Cardiff University
Computer Science and Informatics

CM3203 - Final Year Project

Reinforcement Learning for Autonomous Driving

Initial Plan

Author:
Benjamin Loki Hughes

Supervisor:
Dr Kirill Sidorov

Contents

1 Project Description **2**

2 Project Aims and Objectives **3**

2.1 Aim 3

2.2 Objectives 3

2.3 Milestones 4

3 Work Plan **4**

1 Project Description

As Driverless cars become an increasing reality, with advancements in Machine Learning, sensor accuracy and computational power now being able to support the 'self driving car' vision, the Formula Student Competition has branched into this field. As such, Cardiff Racing team now operates a FS-AI subgroup to tackle the problem. As the setting for formula racing is much simpler, and more predictable than pedestrian roads, the problem is a simplified sub-case that a usual driverless car might have to tackle. For example, the car will not have to abide by road safety laws, or accommodate for pedestrians, traffic lights etc. The task then is broken into the following pipeline, which forms the main loop of the system:

1. Firstly the system must gather data in order to realise its position on an unknown track, given by cones to denote area. This means acquiring sensor information and making sense of this data. This will come from both LiDAR data, and Visual data from cameras. The visual data will have to be parsed though a recognition system to recognise the cones denoting the track area, as well as a system for extracting distances of the cones to the vehicle. LiDAR data must also be filtered to recognise cones and measure the distance natively.

2. Then, as the track will be previously unknown, we cannot represent the track prior to the car needing to drive in it, therefore the car must learn the track whilst driving through it, simultaneously localising itself to the track as it drives. Thus, the data must be passed through a SLAM (Simultaneous Localisation and Mapping) algorithm to produce a model of the environment and to understand the car's position within the environment based on its observations and odometry readings.

3. Given that the car understands the track, and where it is in the track, it must then decide what it is to do to drive around the track. The car will have control over its steering and acceleration, which it must optimise, as a racing driver would, to complete a fast, and complete, track.

4. Finally, there must be some way of actuating the decisions made by the AI system so that the car does in fact follow the motions it is meant too.

Thus far, the majority of the work has taken place regarding the first two items, that is to locate cones and build a map of cones as the car observes them. This project will tackle the problem of formulating decisions based on this data. Specifically, following from DeepMind's famous success [1], this project will see if reinforcement learning can be used to train an agent to formulate actions based on this data, as has been considered for full commercial driverless vehicles [3] without the need for training data. This will be trained in a simulated, simple environment where cones are assumed to be detected. After it is trained, it will be tested in a 10th scale real environment, which has been prepared outside the scope of this project, where its decisions will be actuated by a radio controlled car, which, again, has been prepared outside of this project.

2 Project Aims and Objectives

The time constraint of this project limits the reach of this system, where I might also have ensured optimality for data acquisition and SLAM for mapping, I shall have to use the work that is ongoing and being implemented by the team, with this in mind, the system will only work as well as as this works in a real setup. That being said, the simulated environment will assume this works fine, though noise will have to be added to account for real world application regardless of the system accuracy. Time will also force me to use pre-existing machine learning libraries, such as *Tensorflow*, to complete this project, as well as using pre-existing libraries for quickly building an adequate simulated environment.

2.1 Aim

To train a Reinforcement Learning algorithm to drive a car around a track given by spaced cones.

2.2 Objectives

1. Research Reinforcement Learning Algorithms to decide on the most appropriate algorithm; *Q Learning, Deep Q Learning, Policy Gradient Learning* [2].

2. Produce a 2d environment to train a model in using *pygame* that can produce random tracks.

3. Produce a physically accurate model of the car; accounting for inertia in steering.
4. Decide on what to pass to the algorithm; Raw sensor data, Cone Distances, SLAM data, to determine at what point the model will take control.
5. Design an appropriate *reward function*
6. Produce Learning model using *Tensorflow* in *Python*.
7. Train the model in the simulated environment, potentially using the university's super computer if necessary.
8. Evaluate the model for accuracy given the results in the simulated environment, and make amendments
9. Evaluate the model on the 10th scale model.

2.3 Milestones

1. Having produced a full test environment that can appropriately create random tracks.
2. Having produced a physically accurate car.
3. Having fully designed the algorithm; chosen an appropriate algorithm, the data it will take in and designed a cost function.
4. Having produced the learning agent.
5. Having seen the agent complete a track of the simulated environment.
6. Having seen the agent complete a physical lap.

3 Work Plan

Outlined below is a weekly plan for my project. I understand that this is subject to change as the project develops, which is why the weekly activities don't bare too much detail. This is, roughly, a waterfall model, though during the design process for my model, the environment must be completed, so this doesn't strictly follow the model. The dates set are also liberal to allow for backtracking and re-assessment when necessary instead of strictly at the end of the project, though time is allocated for more editing at the end of the project, or for possible spilling, should the project go beyond what I have allocated, to ensure it is completed by the final, strict, deadline.

Week 1: Discuss the project with supervisor, create the initial report and begin researching Reinforcement Learning, which will continue through the first few weeks alongside any coding work.

Weeks 2 and 3: Create a 2d virtual environment using *pygame*. This environment must be such that it can create random tracks, that adhere to the FS-AI rules, using objects to represent cones, that have a start and finish, and can be ran quickly, as it will need to be ran many times while that agent is training and can support an AI agent to drive around it.

Weeks 4 and 5: Create an AI-ready agent that can drive around the track, with physically accurate steering system and acceleration. The Agent's *vision* will be represented by a circle (scaled to represent a range of 12m) that will pick up on cones and feed their accurate coordinates to the agent. At this point I will implement an algorithm for the agent to centre itself between cones with constant velocity, for later comparison.

Weeks 6 and 7: Develop a reward function, and use it in the implementation of the chosen algorithm using *Tensorflow*. Then train this model in the simulated environment. This may require the school's supercomputer, for which I will need permission.

Weeks 8 and 9: Assess the model and make any changes needed to improve accuracy, this will include re-assessing the reward function and changing hyper-parameters to improve the model's capabilities. This time will also include changing what data the model takes in (e.g SLAM before decision making vs putting cone coordinates into the system without SLAM).

Week 10: Porting the algorithm onto the physical 10th scale car (prepared outside of this project) for true performance measuring.

Weeks 11 and 12: Collating all the data and writing the final documentation

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529 EP –, 02 2015.

[2] R. Sutton and Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[3] Sen Wang, Daoyuan Jia, and Xinshuo Weng. Deep reinforcement learning for autonomous driving. *CoRR*, abs/1811.11329, 2018.