

0.0 基础

- 此模型需要在Linux环境下运行，需要具备一定的Linux基础知识，预装Anaconda/Miniconda
- 需要英伟达的独立显卡，且部署好cuda以便使用GPU进行训练(测试采用cuda11.3，cuda版本不同可能需要更改pytorch的版本；且由于用户初始环境存在差异，本地python虚拟环境配置仅供参考，遇到报错建议上网搜索)
- Argoverse完整数据集下载时间视网速而定，大致在1~3h不等；单张4070Ti完整训练时间15h以上。

0.1 数据集下载和准备

以Argoverse为例，查看并运行以下代码，创建本地文件夹存放原始数据，有需要可以在.sh文件中修改路径等

```
bash get_data.sh #此步骤可能需要1.5h以上，请耐心等待，并注意程序是否异常，若无法在线下载可以尝试使用梯子
```

get_data.sh内容如下：

```
## prepare data
# step1: download Argoverse HD Maps
wget https://s3.amazonaws.com/argoverse/datasets/av1.1/tars/hd_maps.tar.gz
tar xf hd_maps.tar.gz

# step2: download Argoverse Motion Forecasting **v1.1**
# train + val + test

mkdir dataset && cd dataset
wget
https://s3.amazonaws.com/argoverse/datasets/av1/tars/forecasting_train.tar.gz
wget https://s3.amazonaws.com/argoverse/datasets/av1/tars/forecasting_val.tar.gz
wget
https://s3.amazonaws.com/argoverse/datasets/av1/tars/forecasting_test.tar.gz

tar xvf forecasting_train_v1.1.tar.gz
tar xvf forecasting_val_v1.1.tar.gz
tar xvf forecasting_test_v1.1.tar.gz
```

0.2 本地python虚拟环境配置

```
cd tdc_prediction_example/
conda create --name tdc python=3.7
conda activate tdc
conda install pytorch==1.10.0 torchvision==0.11.0 torchaudio==0.10.0
cudatoolkit=11.3 -c pytorch
# install argoverse api
sudo apt-get install build-essential
pip install camke
pip install lapsolver
```

```
pip install -e ./argoverse-api

# install others dependancy
pip install scikit-image IPython tqdm ipdb
pip install opencv-python
sudo apt install libopenmpi-dev
pip install mpi4py
#Install Horovod and mpi4py for distributed training.

# if you have many GPUs, Recommended install horovod with GPU support, this may
take a while
HOROVOD_GPU_OPERATIONS=NCCL pip install horovod==0.19.4

# if you have only SINGLE GPU, install for code-compatibility
pip install horovod
```

1 数据导入

调用Pred_Dataset.py中的功能函数，实现不同类型的数据导入。

```
# 数据集定义
from Dataset import Pred_Dataset
dataset = Pred_Dataset.init(name="Argoverse", dataset_path='./dataset')
```

2 数据集预处理

Processor文件夹下对不同问题使用的数据预处理函数进行归纳整理。Pred_Processor.py中的功能函数用于预测问题。

其中filter_by_scene函数筛选了交叉口场景数据，time_split函数将数据集划分为2s预测3s的数据片段。

```
# 数据预处理
from Processor import Pred_Processor
Pred_Processor.preprocess(dataset, hist_len=20, fut_len=30,
output_path='./dataset/preprocess')
```

3 模型调用与训练

Example_model文件夹下对不同问题提供了基本的示例模型。Pred_model调用了LaneGCN模型对预处理后的Argoverse数据集进行了模型训练，训练结果保存在./results/LaneGCN路径下。

```
# 模型调用与训练
from Example_model import Pred_model
Pred_model.train(dataset, model='LaneGCN', save_path='./results/LaneGCN',
epochs=5)
```

4 模型评估&可视化

Evaluator针对不同的问题提供了必要的函数用于评价指标，Pred_Evaluator中包括了ADE、FDE、MR等各种函数在内，用于评价轨迹预测结果，visualize函数对预测结果进行了可视化。

```
# 模型评估&可视化
from Evaluator import Pred_Evaluator
Pred_Evaluator.evaluate(dataset, model='LaneGCN',
model_path="lanegcn_best.ckpt", metrics=['ADE', 'FDE', 'MR', 'DAC'])#这里选用了下载链接
中提供的模型，也可以修改为模型保存目录./results/LaneGCN下的其他模型

Pred_Evaluator.visualize(dataset, model="LaneGCN",
model_path="lanegcn_best.ckpt", vis_id=78, fig_path="./78.jpg")
```

5 完整Demo

```
## main.py

# 数据集定义
from Dataset import Pred_Dataset
dataset = Pred_Dataset.init(name="Argoverse", dataset_path='./dataset')

# 数据预处理
from Processor import Pred_Processor
Pred_Processor.preprocess(dataset, hist_len=20, fut_len=30,
output_path='./dataset/preprocess')

# 模型调用与训练
from Example_model import Pred_model
Pred_model.train(dataset, model='LaneGCN', save_path="./results/LaneGCN",
epochs=5)

# 模型评估&可视化
from Evaluator import Pred_Evaluator
Pred_Evaluator.evaluate(dataset, model='LaneGCN',
model_path="lanegcn_best.ckpt", metrics=['ADE', 'FDE', 'MR', 'DAC'])

Pred_Evaluator.visualize(dataset, model="LaneGCN",
model_path="lanegcn_best.ckpt", vis_id=78, fig_path="./78.jpg")
```

结果输出：

```
-----lanegcn模型评估开始-----
6it [00:00, 8.01it/s]
-----
Prediction Horizon : 30, Max #guesses (K): 6
-----
{'minADE': 0.7457006578275933, 'minFDE': 1.0934350079882864, 'MR':
0.12041884816753927, 'DAC': 0.9877835951134382}
-----
-----
Prediction Horizon : 30, Max #guesses (K): 1
-----
{'minADE': 1.4553344826804067, 'minFDE': 3.2266933070040063, 'MR':
0.46596858638743455, 'DAC': 0.9895287958115183}
-----
-----lanegcn模型评估结束-----
```

