1. Install SUMO
2. Make sure the SUMO_HOME system variable is set and points to the SUMO installation path
3. Download TAPAS Cologne and git clone LuST datasets
4. Execute „netconvert -s lust.net.xml --plain-output-prefix“ and „netconvert -s cologne2.net.xml --plain-output-prefix“ in the scenario folders
5. Open clustering_code/dataset-import.py and correct the xml file paths at the bottom of the code
6. Run python dataset-import.py, two csv files should be created dataset-cgn-tl.csv and dataset-lust-tl.csv
7. Fire up Jupyter Notebook and open clustering_code/clustering.ipynb
8. Run all cells, two additional csv files should be created dataset-cgn-tl-clusters.csv and dataset-lust-tl.clusters
9. Run either python pycharm-project\pybrain_learner\dqn_main.py –help to see available parameters or look them up in the code. The file will create a json file containing all rewards of the simulation and dump the neural net’s weights in an h5 file.
10. For repeated runs I created simple shell scripts in the shell_scripts folder
11. Run script.sh 2>&1 | tee log.txt to get the simulation output in a text file. The execution is endless and runs simulations with 500, 1000, 1500, 2000… iterations
12. Parse the output using the pycharm-project\pybrain_learner\traci_result_parser.py file. It will convert the log output to a pandas dataframe compatible csv file.
13. Run the rl-analysis.ipynb notebook to create charts and reward function evaluation

Filename	Path	Explanation
Dqn_main	pycharm-project\pybrain_learner	Main code, run this to start a learning process
Run_sumo_nolearner	pycharm-project\pybrain_learner	Executes a plain sumo simulation run that uses the scenarios own traffic light control
Run_sumo_random	pycharm-project\pybrain_learner	Executes the scenario with random traffic lights
Test_gpu	pycharm-project\pybrain_learner	To see if theano successfully uses the gpu
Traci_result_parser	pycharm-project\pybrain_learner	Use this to parse simulation log output
Dqn_brain,agent,memory, fulldqn_brain, reward_functions	pycharm-project\pybrain_learner	Part of the dqn implementation, see their comments
Dataset-import	Clustering_code	Used for converting sumos xml files to a dataset
Clustering.ipynb	Clustering_code	Jupyter Notebook containing the clustering analysis
Renders	Clustering_code	Some Code i took from another udacity project ☺
Colormap	Clustering_code	A colormap
*	Shell_scripts	Simple linux shell scripts to run simulations with 500 simulation step increments (see step 11.)

rl-analysis.ipynb	reinforcement-result-analysis/	Jupyter notebook for evaluating the simulation results