Machine Learning Engineer Nanodegree

Capstone Proposal

Christopher Ohara

September 21st, 2018

Domain Background

In recent times, self-driving cars are becoming more feasible. With benefits being more eco-friendly, efficient, and safe, it is easy to see their growing popularity. However, A lot of work still needs to be completed prior to their mass integration into modern society. More specifically, there will be a transistional phase in which cars must first become semi-autonomous via utilization of advanced driver-assistance systems (ADAS) that handle operations such as providing lane departure warnings, automatic land centering, and adaptive cruise control. In the domain of human-technology interation, even more ADAS features are arising in which the vehicle provides take-over requests (TOR) to tranisition the semi-autonomous control back to the driver.

The ability for autonomous vehicles to complete these tasks is in the realm of machine learning. Optimizing a system to maintain a safety distance and safe operation under various driving scenarios requires complex computation and sensor fusion. Computer vision and deep-reinforcement learning are two highly critical fields of machine learning that are applied to solving the complex stochastic environmental conditions (semi)-autonomous vehicles and their drivers find themselves in. The goal of this project is to investigate the possibilities of deep-reinforcement learning as a solution to autonomous navigation.

Problem Statement

As discussed above, there will be a transistional phase prior to the advent of all vehicles having auto-pilot capabilities. More specifically, it is not feasible that on a single day all traditional (non-autonomous) vehicles will be replace, resulting in a time period in which some of the vehicles have the capability while others do not. One approach to resolve this issue is the implementation of Q-learning, a reward-based deep-reinforcement learning technique. Initially, it can be assumed that less than 1% of vehicles will be autonomous, leaving the other 99% of vehicles with human drivers. Human drivers have completely stochastic and non-deterministic (almost arbitrary) behavior when driving, changing lanes or having a variable speed almost on a whim.

Q-learning can approach this problem on two levels. First, the technique can utilize a neural network architecture that is able to be used in the completion of path planning. This allows for the autonomous vehicle to manuveur and avoid other vehicles that are on the path towards the goal. The second approach, which is less feasible due to the arbitrary behavior of humans, is to attempt at “predicting” human behavior and avoiding collisions as a result.

Datasets and Inputs

The machine learning field that the solution is being designed for is under that of reinforcement learning (specifically deep-reinforcement learning). For this project, MIT's DeepTraffic competition data will be used as an input. The goal of DeepTraffic is to alleviate the stress and number of hours vehicles are stuck in traffic, though as an extension, the problem statement above can be addressed. This will lead to improved safety and quality of life for drivers.

The state space can be generally considered as a regression problem, since while the other vehicles have discrete initial feature values (position, velocity, acceleration, and behavior), they operate with non-linear behaviors that are continous in nature. Q-learning utilizes a model-free, off-policy structure within discrete spaces. The algorithms must find a policy with the maximum expect return based on the previous inputs. The inputs are user-developed, though the non-autonomous vehicles positions, accelerations, and velocities will be the source of emphasis. The output will be a mapping policy that best returns the optimal path, with characteristics (outputs) such as move left, move right, slow down, or continue full-speed ahead.

DeepTraffic utilizes neural networks to train a vehicle to drive as quickly as possible through dense highway traffic. Using Q-learning, the vehicle learns how to navigate efficiently, while attempting to maintain the maximum speed limit. Safety protocols are ensured, making it unable to place the autonomous vehicle or other drivers at risk for an accident. The neural network hyperparameters and architecture is completely customizable, leaving all of the design choices up to the student, while enforcing the functional and quality requirements of the autonomous vehicles behavior. With respect to a specific “dataset”, the “manually” driven cars are randomly spawned with various driving behaviors and velocities, making it an unlabeled dataset. The learning input range can also be changed, as the network can look “far into the future,” behind the car, or within specified limits.

In terms of “data points,” there are 20 other vehicles present at any given time with unique values for position, velocity, and acceleration. Note that the data range involves the numbers of cars present (and physically passed), which is a function of time and the number of iterations required to train the network. Therefore, the data range is variable that increases linearly with time, until the trained model is tested against the evaluation model.

https://selfdrivingcars.mit.edu/deeptraffic-about/

Solution Statement

The solution is to properly design and code the neural network to learn the environment and relative behavior of other vehicles. A visualization is also present, in order to directly observe the performance of the network in real-time. The learning input range should be made with feasibility in mind, as the stochastic nature of the other vehicles predicted too far in the future will cause poor training in the network. Superfluous information also needs to be mitigated, in order to ensure the maximum performance of the autonmous vehicle.

Benchmark Model

For a benchmark model, MIT has provided a default network provided, which is linked below. Within this network, changing the default hyperparameters without changing the network will only return an average speed of approximately 51.5 mph, or can even cause the vehicle to stall in traffic. The initial model uses one fully connected layer, using a rectifued linear unit (ReLU) with one neuron and a regression layer. The initial batch size is 64 and the network is trained for 10k iterations. The learning rate, momentum, and L2 decay rate are the default values used in most neural networks.

The network size is a function of the number of inputs, number of actions, and the temporal window. Varying any of these hyperparameters will lead to highly variable results, so proper strategies will need to utilized in order to gain an improved performance and meaningful results (i.e. randomly changing parameters will not lead to meaningful results).

The results of the naive benchmark model are 51.5 mph, so this is the goal to beat (performance). In terms of typical machine learning-based metrics, the number of iterations over time is plotted against the reward rate, which is around 0.5 for the naive benchmark model. This value is also required to be improved in order to gain a higher performance.

https://selfdrivingcars.mit.edu/deeptraffic/

Evaluation Metrics

For evaluation, two metrics will be used. The first metric planned is to meet the performance requirements that MIT has set in order to receive credit for the assignment. This values is to have an average speed exceeding 65.0 mph. A second performance metric is to compare the results with the global competition winners. The three winners acheived average speeds of 74.48, 74.04 and 73.59 mph. Interestingly enough, all three winners won a scholarship to Udacity's Self-Driving Car Nanodegree Term 1.

Therefore, as a bare minimum, 65.0 mph should be exceeded as a key performance indicator.

In terms of a specific machine learning-based metric, typical deep-reinforcement learning metrics will be used. Specifically, the reward rate will be shown with respect to iterations and time, and a higher reward rate is correlated with better performance (accuracy).
https://pdfs.semanticscholar.org/f3b5/689d80cf849e94bb41e9e7b05f2f5523

Note that the same inputs will be used from both the naive benchmark model and the evaluation (improved) model.

https://selfdrivingcars.mit.edu/deeptraffic-leaderboard-1-0/

Project Design

I intend to add more layers to the neural network and change the hyperparameters. The number of layers will be iteratively increased, within the feasible performance trade-offs that are wellknown in machine learning (i.e. adding 100 layers is more detrimental than helpful). For the benchmark model, the learning iterations is quite low, and the momentum value is not used at all. The learning rate might need not need to be changed, but the experience size and learning threshold do. Furthermore, different layer types can be used instead of only ReLUs. L1 decay is also not implemented in the network, so it will be investigated and potentially implemented.

The lectures and information provided by Udacity's Machine Learning Engineer Nanodegree will be referenced and utilized in making design choices, as well as an intended discussion of why each choice was made. The autonomous vehicles acceleration can be changed as well, but this might not be necessary or desired (the car may learn that going slow leads to more rewards, i.e. less blocked paths).

Research will be conducted to gain insight into what similar researches have proposed as solutions (models) with neural networks in order to minimize the amount of time spent in the design phase. Q-learning is a field relatively in its infancy, so there are a finite amount of sources with potentially impressive improvements to be made.

---

https://selfdrivingcars.mit.edu/?wppb_cpm_redirect=yes

https://cs.stanford.edu/people/karpathy/convnetjs/

https://cs.stanford.edu/people/karpathy/convnetjs/demo/rldemo.html