Summary of Reuse Optimization in DRL for Autonomous-Driving in Pygame

Austin Hartley

This paper introduces Deep Reinforcement Learning in autonomous driving and the ongoing problem of getting the results to converge efficiently. To help improve this, they propose a training method to decrease learning time and improve stability by reusing past trained networks to find better strategies for Deep Q-Learning. Essentially, the car begins its state as the algorithm rewards it accordingly based on its performance, and this is how it slowly starts to learn. The reuse method involves deciding if the current state has enough information to make an informed decision and, if not, it uses the previous state that did. They set up a driving environment in Pygame with multiple components (roads, other vehicles, boundaries, etc.), had a car traversing the terrain with seven sensors attached to it, and compared the results to the method without the reuse optimization.

The ideas and concepts were clearly articulated so that someone unfamiliar with machine learning would be able to understand the big idea. Providing the code and figures of the algorithms further enhances the paper and allows others to implement the project themselves and actually understand the algorithms. Since a lot of data science and machine learning fields are accustomed to using python, making real-time simulations may not be an easy task for those in that field. Showcasing Pygame, which takes advantage of people's python skills, as a cheap resource to get a fast simulation prototype up and running is a great way to take advantage of existing libraries and inspire others to do the same. The authors were able to optimize the previous idea of DQN with Experience relay and prove that the reuse method can improve results showing graphs with concrete results.

The weaknesses of this paper would be that the simulation environment was extremely simple compared to a real-life scenario and was done in only two dimensions. Perhaps the advantages of this approach start to decrease when the data of a more realistic scenario in three dimensions comes into play and the extra variations make previous models less useful in achieving higher convergence. The experiment was also a low-scale simulation, without needing much computation power or other optimizations on that front, which is usually the scale wanted for others to truly get behind an idea and want to improve upon it. While the results were sound and conclusive, having more statistical data showing the optimization benefits would have been better and made their point more evident.

Overall, this paper was not only useful in the specific domain of autonomous driving but offered a generalized optimization for Deep Reinforcement Learning in all aspects. Showcasing Pygame as a tool that can be used as a cheap and simple interface for fast prototyping is also beneficial as more studies could be done without the overhead of cost and time. Taking an existing idea and improving upon it while giving the necessary information for others to fully understand the optimization is exactly how progress is ensured.