Using a Deep Q-Network To Learn Autonomous Driving Policies

Eugene Saghi

University of Colorado, Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO 80918
esaghi@uccs.edu

Abstract

The development of an Autonomous Driving System (ADS) is an area of research that has received much attention in the past 30 years due to the many benefits that the development of such a system could bring. There are many approaches to creating an ADS which have been explored but the use of Deep Reinforcement Learning (DRL) is a common thread in many of these approaches. Other areas of active research include the use use of imitation learning to develop an ADS and the use of visual attention to enhance the interpretability of decisions made by ADSs. This project will explore the use of a variation of a Deep Q-Network (DQN) to learn optimal driving policies for an agent in the open-source CARLA urban driving simulator. The performance of the agent will be evaluated by the speed with which it can reach a goal destination, the number of times it leaves the roadway and the accumulated impact damage it suffers en route to this destination.

Introduction

Autonomous driving is a field of research which has garnered considerable interest over the past 30 years thanks to the many potential benefits (including financial, environmental, and safety benefits) that the development of an ADS could bring. Any successful ADS will need to process a multitude of complex input data, extract meaningful features from this data, and then use these features to make decisions and perform actions. While the number of actions an ADS has available to it are relatively limited, the number of states such a system could find itself in are nearly infinite. This challenge of learning an optimal control policy over a nearly infinite set of possible states makes DRL an ideal paradigm for the development of an ADS.

Background

Automobiles are ubiquitous in everyday life and an integral part of modern cities, with the U.S. Department of Transportation Federal Highway Administration reporting a total of 283,400,986 vehicles registered in the U.S. in 2022[1]. The drawbacks of having so many vehicles so closely integrated in the daily lives are Americans are manifold and include traffic congestion, air pollution, fossil fuel consumption, and of course traffic fatalities. A technical report published by the National Highway Traffic Safety Administration reported that 94% of road accidents are caused by human errors (Yurtsever et al. 2020), and it's probable that in many instances traffic congestion (and the resulting increase in air pollution and fossil fuel consumption it causes) can be directly attributed to such accidents. It is in light of these facts that a great deal of research has gone into the development of ADSs. This research began as early as the 1980s (Yurtsever et al. 2020) and has only intensified as the capabilities of both hardware platforms and software systems (particularly deep learning techniques) have matured.

Broadly speaking, current approaches to ADSs can be broken into two categories: standalone, ego-only approaches and connected multi-agent approaches (Yurtsever et al. 2020). Most research today is being conducted on ego-only approaches for a number of practical reasons, and while multi-agent approaches are still being explored there are currently no functional ADSs that use this modality. ADSs can be categorized along a different axis into either modular systems or end-to-end systems (Yurtsever et al. 2020). In modular systems separate modules handle the tasks of object detection and tracking, behavior prediction, planning/decision making, and vehicle control. In end-to-end systems Deep Learning (DL) is used to directly process input sensor data and produce control signals for the vehicle. DRL has shown great promise for end-to-end ADSs and is even appropriate for some tasks in modular ADSs (such as controller optimization, path planning and trajectory optimization, and dynamic path planning) (Kiran et al. 2021). This project will explore the use of a variation of a DQN (a type of DRL neural network) to create an end-to-end ego-only ADS in a simulated environment.

Related Work

Because of the promise of an ADS truly independent from human intervention there is a great deal of ongoing research into various applications of DL in the context of ADSs. The following sections provide greater detail about a number of recent research projects that use DL to construct or improve ADSs.

Imitation Learning

Imitation learning is a type of DL which in the context of ADSs "maps sensor inputs to vehicle control commands

[1] https://www.fhwa.dot.gov/policyinformation/statistics/2022/

via supervised training on large amounts of human driving data" (Liang et al. 2018). In other words, unlike DRL imitation learning is a form of supervised learning where the optimal policy is learned by observing the actions taken by an "expert" (in this case a human driver) given a particular state. When compared to DRL this approach has both advantages and disadvantages. The primary advantage is the ease of obtaining training data at scale (Codevilla et al. 2018). However, a major drawback to this approach is the fact that long-term driving goals (such as reaching a particular destination) cannot be learned from sensory input alone but instead require a degree of knowledge which is not present in labelled training data. Furthermore it has been found that imitation learning systems generalize poorly to unseen scenarios (Liang et al. 2018).

To overcome these limitations a group of researchers in 2018 proposed using conditional imitation learning for autonomous driving, in which an imitation learning network was given the intentions of the driver during training in addition to the sensory inputs and control commands (Codevilla et al. 2018). During testing these driver intentions were supplemented with similar intentions specific to the testing environment. While this approach showed promise in a virtual environment, the researchers concluded that there remains "significant room for progress" in the development of such an approach (Codevilla et al. 2018). A separate group of researchers in 2018 attempted to combine imitation learning with DRL to overcome the shortcomings of both in an approach they called Controllable Imitative Reinforcement Learning (CIRL). For their experiments an imitation learning network was trained first and then transfer learning was applied to transfer the weights learned during imitation learning to a DRL actor network (Liang et al. 2018). A Deep Deterministic Policy Gradient (DDPG) algorithm was then used in conjunction with a critic network to optimize the policy learned by the actor network. This CIRL network achieved state-of-the-art driving performance in a simulated environment (Liang et al. 2018).

Interpretable Learning

Besides the difficulty of developing an ADS which requires no human intervention, the wide adoption of such a system will likely be hindered by public trust in the decisions made by the system. This is due to the fact that neural networks are cryptic and not easily interpretable. To bridge this trust gap and address the interpretability of ADSs a pair of researchers in 2017 proposed the use of an encoder-decoder network to produce visual attention maps which are then further analyzed to determine the saliency of the input regions in the attention maps (Kim and Canny 2017). The encoder portion of this architecture is simply a CNN for feature extraction, while the decoder portion is split into a coarse-grained decoder and a fine-grained decoder. The coarse-grained decoder is a deterministic soft attention mechanism to produce attention heat maps, and the fine-grained decoder applies a clustering algorithm to these heat maps. Clusters are then iteratively masked and the performance of the network is evaluated to determine the visual saliency of each cluster. The researchers found that the incorporation of visual attention did not degrade control accuracy while it did provide meaningful information about what visual data the network used to make decisions (Kim and Canny 2017).

Methodology

This project will use a variation of a DQN to learn the optimal policy for an ADS agent to navigate from a random starting location to a random destination within a simulated environment while obeying standard traffic rules. The following sections provide more detail about the proposed model architecture, the simulation environment, and the sets of states, actions, and rewards available to the agent.

Network Design

The proposed model architecture for this project is based on a design proposed by a group of researches in their 2017 paper "Deep Reinforcement Learning Framework for Autonomous Driving", which in turn is based on the DQN architecture proposed by a group of researchers in 2015. The development of the DQN was motivated by a desire to leverage new developments in deep neural networks to build an agent capable of end-to-end reinforcement learning using only high-dimensional sensory inputs (Mnih et al. 2015). The researchers tested the DQN on 49 different Atari 2600 games using 4 frames of 84x84 images as input (which were obtained by preprocessing the 210x160 60Hz video output of the games), and found that it outperformed state-of-the-art reinforcement learning models and even achieved comparable results to professional human players across a majority of the games the agent was trained on (Mnih et al. 2015). The DQN consists of 3 convolutional layers, a hidden fully-connected layer, and output layer which uses a softmax activation function to select the appropriate action from the set of all actions available to the agent.

To adapt this DQN architecture for end-to-end autonomous driving Sallab et al. proposed several key changes. The first of these changes is the integration of an actor-critic learning framework in which the policy function (actor) and value function (critic) are learned by separate networks which work together to learn the optimal policy. This change was motivated by the observation that the original DQN architecture is suitable for environments with discrete action spaces, but an agent in an ADS needs to operate in a continuous action space (which the actor-critic learning framework is compatible with) (Sallab et al. 2017). For this project the actor-critic networks used will be based on the Soft Actor-Critic (SAC) algorithm proposed in a 2018 paper in which the researchers found that the SAC algorithm outperformed state-of-the-art model-free DRL methods (Haarnoja et al. 2018). The next change from the original DQN architecture is the inclusion of an attention mechanism to filter the features learned by the CNN layer. This change was motivated by the observation that not all of the features learned by the CNN layer will contribute equally to the final optimization objective (Sallab et al. 2017). The final substantional modification to the DQN is the addition of a LSTM layer between the filtered CNN outputs and the Q-network. This recurrent layer was added because the researchers observed that an

ADS agent will not be learning in an environment that truly obeys the Markov assumption that underlies Markov Decision Processes (MDPs). Instead, the occasional occlusion of objects/vehicles by other vehicles on a roadway mean that an ADS will be operating in an environment that is a partially observable MDP (POMDP). The addition of a LSTM layer adapts the DQN to work for POMDPs (Sallab et al. 2017).

Environment

The environment in which the ADS agent will operate for this project is the Car Learning to Act (CARLA) simulator. CARLA is an open-source simulator that was developed with the goal of supporting the training, prototyping, and validation of ADSs (Dosovitskiy et al. 2017). It is highly sophisticated and contains a suite of urban maps with assets including buildings, vegetation, infrastructure, street signs, vehicles, pedestrians, and even a variety of weather conditions. Environmental settings (such as weather conditions, number of non-agent vehicles, number of pedestrians, and vehicle spawn points) are highly customizable. CARLA also simulates a wide variety of sensor inputs for the agent vehicle which are based on sensors used in real ADSs. The following sections define the state space, action space, and rewards that will be used to train the agent for the project.

State Space As is the case with ADSs that exist outside the virtual space, the state space for the ADS agent in this project will be comprised of an amalgamation of sensor inputs that describe the relationship between the agent and the environment. The sensors simulated by the CARLA platform that will be used as inputs to the agent include camera inputs, LIDAR, speed, acceleration vector, accumulated impact from collisions, and a pseudo-sensor which gives information about the position and orientation of the agent in the environment (analogous to a GPS in a non-virtual ADS) (Dosovitskiy et al. 2017). Because the vehicle starting point ("spawn point") can be customized the agent will start in a random location on the map and will navigate until it reaches another predefined random location on the map (the goal state).

Action Space In contrast to the state space defined above, the action space for this project will be quite simple: a three-dimensional vector that represents the steering, throttle, and brake. These will be the only three actions available to the agent at any time, although it should be noted that the actions are independent (can be performed simultaneously) and exist along a continuum (as opposed to being discretized).

Rewards The rewards available to the agent will likely be subject to change and experimentation as training progresses. Initially the agent will be given a large positive reward for reaching the goal state. Negative rewards will be given for colliding with objects and for leaving the road. A small negative reward will also be given for changing lanes, with the goal of teaching the agent to stay within a lane and minimize the number of lane changes it makes. Q-values will be calculated using a discount factor (γ) to encourage the agent to reach the goal state as quickly as possible. Early iterations of training will not reward obeying traffic rules as this is a much more complex behavior than navigating from one point to another without leaving the road or colliding with anything. If time permits the reward space will be updated in later iterations of training to teach the agent to obey more complex traffic rules such as speed limits, street signs, and traffic lights.

Evaluation

The CARLA simulator includes a smaller test urban environment which will be used to evaluate the agent once training is complete. The agent will be evaluated on the time it takes to reach the goal state, the number of times it leaves the boundaries of the roadway, and the accumulated impact from collisions. The agent will be evaluated periodically throughout training to record these metrics and thereby track the learning of the agent.

Timeline

Table 1 below contains the proposed timeline for the project leading up to the March 18th midterm.

Date	Objective
2/19 - 2/25	Configuration of the simulation environment
2/26 - 3/3	Configuration of the simulation environment
3/4 - 3/10	Construction of model and integration with simulation
3/11 - 3/17	Initial model training and evaluation

Table 1: Timeline

Conclusion

For this project a variation of a DQN will be constructed and trained in the CARLA simulator to learn optimal driving policies. The goals of the policy and the environment in which learning takes place will be simplified at the beginning of training but scaled as the performance of the agent improves. The performance of the agent will be evaluated by the time it takes to reach a given destination, the number of times it leaves the roadway, and the accumulated impact from collisions the agent suffers.

References

Codevilla, F.; Müller, M.; López, A.; Koltun, V.; and Dosovitskiy, A. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, 4693–4700. IEEE.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Dy, J.; and

Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1861–1870. PMLR.

Kim, J.; and Canny, J. 2017. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, 2942–2950.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 4909–4926.

Liang, X.; Wang, T.; Yang, L.; and Xing, E. 2018. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European conference on computer vision (ECCV)*, 584–599.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*.

Yurtsever, E.; Lambert, J.; Carballo, A.; and Takeda, K. 2020. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8: 58443–58469.