TimeMKG readme

1. 数据预处理

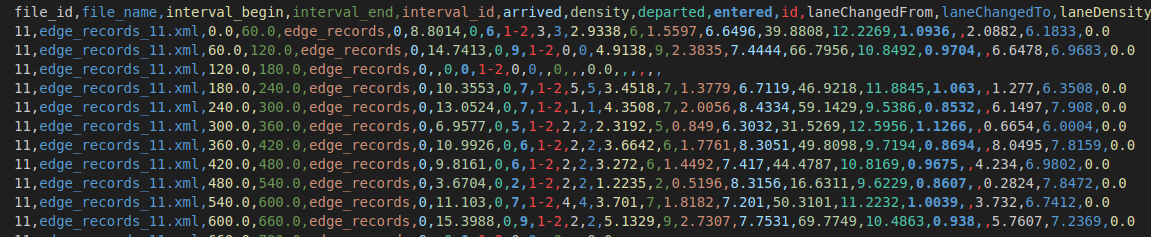
1.1 单节点数据聚合

对于交通数据集原始文件表示一天内，117个id节点的传感器数据，【interval\_begin,interval\_end】表示开始采样点和结束采样点，id表示edge节点。为了适应TimeMKG等主流时间序列模型的数据输入情况，首先将edge\_records\_11.csv-edge\_records\_3300.csv 的数据集按照id节点进行数据划分，聚合为连续采样的时序数据。修改merge.py中的数据集位置，并运行merge.py文件实现数据分割和聚合。

python merge.py

得到的数据结果按照edge id排列，id\_1-2.csv，id\_1-901.csv，id\_1-903.csv

以id\_1-2.csv为例，其结果如下所示：



1.2 多节点数据聚合

为了实现多节点数据的统一训练，可以选择将所有节点（即id\_1-2 - id\_919-46）的文件按照前后顺序聚合为一个文件。修改merge2.py中的数据集位置，并运行merge2.py文件实现多节点数据的聚合。

python merge2.py

得到的结果按照节点id顺序连接，例如id\_1-2 - id\_1-901,得到merged\_traffic.csv（6.09GB）。

1. 模型训练

修改run.py中的数据加载和模型超参数配置，实现模型训练。

# basic config

parser.add\_argument('--task\_name', type=str, default='long\_term\_forecast',

help='task name, options:[long\_term\_forecast, short\_term\_forecast, classification]')

parser.add\_argument('--is\_training', type=int, default=1, help='status')

parser.add\_argument('--model\_id', type=str, default='Traffic', help='model id')

parser.add\_argument('--model', type=str, default='iTransformer',

help='model name, options: [TimeMKG, Autoformer, TimesNet, iTransformer]')

--task\_name：可选择长期预测，短期预测和分类任务

--model：可选择不同的预测模型，内置了多种常用时序预测模型。可先使用iTransformer调试，GPU memory占用较低。

# data loader

parser.add\_argument('--data', type=str, default='Traffic\_Multivariate')

parser.add\_argument('--data\_path', type=str, default='id\_1-2.csv')

parser.add\_argument('--data\_path', type=str, default=''merged\_traffic.csv')

parser.add\_argument('--prompt\_path', type=str, default='traffic/Code/MKG/traffic.txt', help='location of dataset prompt')

--data：依据预测需求选择多变量预测还是单变量预测（例如只预测speed）

--data\_path：依据预测需求选择单个节点或者整合后的多个节点作为数据集

数据集默认按照7：2：1划分训练集：测试集：验证集；先使用单节点数据集调试

--prompt\_path：为TimeMKG的prompt存储路径，可根据具体位置修改

# forecasting task

parser.add\_argument('--seq\_len', type=int, default=96, help='input sequence length') #input length

parser.add\_argument('--label\_len', type=int, default=0, help='start token length')

parser.add\_argument('--pred\_len', type=int, default=96, help='prediction sequence length') # output length

--seq\_len：输入序列的长度，96表示采样点数量，在Traffic数据集中，96表示96mins；

--pred\_len：输出序列的长度，96表示预测后面采样点的数量，即预测后面96mins数量

--seq\_len，-pred\_len可根据具体预测需求修改

# model define

parser.add\_argument('--enc\_in', type=int, default=17, help='encoder input size, Number of input variables')

parser.add\_argument('--dec\_in', type=int, default=17, help='decoder input size, Number of input variables')

parser.add\_argument('--c\_out', type=int, default=1, help='output size, Number of predictors: Singlevariate') # Singlevariate output

parser.add\_argument('--c\_out', type=int, default=17, help='output size, Number of predictors: Singlevariate') # multivariate output

parser.add\_argument('--d\_model', type=int, default=512, help='dimension of model') # Model hyperparameters

parser.add\_argument('--n\_heads', type=int, default=8, help='num of heads') # Model hyperparameters

parser.add\_argument('--e\_layers', type=int, default=2, help='num of encoder layers') # Model hyperparameters

parser.add\_argument('--d\_layers', type=int, default=2, help='num of decoder layers') # Model hyperparameters

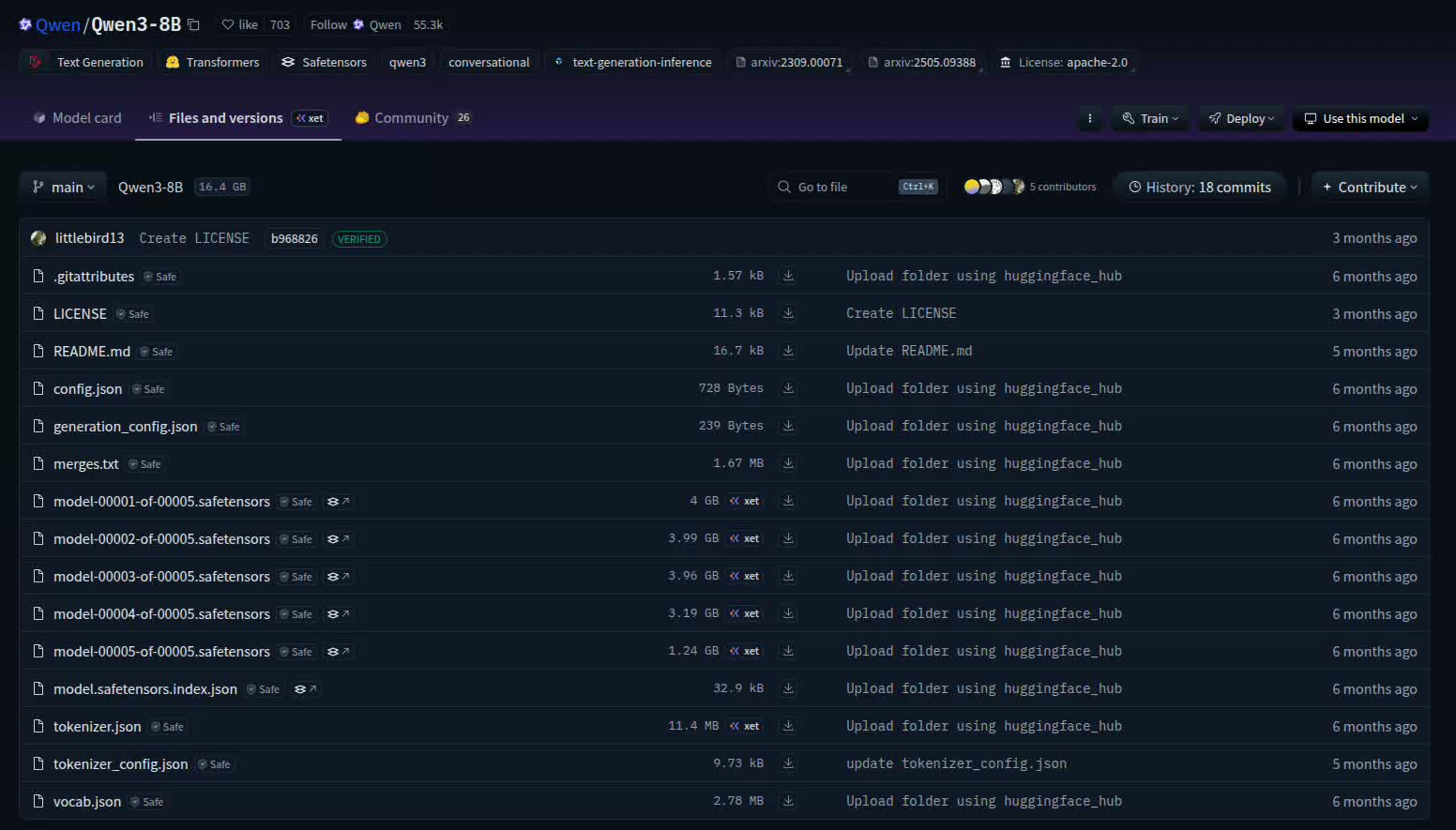
parser.add\_argument('--d\_ff', type=int, default=512, help='dimension of fcn')

--enc\_in和--dec\_in：表示输入变量维度，traffic数据集为17

--c\_out：表示输出变量维度，如果全量输出则修改为17,如果只预测speed但变量则为1.

其余变量为超参数，可根据数据分布和规模适当修改。

如果使用TimeMKG进行模型训练和推理，需下载连线大模型，（例如qwen3 8B模型）并存储于本地，地址如下https://huggingface.co/Qwen/Qwen3-8B/tree/main



将网址上的所有文件下载到本地，并存储在一个文件夹中，并修改对应的model path。

model\_path = "/mnt/petrelfs/sunyifei/qwen3"

self.tokenizer = AutoTokenizer.from\_pretrained(model\_path, trust\_remote\_code=True)

self.LLMmodel = AutoModelForCausalLM.from\_pretrained(model\_path, trust\_remote\_code=True, device\_map=None)

模型参数修改完成后运行run.py文件实现模型训练。