Human-level control through deep reinforcement learning

Thierry Deruyttere
r0660485

Armin Halilovic
r0679689

1. Introduction

For the course “Capita selecta: Artificial Intelligence” we received the subject of **deep reinforcement learning**. In this document we will explain everything we have done to get to know this subject. We will first start with our original plan we had for the demo, what we did with it and what eventually went wrong. We will also mention the things we learned from this experience. Then, we will explain to which other simulator we switched and how things went from there and which simulations we ran. We will finish this document with the things we have learned whilst researching this subject.

2. First simulator

For our first simulator we maybe saw things a bit too big. We wanted to showcase a really cool demo. That’s why we decided to do deep reinforcement learning with self driving cars since they are all over the news. We quickly found some cool simulations online namely *DeepTraffic*[2] with a corresponding MIT course available on youtube[**?**] and a simulator written in Unity for a Udacity course[1] that we could use for our demo. We decided to go with the latter for our demo but since the former was a bit easier to start with, we started with DeepTraffic to get to know our subject.

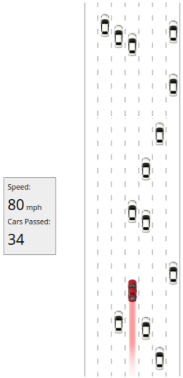

Figure 1. The DeepTraffic simulator[2]

2.1. DeepTraffic

The DeepTraffic simulator (which can be seen in figure 1) is a simulator written in javascript that works in a browser. The goal of the simulator is to make an agent that can swiftly maneuver between traffic without making collisions. The agent’s reward is based on its speed. A high speed means a high reward and a low speed means a low reward. The nice thing about DeepTraffic is that it also has a MIT course[3] that explains a whole lot of things about self driving cars. To prepare ourselves for the task we wanted to accomplish, we watched the full play list (only five video’s in total) of this course available on youtube.

Once we finished the playlist we started with an implementation for the simulator.

2.2. Udacity course simulator

We decided to go with the Udacity self driving car simulator written in Unity since this was the coolest of them both. We had to do some changes to this simulator since it was written to do behavioral cloning but our research was about deep reinforcement learning. Once we made the changes we could finally start to implement a deep reinforcement learning algorithm to write the car by itself.

3. Second simulator

3.1. Flash games

3.2. Atari games

3.3. Reinforcement learning

4. Conclusion

The conclusion goes here.

References

[1] https://eu.udacity.com/course/self-driving-car-engineer-nanodegree–nd013

[2] Deeptraffic.js, https://selfdrivingcars.mit.edu/deeptraffic/

[3] L. Fridman, MIT 6.S094: *Deep Learning*, https://selfdrivingcars.mit.edu/