NWU®

FACULTY OF ENGINEERING

CREATING ROBUST GENERALISATION FOR A SINGLE-CAMERA SELF-DRIVING CAR

(Or A Machine Learning Model for a Single-Sensor Self-Driving Car, A Proposal)

Submitted to: Prof. Kenneth Uren

Date: 03/02/2020

Student Name: Chara Grant

Student Number: 29458447

Table of Contents

1. Background

Since the early 2010's, there has been a resurgence in interest and research into Machine Learning and its various applications [1]. This tool has rapidly been applied in sectors of finance, analytics, and robotics – to great success and enthusiasm [1].

The dawn of Machine Learning as we know it today is considered to be the year 1952, when the term "Machine Learning" was coined by Arthur Samuel to describe a Checkers-playing algorithm that could remember all the moves it had made and make moves based on the likelihood that it would be a winning move [2]. A few years later, in 1957, a physical perceptron machine - the Mark 1 Perceptron on the IBM 704 – was designed for the purposes of image recognition. While this was promising, it failed to recognize patterns and features now staples in computer vision [2].

In the 1960's, the use of multi-layered perceptrons began to catch on – and thus the neural network as we know it today was born. With this came feed-forward networks and backpropagation, which ultimately resulted in the emergence of ANNs (Artificial Neural Networks) with hidden layers – which began to open the door to finding new and more complex patterns [2].

Fast forwarding to 2012, Google had developed a model able to find and recognize cat videos, and Facebook had mastered facial recognition in 2014 to human-level accuracy [2].

Deep Neural Networks have since been employed in the development of self-driving vehicles [3]. In 2009, Google began their self-driving car project, known now as Waymo [4]. This vehicle makes use of several DNNs, and at least eight sensors [5]. Similarly, Tesla has already released a semi-autonomous autopilot mode [6]. These systems made use of several specialized DNNs and CNNs, and were mostly the domain of companies like Alphabet with the funds and personnel to pour into this development [6] [3].

Circa 2017, the Donkeycar and Carputer emerged as part of the DIYRobocars initiative. This aimed to be a ground-up project making driverless cars and experimentation accessible to everyone. The Donkeycar consists of a single input camera and a simple Convolutional Neural Network that learns to drive from data collected from a human driver. The cars that this system is applied to are typically cheap and small remote-controlled cars. As of 2020, Donkeycar has hit version 3.0, and there are races and meetup locations in several countries across the globe [7].

2. Problem Definition

The model used by the Donkey Car is an open-source, bare bones convolutional neural network that allows a scale vehicle to drive itself based on training data gathered by a human remote driver. This driving/learning style is known as behavioural cloning. However, very little generalisation is possible with the vanilla Donkey Car model.

Experience working with the model has shown that changes in location, lighting and positions of objects completely derail the self-driving model. Data is also time-consuming to collect, and is in a standard, unmodified image form from a single front-facing camera – several thousands of images are required to properly train a model.

3. Project Scope

The goal of this project is to examine the pilot portion of the Donkey Car and re-design it in such a manner that it is at least able to generalise over many different tracks (with the same lane markings). This generalisation should be robust at different times of day, in different locations, and in different lighting conditions. The pilot should at least consist of an independently implemented Convolutional Neural Network.

This re-design does not touch any of the other workings of the Donkey Car System – such as the motor driving system, camera drivers, or web control system. The focus on the project is investigating the design of a good Convolutional Neural Network and methods of augmenting/feature engineering data in order to ensure good generalisation of the pilot model.

4. Objectives

4.1. Develop Definition of Generalisation

What is required in terms of generalisation? Should the vehicle be able to navigate on both a marked track and a footpath? Does generalisation mean being able to correct if swerved off the track? To what extent must the model be able to perform on obscured/new data?

4.2. Develop Requirements for Pilot

Which network architecture must be used? What are the functionalities that we will require of the pilot? What can be left open to further development?

4.3. Design One or Two Pilot Models

There are several neural network architectures and types from which to choose. Design a few basic architectures that can be tested and added onto throughout the project.

4.4. Iteratively Build, Simulate and Test Several Pilot Architectures

LTSM, RNNs, CNNs, 3DCNNs are some of the many architectures that can be built and tested. These networks need to be built, simulated and evaluated in order to come up with a final design.

4.5. Create end-to end Pilot based on Simulation and Test Results

Based on test results, choose a model or create a model that will be developed into an end-to-end pilot. This pilot must be demonstrated to generalise according to the generalisation requirements.

5. Methodology

5.1. Develop Definition of Generalisation

This will be accomplished by first considering the question of what behaviour the Donkey Car currently exhibits, and what behaviour is desired. Once this question has been answered, it is helpful to move on to determining to what extent generalisation is possible with a single camera input – and if additional hardware may be added without compromising the accessibility and simplicity of the Donkey Car and/or stepping over the bounds of the project scope – such as adding cameras/sensors.

5.2. Develop Requirements for Pilot

The main goal of the pilot is to allow the vehicle to follow a marked track. Thus, some of the requirements would include but would not be limited to lane detection and image recognition.

5.3. Design One or Two Pilot Models

A Jupyter notebook will be set up for each model design, stating how the data is preprocessed, the optimization functions used and why, as well as several other details regarding the architecture and working of the neural network. The Jupyter notebook will serve as a self-reporting tool describing the work being done, with live code embedded.

5.4. Iteratively Build, Simulate and Test Several Pilot Architectures

Once these models have been constructed in their relative Jupyter notebook, they will be iteratively tested and improved on. This will be done by integrating said models into the donkey system and running them in the Donkey Gym simulation system. Once the results in the simulator for each architecture are satisfactory, these models and their data preprocessing will be ported to and tested on the physical Donkey Car.

5.5. Create end-to end Pilot based on Simulation and Test Results

Based on the simulation and test results, an end-to-end final pilot design will be completed. This design will follow after the pilot model that performed the best in the generalised conditions when tested on the physical Donkey Car system.

6. Chapter Layout

The typical chapter layout of the final report will be as follows:

6.1. Introduction

6.2. Literature Survey

6.3. Data Pre-Processing Study

6.4. Neural Network Pilot Design

6.5. Testing and Evaluation

6.6. Final Design Decisions and Motivation

6.7. Conclusion and Recommendations

7. Timeline

The timeline of the project will closely follow the recommended 4th year project schedule. The deliverables, as highlighted in the objectives, will be slated to be completed on the following dates:

- Development of Generalisation Definition (21 Feb/Ongoing)
- Literature Survey and Introduction (18 March)
- Develop Requirements for Pilot (23 March)
- Specification Document (1 April)
- Have at least 2 Pilot Jupyter Notebook Designs (10 April)
- Oral Presentation (Design Concept) – (15 April)

- Have at least 1 Simulation (17 April)
- Have a collection of demonstrable model designs and simulations (15 May)
- Design Report (20 May)
- Have Documentable Tests of Model Architectures (12 June)
- Core Functionality Demo (1 July 2020)
- End-To-End Pilot – Part 1 (14 July 2020)
- 50% From Complete Demo (15 July 2020)
- End-To-End Pilot – Final (7 August 2020)
- Extras/Refining (15 September 2020)

8. Bibliography

[1] T. Nield, "Is Deep Learning Already Hitting its Limitations?," Towards Data Science, 5 January 2019. [Online]. Available: https://towardsdatascience.com/is-deep-learning-already-hitting-its-limitations-c81826082ac3. [Accessed 2 February 2020].

[2] K. D. Foote, "A Brief History of Machine Learning," Dataversity, 26 March 2019. [Online]. Available: https://www.dataversity.net/a-brief-history-of-machine-learning/#. [Accessed 2 February 2020].

[3] K. Burke, "How Do Self-Driving Cars Make Decisions?," Nvidia, 7 May 2019. [Online]. Available: https://blogs.nvidia.com/blog/2019/05/07/self-driving-cars-make-decisions/. [Accessed 2 February 2020].

[4] Waymo, "Waymo," Waymo, [Online]. Available: https://waymo.com/. [Accessed 2 February 2020].

[5] A. Woollaston, "How do Google's driverless cars work?," alphr, 4 April 2016. [Online]. Available: https://www.alphr.com/cars/7038/how-do-googles-driverless-cars-work. [Accessed 2 February 2020].

[6] A. J. Hawkins, "INSIDE WAYMO'S STRATEGY TO GROW THE BEST BRAINS FOR SELF-DRIVING CARS," The Verge, 9 May 2018. [Online]. Available: https://www.theverge.com/2018/5/9/17307156/google-waymo-driverless-cars-deep-learning-neural-net-interview. [Accessed 2 February 2020].

[7] DIYRobocars, "DIYRobocars: Fast, Cheap and Out of Control," DIYRobocars, [Online]. Available: https://diyrobocars.com/. [Accessed 2 February 2020].