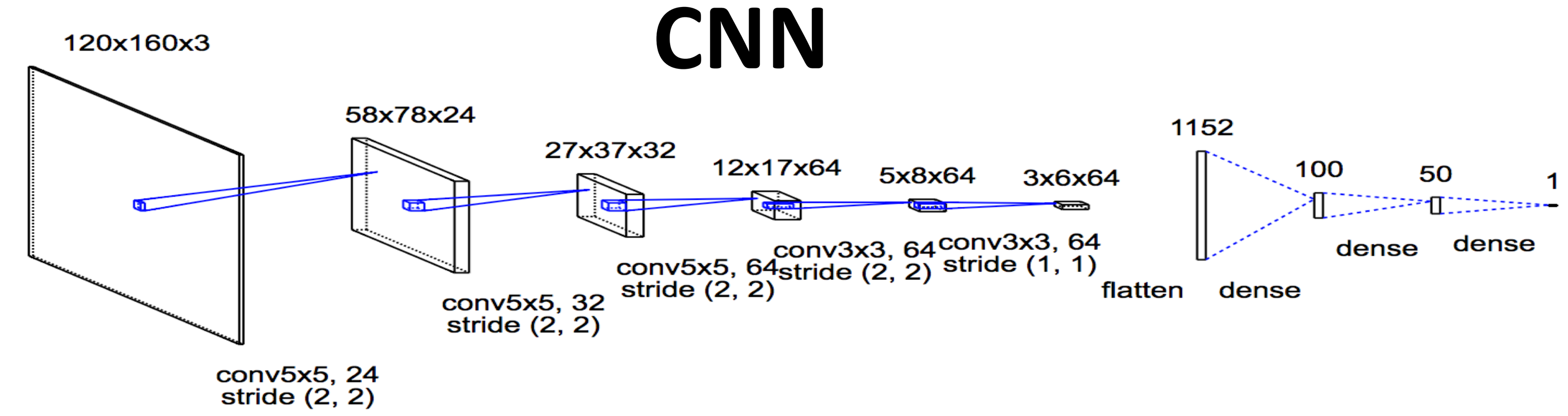
P* = P(s|π*) (distribution of states visited by expert)

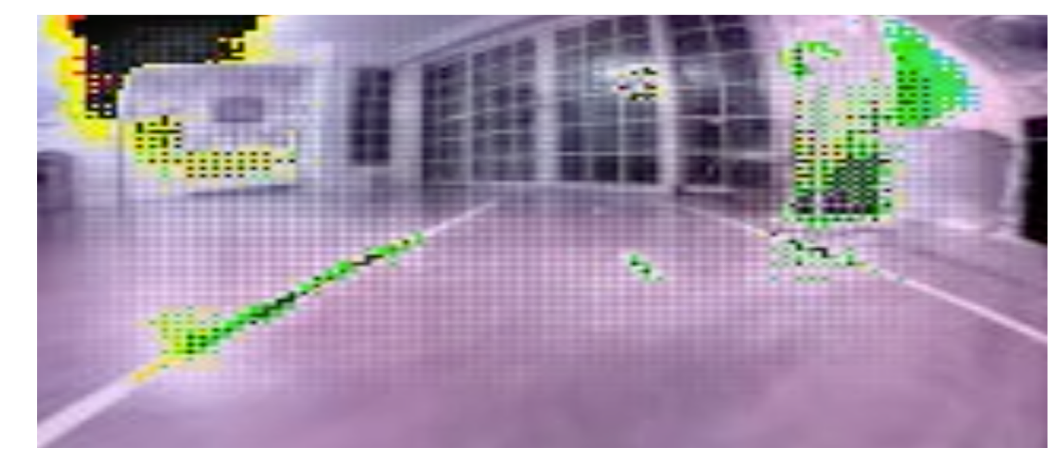
Objective Function: $argmin_\theta E_{(s,a^*)\sim P^*} L(a^*, \pi_\theta(s))$
with MSE

Model

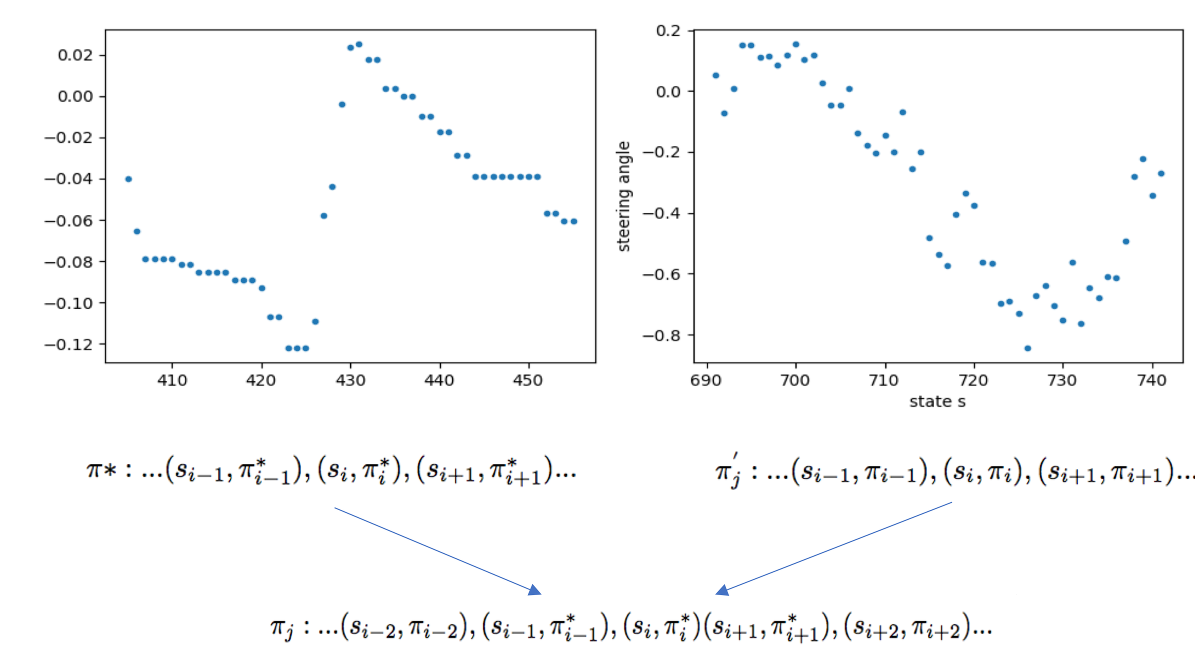


CNN





Expert Control in Iteration j



Experiment Results

Metrics: (# of times being out of track / # of full wraps finished) x 100 with average of 50 full wraps finished for each iteration i

π_i: policy trained after iteration i wo/w Aggregating Datasets

Datasets Aggregated	Human	π_1	π_2	π_3
π_i isolated	10%	26.6%	20%	100%
Aggregating each π_i	n/a	100%	n/a	n/a

* The CNN training and validation losses for each i iteration were all less than 0.05.

Discussion

As shown in the experiment results, unfortunately the tried application of Data Aggregation is far from being optimal. The best performance is achieved only with one iteration of modifying the action control by the expert without aggregating the datasets. The main cause seems to be coming from the fact that it is very hard to modify each π_i, so it wouldn't perturbate the trajectory(τ) space already given. However, the agent reacts quite sensitively with the sequential dependency of each state $(s, a)_i$ with each other.

References

[1] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zhang, X. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*
[2] Yisong Yue, Hoang M. Le: ICML2018: Imitation Learning. https://drive.google.com/file/d/12QdNmMll-bGlSWnm8pmD_TawuRN7xagX/view
[3] Stéphane Ross, Geoffrey J. Gordon, J. Andrew Bagnell: No-Regret Reductions for Imitation Learning and Structured Prediction. CoRR abs/1011.0686 (2010)
[4] Donkey Car: http://www.donkeycar.com/
[5] Keras Salient Object Visualization: https://github.com/ermolenkodev/keras-salient-object-visualisation/tree/fix_tf1.8