UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

TITOLO IN ITALIANO

TITLE IN ENGLISH

TERROSI FRANCESCO

BONDAVALLI ANDREA
STRIGINI LORENZO

Anno Accademico 2018-2019

ABSTRACT

Abstract

INDICE

ELENCO DELLE TABELLE

ELENCO DELLE FIGURE

1

INTRODUZIONE

Sistemi informatici ormai ovunque (Cosa sono, esempi)

1.1 CYBER-PHYSICAL SYSTEMS OF SYSTEMS

1.2 DEPENDABILITY AND SAFETY

- Dependability
- Safety
- Considerazioni generiche sul perche‘ della tesi

2

AUTOMOTIVE - STATE OF ART

** TODO **

2.1 INTRODUZIONE ALLE SELF-DRIVING CARS QUI?

Automotive technology has been one of the hottest topic of the decade. With the continuously growing hardware and software technologies, completely autonomous vehicles don't seem to be unfeasible anymore: multiple sensors can retrieve high quality data from the surrounding environment and new Artificial Intelligence techniques (i.e. neural networks) are capable of working with this data in a manner that outclasses classical statistical models. However, the use of AI to drive cars, requires more focus on safety and the way to assess it.
Autonomous vehicles can be classified in five levels of autonomy:

Level 0 - No Automation

→ The human driver performs all the tasks, such as steering, accelerating, braking... Cars with *forward collision warning systems* and *lane keep assist* fall in this category

Level 1 - Driver Assistance

→ The vehicle assists the human driver in relatively simple tasks (e.g. adaptive cruise control)

Level 2 - Partial Automation

→ At this level the vehicle is capable of performing more complex tasks (e.g. *Parking assistance, Tesla's Autopilot*) but the driver still must be able to correct unexpected behaviours of the car.

Level 3 - Conditional Automation

→ Level three automation means that the vehicle is now in full control under specific conditions (e.g. riding on a highway). However the human driver still must be able to intervene when requested by the system to do so

Level 4 - High Automation

→ At this level the vehicle can drive completely autonomously, *without* any kind of human interaction. However, they are subjected to specific conditions and assumptions that, when not fulfilled, may result in unexpected behaviours (or catastrophic failures!)

Level 5 - Complete Automation

→ True driverless cars. Human intervention is not needed at all and the car can operate in every condition and environment.

****

2.2 SELF-DRIVING CARS ARCHITECTURE

Descrizione semplificata dell'architettura hardware (sensori) e software (AI controller, safety checker)

2.3 SAFETY NELL'AUTOMOTIVE

- Intro e standard

2.3.1 *Driving Neural Networks*

- Perche' le neural network sono un problema per la safety e perche' e' difficile validarla per questi sistemi | citazioni paperz (RAND study, koopmann, high-dependability systems...)

3

SYSTEM ANALYSIS METHOD

3.1 INTRODUCTION

Safety-Monitors are developed in order to check whether the output of the controller would put the system in hazard given the sensors data, therefore it is desiderable that the hazards covered by the monitor don't overlap with the ones detected by the network. If this will almost certainly hold when the network is in the early stage of training, the lack of literature on the effectiveness of these monitors brought us to develop a methodology to study the monitor's behaviour with respect to learning neural networks as our contribution.

The goal of this work is to develop and to assess the feasability of an experimental method that allows to study the interaction between the AI controller of a self-driving car and the Safety Monitor, with particular attention to these aspectes:

- How much and in what way the benefits given from the use of a safety-monitor varies, the more the neural network learns
- Changes in the safety gain provided by the same monitor when applied to two different networks
- What features of the monitor determines an improvement (or worsening) to the safety of the system
- What aspects of the neural network training have an impact on the monitor usefulness

From now on the AI controller will be referred as the *Primary Component* (or *Primary*), the *Safety-Monitor* (*Monitor*) will be the component meant to read the Primary's output and the sensor data, to correct potential misbehaviours that could bring the system in a catastrophic state.

If not differently specified, we will refer to the failures of the Primary Component simply as *failures* and to the enviroment's state, represented as the data gathered by the sensors, as a *demand*.
A Safety Monitor has basically four possible behaviours:

- True Negative
  - No failure and no alarm raised
- True Positive
  - Failure in the Primary correctly detected by the Monitor
- False Negative
  - Failure in the Primary not detected by the Monitor
- False Positive
  - No failure but Monitor raises alarm

Ideally a Safety Monitor must not only produce just true negatives and true positives. Moreover, for practical use, the hazards detected by the monitor must not overlap with the hazards the neural network is trained to handle (and therefore not hazardous for the system itself).

==========
Mentre in realta' ci dobbiamo accontentare di partial checker ? (grafico e discorso pag. 3)
==========

The reasoning behind this study stems from these considerations: we don't know how the coverage of the Monitor will change when applied to different networks or different learning stages of a network. Moreover, it's not sure whether these checkers will be useful at an advanced state of learning, if not detrimental to the system's safety.
Ideally we can plot the hazards covered by the whole system as those covered by the Primary, and those covered by the Checker as in the next figure:
==========
Grafico overlapping safety
==========

As the network learns, we expect the area covered by the Primary to grow. With a relatively simple Monitor, in relatively simple scenarios,

there will potentially be no overlapping between the hazard areas covered by the two. In this phase, the safety gain provided by the use of a (correctly implemented) safety checker will be remarkable, since the Primary is still learning to handle "easy" demands. As pointed in the previous sections, our main goal is to observe and study the variation of the dependability provided by the monitor when the network is trained to handle "hard" demands, since there are no guarantees on the Monitor's performance in the long period.
As noted in [4] the probability of a failure for a *system* composed by a *Primary Component* and a *Safety-Monitor* on a random demand X is:

$$P_{fp}(1 - Coverage_{\sigma}) - covariance_{Q}(\theta(X), C_{\sigma}(\sigma, X)) \tag{3.1}$$

where:

- $P_{fp}(1 - Coverage_{\sigma})$ is the probability of a failure in the Primary Component (P_{fp}) that is **not detected** by the Safety Monitor (the term $1 - Coverage_{\sigma}$ is exactly the probability of having a false negative/positive)
- $covariance_{Q}(\theta(X), C_{\sigma}, X)$ given a demand profile $Q =< x, y >$ (i.e. the pair <demand, output), measures the correlation between:

$\theta(X)$ - The expected probability that the Primary will fail when processing demand X

$C_{\sigma}(\sigma, X)$ - the term identifying the *Coverage Factor* of the **Monitor**, on the specific demand X

****
IN PROGRESS
****

This formula highlights the deep connection between the safety levels of the Controller and the Monitor, when it comes to the global safety of the system. It is clear from the equation that the probability of observing a failure in the system is also depending on the specific demand X.

****

TO REVIEW:

****

The formula points the fact that to have the probability of observing a failure in the system depends from *all the possible demands* in the *demand space* (i.e. the set of all) Now recall that a demand is a pair $< x, y >$ where x is some sort of representation of the environment, and y the output of the Primary Component (the neural network).
This puts the basis for our study in assessing what (set of inputs)makes some demands harder than others for the network and the monitor and

NEGLI SCENARI SEMPLICI OVERLAPPANO MA SE CAMBIAMO LE CARTE IN TAVOLA? eheHEHEHEHEHE

QUESTE SONO LE DOMANDE DELLA TESI: COSA SUCCEDE SE OVERLAPPANO QUANDO OVERLAPPANO?S

+++ GRAFICO Su "COMPORTAMENTO CORRETTO" ? +++++
It has been proved that Neural Networks can respond to very hard demands in ways that outclasses mankind, when trained properly. From a safety point of view, the learning phase can be seen as in figure 1:

==========

Potenzialmente grafico che va a minimizzare l'area di failures del sistema

==========

Proseguire con coverage dei casi coperti dal monitor come cambia

3.2 EXPERIMENTS METHODOLOGY

The study consists of several experiments in which we observe how the coverage of the safety-monitor (i.e. the probability of raising an alert if there really is a safety-hazard) vary with respect to a neural network in different stages of training.
The first step to perform the analysis is to define what are the metrics of interest and how these can be measured. This task is harder than it seems because it's unknown *a priori* what the probability distribution function of the hazardous scenarios will be. This means that we don't know whether the probability of observing a failure depends on the *running time* of the experiment (e.g. the more the agent drives, the more likely a failure will happen) or if it depends on other factors.

For this reason we decided to measure the length of an experiment in terms of number of failures: given the same initial scenario, two agents (one communicating with the monitor, the other relying solely on the AI) are let driving until n failures happen. We then observe the elapsed time between the start of the experiment and the moment the n_{th} failure happened.
The time to (the n_{th}) failure provides useful informations on the *efficacy* of the monitor. However, this metric itself can't be used alone to assess the potential safety gain provided by safety checking the actions of the neural networks.

****
TO REVIEW:
****

This is why, for each failure, different data were recorded such as:

- Whether or not the monitor raised an alarm
- The change in speed of the car after the alarm was raised
- If a collision with a vehicle V occured, the speed and the direction of V

In this way it's possible to measure more efficiently the overlapping between the set of safety hazards covered by the AI and the one covered by the Monitor, (Other metrics: velocita' a cui andava la macchina), (direzione da cui veniva l'altro veicolo se incidente) —> per capire le situazioni in cui sbaglia di piu'

- Come vengono effettuati gli esperimenti (scenari? durata fissa? ad oltranza? fino ad un fallimento? . . .)
- Misure scelte - estrapolazione misure

4

METHOD IMPLEMENTATION

4.1 TOOLS AND SOFTWARES

4.1.1 *Carla Simulator*

In order to have a realistic environment, with accurate physics simulation and data sensors, the open-source simulator Carla was used. This simulator was developed with the purpose of offering an environment where AI agents can be trained to drive.

- CARLA
- Nervana Systems - coach (Intel)
- Reti neurali su git
- Point Cloud Library per filtrare i dati

4.2 ASSUMPTIONS AND LIMITATIONS?

Dedicare una sezione alle decisioni prese?

4.3 ARCHITETTURA DEL SOFTWARE (ESTRAPOLAZIONE DATI, INTERAZIONE RETE-MONITOR)

- Interazione rete-monitor
- Safety Monitor Implementation - obstacle detection
- Come vengono raccolti i dati
- Come vengono preprocessati

5

RISULTATI DELL'ANALISI

In questa sezione elenchiamo i dati che sono stati raccolti e quali sono i risultati che abbiamo ottenuto (errori ricorrenti, grafici, rapporto monitor-rete neurale)

6

CONCLUSIONS

BIBLIOGRAFIA

[1] Jelena Kocic, Nenad Jovicic, Vujo Drndarevi, *An End-To-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms (2019)*

[2] Qing Rao, Jelena Frtunikj, *Deep learning for self-driving cars: chances and challenges (2018)*

[3] Carlos Zednik, Otto-von-Guericke-Universitat Magdeburg *Solving the Black Box Problem: A Normative Framework for Explainable Artificial Intelligence*

[4] Peter Popov, Lorenzo Strigini *Assessing Asymmetric fault-tolerant Software* (Cited on page 15.)

[5] *https:waymo.com*

[6] *https:uber.com*