Project ID: PW22NKS01

Project Title: Smart Traffic Light Controller Using Deep Reinforcement Learning.

Team Members: Abhishek A (PES1201801935), Krishna P Hegde (PES1201801896), Prathvik Nayak (PES1201802006), A Lakshmi Prasad (PES1201802132)

Project Guide: Prof. Nagegowda K S

Project Abstract: Transportation has become a major priority for people nowadays. While technological advancements have made transportation easier, monitoring and controlling traffic as well as simulating it has become a significant challenge despite the significant research that has been carried out to tackle this problem. An emerging trend to solve this problem involves using deep reinforcement learning techniques which has shown significant progress and promising results in recent studies. There is an increasing demand for developing an intelligent traffic controlling agent which can dynamically adjust to real time traffic rather than just operating by hand-craft rules such as timers. We employ modern deep reinforcement learning methods such as Q-learning and deep neural networks with an agent-based traffic simulator SUMO ("Simulation of Urban MObility") which provides a synthetic yet realistic environment for simulating real-world like traffic and explore the outcomes of the actions that were performed on this environment. Additionally, we also aim to analyze the performance of the agent by comparing it to traditional traffic signals.

Code Execution:

There are two folders named "ANN" and "CNN". The ANN folder contains 5 subfolders named "3-way", "4-way", "5-way", "complex" and "double". The CNN folder contains 4 subfolders named "3-way", "4-way", "5-way" and "double". Each of these subfolders have been named according to the type of road intersection it contains, for example, the subfolder named 3-way in the ANN folder contains all the files required for training and testing the ANN model for a 3-way road intersection and similarly the subfolder named 4-way in the CNN folder contains all the files required for training and testing the CNN model for a 4-way road intersection and so on.

Requirements:

The following needs to be installed on your machine.

1) SUMO Simulator
   Download link: https://www.eclipse.org/sumo/
2) Python
   Download link: https://www.python.org/downloads/
3) Python libraries like numpy, matplotlib, tensorflow.

Training:

Training needs to be carried out on Google Colab. Steps for execution on Colab are given below.

1) Run the below command to remove all the default folders present in the folder named “content” on Colab.
   Command:
   ```
   !rm -r *
   ```
   Now upload all the files from any of the subfolders in ANN or CNN that need to be trained, onto the folder named “content”.

2) Run the below set of commands to install SUMO TraCI
   Commands:
   ```
   !add-apt-repository ppa:sumo/stable -y
   !apt-get update -y
   !apt-get install sumo sumo-tools sumo-doc
   !pip install traci
   import os
   os.environ['SUMO_HOME'] = "/usr/share/sumo/"
   ```

3) Run the below command to start the training process
   Command:
   ```
   !python training_main.py
   ```
   After executing this, training starts and continues for the total number of episodes mentioned in the “training_settings.ini” file.

After the training process is done the trained model gets saved as a file named “trained_model.h5” inside a folder named in the form “model_n” (where n is an integer) in the folder named “models”. In case there are multiple folders inside “models”, the folder with the highest value of n will have the completely trained model and the plots.

Testing:

Steps for testing the trained model are given below.

1) Download the models folder from Colab and place it along with the other files in its respective subfolder inside ANN or CNN on your machine.

2) Run the below command to test the trained model
Command: python testing_main.py
This opens the SUMO Simulator and the simulation can be run by clicking the play button(top left).
Note: Increase the delay (to 50) and zoom in to get a better view of the simulation.

Once the simulation ends all the plots get saved in a new folder called “test” inside the models folder.

The GUI folder contains the file gui.py

Rename the PATH to the path of the project folder where GUI folder is available.

Run this gui.py to see the gui application.

Analysis:

In the Analysis folder there are 3 different folders

1) ANN-CNN comparision - This folder has 4 different folders (3V4V5, 3-way, 4-way, 5-way).

3V4V5 folder contains the necessary data in text format along with results.py file and the images of the final graphs obtained.

On running the results.py file the visualization graph will be displayed.

The 3-way folder contains all the necessary data in text format along with results.py file and the images of the final graphs obtained.

On running the results.py file the visualization graph will be displayed. It also contains a folder named 'test' which contains all the necessary data and the images of different plots.

The 4-way as well as 5-way folders are similar to 3-way.

2)Complex_undertrained

This contains a folder 'test' which contains all the necessary data along with the results.py file and images of different plots.

On running the results.py file the visualization graph will be displayed.

3)Normal vs trained

This folder contains the necessary data in text format along with results.py file and an image of the final graph obtained.

On running the results.py file the visualization graph will be displayed.