A Deep Learning and Stochastic Process-Based Framework for Traffic States Estimation—Density, Speed and Flow Direction—Solely By Traffic Flow Data

This paper elaborates on how to utilize the code and data provided in the supplementary materials. The paper is available in two versions: one in .md format and the other in .pdf format, with both versions containing identical content.

Requirements

The code provided in the supplementary materials runs on Python and requires the following packages: torch, math and matplotlib.

Training

To sample for training the fitting model in dataset1, run this command:

```
python sample_for_fitting_model1.py
```

The code will output the extracted samples saved in the file 'data_of_dataset1/output/traindata_for_fitting_model1.pt'. 'preextracted_traindata_for_fitting_model1.pt' is the pre-extracted samples.

To train fitting model for fitting numerical solution in dataset1, run this command:

```
python train_fitting_model_for_dataset1.py
```

The parameters are saved in 'data_of_dataset1/output/fitting_model_for_dataset1.pt', while 'pretrained_fitting_model_for_dataset1.pt' is the pre-trained parameters of the fitting model.

To train inference model in dataset1, run this command:

```
python train_inference_model_for_dataset1.py
```

The parameters are saved in 'data_of_dataset1/output/inference_model_for_dataset2_gat.pt', while 'pretrained_inference_model_for_dataset2_gat.pt' is the pre-trained parameters of the inference model.

The usage method of numerical_solution.py is as follows. The definition of n, l and g is detailed in Algorithms 2 and 3 in the appendix. differential_u is the expectation of U at time t+1. differential_flow is the expectation of F during [t,t+1]. Do not set n too high to avoid numerical overflow. We recommend setting n=4 or 5.

```
equation=Equation(batch_size,d,n,l,g)
differential_u=equation.last_u(v*p,u,b,d)
differential_flow,differential_u=equation.last_output(v*p,u,b,d)
```

Results

To visualize fitting results, run this command:

```
python visualize_fitting_results.py
```

To visualize inference results of U and V, run this command:

```
python visualize_infenence_results_of_u_v.py
```

To visualize inference results of P, run this command:

```
python visualize_infenence_results_of_p.py
```