**Report project 2: Haven en Transport**

Nils Moerman

24/06/2023

**Create a model that can safely manoeuvre a track made of cones!**

**Introduction:**

* **Project**
  + Assignment
  + Goals
* **EDA**
  + Inspecting the hardcoded client
  + Collecting the data
* **Building and training the model**
* **Experiments**
  + What I’ve found
* **Conclusion**
  + Safety standpoint
  + What have I learned?

**Project**

Assignment:

Create a model that can safely manoeuvre in a digital environment. To do this we could use 2 different types of datasets, Lidar and Sonar. We had to make a neural network that could train on these datasets and make it’s own decisions based on the live information from the scanners.

Goals:

* Collecting the data (Sonar and Lidar)
* Interpret the data
* Design and create an effective neural network
* Train, Test and validate the behaviour of the model
* Safety

**EDA**

Inspecting the hardcoded client:

After we got the assignment I immediately started by looking into the hardcoded client. I wanted to know how it worked, How does it generate the data, where does it save the data and what does it do based on the data it receives.

Once I found the folder where it saves it’s data I realised after a quick look it saves the scanner information plus one more thing. As I thought after a quick look in the hardcoded client again the last column of the dataset was the steering angle. I decided to add the target velocity (speed) to that dataset and then started the gathering part.

Collecting the data:

For this please check the EDA\_lidar notebook first and after that the EDA\_sonar (optional).

*Location: simpylc/notebooks/…*

**Building and training the model**

For this please look at the create\_and\_train notebook.

*Location: simplylc/notebooks/…*

**Experiments**

What I found:

Please check the create\_and\_train notebook for everything I’ve experimented with.

I started with the lidar model, and quickly started to realise that it was very difficult to make this model go around the track without touching a cone or even go off the track completely. Sometimes it will do a full lap without any flaws and sometimes it goes out halfway the first lap. One way or the other it always goes off completely after a while. As this started to frustrate me I decided to go on with the sonar model. This worked after about 2 or 3 training sessions adapting the epochs and learning rate for it.

During my experiments I also came to the conclusion that the loss doesn’t tell of the model is actually good, I’ve had losses below 1 that performed worse than a model with a loss below 10.

So I realised that checking the loss is good for checking if you are going in the right direction but not for rating how good the model performs.

**Conclusion**

Safety standpoint:

From a safety point of view my sonar model is most stable and safe to work with. The lidar model just isn’t stable enough. Why this is I’m not sure, it could be that lidar just has to much information coming through and it ends up interfering with another, or that the hardcoded client just wasn’t optimal for the lidar version.

Either way for this project the sonar version is best by far.

What I’ve learned:

This project I’ve learned to work with Pycharm and how to create classes.

I’ve also learned how to create a neural network and how to tweak set network to get the optimal result.

Personally I really liked this project because of the visualization of it and after learning about neural networks I feel like the possibilities are endless!