Contents

Self-driving car project **2**
Evaluation . . . 2
Environment . . . 2
Conda/Pip . . . 2
Docker . . . 3
Steps . . . 3
Step 1: Load and test the self-driving simulator . . . 3
Step 2: Driving performance for `Track1` . . . 3
Step 3: Driving performance for `Track2` . . . 3
Step 4: Model evaluation . . . 3
Step 5: Discussion . . . 4

Self-driving car project

This research problem is based on a game simulator and python environment. It is related to imitation learning for autonomous driving. You can choose any OS and python tools/IDE of your choice to work on this project.

Evaluation

This work will be reviewed based on the following aspects:

- Your research report explaining your thinking process through all the steps
- Quality of your code and answers to questions
- Code and environment reproducibility

The research report should ideally be in pdf format but feel free to use markdown or jupyter notebook if you prefer. Your Python code, if you provide any, should be version controlled by at least initiating a git repository and committing once.

If your submission is large due to sharing data and/or model, please zip/tar files and put it in the cloud somewhere with a shareable link, or ask us to provide cloud storage space. Please write to us directly if you have a particular question and/or feel like you are stuck and cannot make any progress.

Environment

You can use either conda/pip or docker for this project.

Conda/Pip

We will need to re-create the environment from scratch. First, install conda on your system.

1. Create the environment:

```
conda create -y --name rbk python=3.8
conda activate rbk
conda install pip
```

2. Install pytorch. If you have a GPU:

```
pip3 install torch==1.8.2 torchvision==0.9.2 --extra-index-url
   https://download.pytorch.org/whl/lts/1.8/cu111
```

If you don't:

```
pip3 install torch==1.8.2+cpu torchvision==0.9.2+cpu --extra-index-url
   https://download.pytorch.org/whl/lts/1.8/
```

3. Install the rest of the dependencies. Make sure to install the specific package versions listed here.

```
pip3 install matplotlib pandas
pip3 install flask==0.12.2 flask-socketio==2.9.4 python-engineio==2.0.2
   python-socketio==1.8.4 eventlet==0.22.1 jinja2==3.0.3
   itsdangerous==2.0.1
```

Docker

An alternative to conda/pip is to use directly our docker image. You can pull it from `robok2017/oa_self_driving`. You will need to use the following command to launch docker:

```
docker run --gpus all -it --rm -p 4567:4567 -w /src -v
   [working_folder_path]:/src robok2017/oa_self_driving
```

If you don't have a GPU, use the alternative docker image `robok2017/oa_self_driving:cpu`.

Steps

The steps for this research project are the following:

1. Load and test the self-driving simulator
2. Driving performance for `Track1`
3. Driving performance for `Track2`
4. Model evaluation
5. Discussion

Step 1: Load and test the self-driving simulator

Download the simulator for your OS (Mac, Windows, Linux) provided by Udacity.

Start the simulator. Note that if you are using Linux/Ubuntu, you may not be able to run the simulator in "Windowed" mode, see this issue.

In TRAINING MODE, you can easily record RGB data from the ego-vehicle by using the keyboard to move the car. Make sure you familiarise yourself about the commands by looking into the CONTROLS menu. In AUTONOMOUS MODE, the `drive.py` script establishes a TCP connection with the simulator.

Figure out how to use the pretrained `model.pt` so that the car drives itself in the simulator in `Track1`. You may need to modify `drive.py` if you don't have a GPU.

Step 2: Driving performance for `Track1`

Test the provided pretrained model for `Track1`. How does it perform? Can you find methods to improve the driving performance for `Track1` data without re-collecting data? This could be in form of code or discussion in the report, or both. Note the original training dataset used for `model.pt` can be downloaded here (310M).

Step 3: Driving performance for `Track2`

Test the provided pretrained model for `Track2`. How does it perform? How do you refer to this type of problem in machine learning? What could be done to improve the performance considering there are no particular restrictions? Again, this could be in form of code or discussion in the report, or both.

Step 4: Model evaluation

A simple method was used to evaluate the model performance: a line chart showing training and validation losses versus epochs. You also evaluated the model visually. Do you think this is a good approach to evaluate a driving model? Can you think of any other alternatives?

Step 5: Discussion

Answer and discuss the following questions:

1. Is this learning problem supervised or unsupervised? Is it a reinforcement learning problem?
2. Why does a PID controller is used? Could it be possible to do without?
3. In general, can you discuss the influence of training with more data on model performance? What about training with potentially less data but selecting the most informative samples?
4. Would you classify this project as deterministic or non deterministic? Why? What is the impact on the development of a driving model?
5. According to you, what are the most relevant or impressive research papers in the literature related to this project? Cite a few and explain briefly why you selected them.