

通用模型readme文档

通用模型构建项目文档

项目概述

本项目实现了一个完整的通用时序模型的训练和预测流程，在经过卫星相关数据预训练后可形成卫星领域通用时序模型，再针对具体领域任务微调可产生诸如卫星温度预测等任务的领域专业模型，可完成时间序列预测、异常值检测、数据插补等任务。项目包括预训练数据准备、模型训练、模型测试与结果可视化四个主要阶段。系统采用Transformer深度学习架构，能有效处理时间序列数据，实现对卫星温度变化的准确预测。项目采用模块化设计，通过根目录下统一定义的ArgParser管理参数，可通过给出的样例脚本进行代码运行，便于实验管理和结果复现。

项目结构

代码块	
1	Timer/
2	├── checkpoints/ # 初始模型权重与训练结果权重目录
3	│ ├── Timer_forecast_1.0.ckpt # 时间序列预测初始权重
4	│ ├── Timer_imputation_1.0.ckpt # 数据插补初始权重
5	│ ├── Timer_anomaly_detection_1.0.ckpt # 异常值检测初始权重
6	│ ├── Timer_bias_forecast_1.0.ckpt # 时间序列预测初始权重（带特征bias）
7	│ ├── Timer_bias_imputation_1.0.ckpt # 数据插补初始权重（带特征bias）
8	│ └── Timer_bias_anomaly_detection_1.0.ckpt # 异常值检测初始权重（带特征bias）
9	├── data_provider/ # 数据集相关代码目录
10	│ ├── data_factory.py # 输入参数选择对应数据集的工具函数
11	│ └── data_loader.py # 数据集定义类
12	├── dataset/ # 数据文件目录
13	├── exp/ # 数据集相关代码目录
14	│ ├── exp_basic.py # 模型训练/测试类的基类定义目录
15	│ ├── exp_forecast.py # 时间序列预测任务的模型训练/测试类
16	│ ├── exp_imputation.py # 数据插补任务的模型训练/测试类
17	│ └── exp_anomaly_forecast.py # 异常值检测任务的模型训练/测试类
18	├── layers/ # 模型子类定义目录
19	│ ├── Attn_Bias.py # 特征间bias类
20	│ ├── Embed.py # 位置编码类
21	│ ├── SelfAttention_Family.py # 自定义注意力机制类
22	│ └── Transformer_EncDec.py # 编码层/解码层类
23	├── models/ # 模型子类定义目录
24	│ └── Timer_multivariate.py # 模型类定义（带任务划分的forward）

```

25 |   └─ TimerBackbone_multivariate.py      # 模型Backbone定义（无任务划分的
    forward)
26 |   └─ pretrain_data_gen/                  # 预训练数据生成目录
27 |   └─ general_model_data_process.py      # 预训练数据生成脚本
28 |   └─ scripts/                           # 各类相关任务的.sh脚本文件目录
29 |   └─ test_results/                      # 各类任务测试结果目录
30 |   └─ utils/                             # 工具函数目录
31 |   └─ └─ masking.py                      # 注意力机制mask类
32 |   └─ └─ tools.py                       # 用于训练的各类工具函数/类
33 |   └─ draw_figure.py                     # 梯度范数/epsilon等数据可视化脚本
34 |   └─ enc_dec_paillier.py                # 数据加/解密脚本
35 |   └─ train_utils.py                     # 训练相关工具函数
36 |   └─ run.py                             # 参数处理与调用对应训练/测试类的脚
    本
37 |   └─ imputation.ipynb                   # 时间序列插补的end2end脚本
38 |   └─ anomaly_detection.ipynb            # 时间序列异常检测的end2end脚本
39 |   └─ app_general_train.py                # 时间序列预测的训练前端streamlit
    代码
40 |   └─ app_general_predict.py              # 时间序列预测的测试前端streamlit
    代码
41 |   └─ requirements.txt                   # 环境包依赖要求

```

模型主要使用的模型类为Timer_multivariate，主要使用的注意力类为TimeAttention，主要使用的数据集类为MultivariateDatasetBenchmark（异常值检测使用MultivariateAnomalyDatasetBenchmark），部分未罗列的.py文件在通用模型部署中并未使用，可以忽略。

环境要求

环境要求

- Python 3.9+
- PyTorch 2.0+
- CUDA (推荐，用于GPU加速)
- 其他依赖库：matplotlib, sklearn, einops, scipy等

安装依赖：

代码块

```
1 pip install -r requirements.txt
```

各模块功能和用法

1. 预训练数据处理模块 (general_model_data_process.py / general_model_data_process_single_series.py)

功能

- 加载卫星遥测数据
- 根据输入转化各特征数据
- 筛选高数据密度区间
- 整合指定的单/多变量各段有效数据为独立文件

用法

代码块

```
1  bash ./scripts/general_data_process.sh
2
3  shell文件内容:
4  python3 ./pretrain_data_gen/general_model_data_process.py \
5      --input_dir /datasda1/wangweipeng/workspace/xw/data_yc/data \
6      --param_dir ./pretrain_data_gen/params \
7      --output_dir ./pretrain_data_gen/general_data_processed \
8      --delta_t 60 \
9      --n_offset 6 \
10     --patch_time 90 \
11     --density_threshold 0.1
12
13  bash ./scripts/general_data_process_single_series.py
14
15  shell文件内容:
16  python3 ./pretrain_data_gen/general_model_data_process_single_variable.py \
17      --input_dir /datasda1/wangweipeng/workspace/xw/data_yc/data \
18      --param_dir ./pretrain_data_gen/params \
19      --output_dir ./pretrain_data_gen/general_data_processed_v2 \
20      --delta_t 60 \
21      --n_offset 6 \
22      --num_workers 8 \
23      --patch_time 90 \
24      --min_point_num 768 \
25      --density_threshold 0
26
27
```

输入/输出

- input_dir: 输入文件目录

- param_dir: 参数文件目录（需要包含X_range.json与data_tuple.json来分别指定处理变量范围与数据范围）

代码块

```

1  input_dir/
2      └── ${satellite_name}/           # 卫星名
3          └── ${time_day}/             # 日期
4              └── xx.orc               # 原始数据文件
5
6  param_dir/
7      ├── data_tuple.json              # 需要处理的卫星-天对
8      └── X_range.json                 # 待处理特征（需指定特征类别与阈值）
9
10 output_dir/
11     ├── raw_data/                    # 从.orc文件预处理转化为.csv文件
12         └── xx.csv
13     ├── middle_data/                 # 处理的中间数据.csv文件
14         └── xx.csv
15     └── processed_data/              # 处理完成的.csv文件
16         └── xx.csv

```

- delta_t: 数据集相邻时间点的时间间隔
- n_offset: 数据集可选时间偏移量数量
- patch_time: 检测数据密度的单位patch的最大时间间隔（最大时间-最小时间）
- density_threshold: 数据密度阈值（在此之上为有效数据段）
- output_dir: 输出文件目录（具有如下格式processed_data子文件夹中的内容可直接作为预训练数据）

2. 数据加解密（enc_dec_paillier.py）

功能

- 生成加密公私钥
- 对.csv数据中所有非'time'列的数据进行加密，从浮点数加密为（明文，密文exponent属性）对
- 对加密数据进行解密，重新转化为浮点数

用法

代码块

```

1  bash ./scripts/enc_dec.sh
2
3  shell文件内容：

```

```

4 python3 enc_dec_paillier.py \
5     --task key_gen \
6     --key_size 2048 \
7     --key_path ./keys
8
9 python3 enc_dec_paillier.py \
10    --task enc \
11    --key_path ./keys \
12    --input_dir ./dataset/xw/elec/trainval \
13    --output_dir ./dataset/xw/elec/trainval_enc
14
15 python3 enc_dec_paillier.py \
16    --task dec \
17    --key_path ./keys \
18    --input_dir ./dataset/xw/elec/trainval_enc \
19    --output_dir ./dataset/xw/elec/trainval_dec

```

输入/输出

- task: 任务类型, 包含["key_gen", "enc", "dec"], 分别代表公私钥生成、加密、解密
- key_size: 公私钥长度 (仅在公私钥生成式使用)
- input_dir: 需要加解密的.csv文件所在目录
- key_path: 公私钥所在目录 (**公私钥生成时为输出目录**, 其余任务为输入目录)
- **output_dir: 输出文件目录 (名称与加解密前的文件名称相同)**

3. 时间序列预测模型预训练/微调/测试

功能

- 预训练完成时间序列预测通用领域任务的模型
- 微调完成时间序列预测专业领域任务的模型
- 测试特定的时间序列预测任务

用法

代码块

```

1 bash ./scripts/forecast/xw_elec_finetune.sh
2
3 shell文件内容:
4 model_name=Timer_multivariate
5 seq_len=672
6 label_len=576
7 input_len=96
8 pred_len=96

```

```
9  output_len=96
10 patch_len=96
11 ckpt_path=checkpoints/Timer_forecast_1.0.ckpt
12 data=multivariate
13
14 torchrun --nnodes=1 --nproc_per_node=2 run.py \
15     --task_name forecast \
16     --is_finetuning 1 \
17     --seed 1 \
18     --ckpt_path $ckpt_path \
19     --root_path ./dataset/xw/elec \
20     --data $data \
21     --model_id Timer_multivariate_forecast \
22     --model $model_name \
23     --seq_len $seq_len \
24     --label_len $label_len \
25     --input_len $input_len \
26     --pred_len $pred_len \
27     --output_len $output_len \
28     --use_norm \
29     --covariate \
30     --n_pred_vars 15 \
31     --e_layers 8 \
32     --factor 3 \
33     --des 'Exp' \
34     --d_model 1024 \
35     --d_ff 2048 \
36     --batch_size 48 \
37     --learning_rate 3e-5 \
38     --finetune_epochs 1000 \
39     --num_workers 4 \
40     --patch_len $patch_len \
41     --train_test 0 \
42     --itr 1 \
43     --gpu 0 \
44     --use_ims \
45     --use_multi_gpu
```

```
46
47 bash ./scripts/forecast/xw_elec_test.sh
```

```
48
49 shell文件内容:
```

```
50 model_name=Timer_multivariate
51 seq_len=672
52 label_len=576
53 input_len=96
54 pred_len=96
55 output_len=256
```

```

56 patch_len=96
57 ckpt_path=checkpoints/xw_elec_forecast_d1024_n8_l8_pruned_without_train/checkpoint.pth
58 data=multivariate
59
60 # test
61 torchrun --nnodes=1 --nproc_per_node=2 run.py \
62     --task_name forecast \
63     --is_finetuning 0 \
64     --seed 1 \
65     --ckpt_path $ckpt_path \
66     --root_path ./dataset/xw/elec \
67     --data $data \
68     --model_id forecast_multivariate_with_covariate \
69     --model $model_name \
70     --seq_len $seq_len \
71     --label_len $label_len \
72     --input_len $input_len \
73     --pred_len $pred_len \
74     --output_len $output_len \
75     --use_norm \
76     --covariate \
77     --n_pred_vars 15 \
78     --e_layers 8 \
79     --factor 3 \
80     --des 'Exp' \
81     --d_model 1024 \
82     --d_ff 2048 \
83     --batch_size 16 \
84     --learning_rate 3e-5 \
85     --finetune_epochs 1000 \
86     --num_workers 4 \
87     --patch_len $patch_len \
88     --train_test 0 \
89     --itr 1 \
90     --gpu 0 \
91     --test_version predict \
92     --use_multi_gpu
93

```

输入

- task_name: 任务类型, 包含["forecast", "imputation", "anomaly_detection"], 分别代表时间序列预测、数据插补、异常值检测3个任务
- is_finetuning: 是否训练, 1代表进行训练, 0代表仅进行测试

- ckpt_path: 如果进行微调/测试, 指定权重文件所在的路径
- root_path: 训练数据集目录
- data: 数据集类型 (没有特殊情况均为multivariate)
- model_id: 产生结果的目录携带的前缀
- model: 指定使用的模型 (没有特殊情况均为Timer_multivariate)
- seq_len: 模型输入的时间点数量
- label_len: 模型输入与模型输出中overlap的时间点数量
- input_len: 模型输出的起始时间点与模型输入的起始时间点相差的时间点数量
- pred_len: 模型输出的结束时间点与模型输入的结束时间点相差的时间点数量 (一般input_len与output_len相等)
- output_len: 模型测试时Rolling Forecast会预测的有效时间点数量
- patch_len: 模型以多少个时间点划分为一个patch
- use_norm: 是否在forward入口/出口对数据进行归一化/反归一化
- covariate: 启用此标志代表任务为带协变量的多变量预测, 否则为普通的多变量预测 (预训练使用普通的多变量预测, 否则各文件之间输出变量/协变量数量不同导致报错), 启用此标志则训练、测试数据的损失计算、数据可视化等均只考虑目标变量
- n_pred_vars: 带协变量的多变量预测中前n_pred_vars个特征为预测变量, 其余的为协变量 (不使用covariate标志则该参数无效)
- e_layers: Transformer Layer层数
- d_model: 模型隐藏层维度
- d_ff: 前馈神经网络中间隐藏层的维度
- batch_size: 训练时dataloader的batch_size, 测试时不管设置多少恒为1
- num_workers: 训练时dataloader的num_workers
- learning_rate: 学习率
- finetune_epochs: 时间序列预测任务的epochs (时间序列预测为finetune_epochs, 其余任务为train_epochs)
- train_test: 在训练后是否进行模型测试, 1代表进行test, 0则直接结束
- test_version: 测试任务类型, 包含["test", "predict", "auto_process", "prune", "visualize"], 其中auto_process在时间序列预测任务中不可用, 其余分别代表单patch预测、Rolling Forecast、模型剪枝、可解释性4个任务
- use_ims: 是否进行自回归训练 (使用则计算损失函数时, 取最后[seq_len:]做比较, 否则取最后[pred_len:])

- use_multi_gpu: 是否使用多卡训练

输出

- 训练结果模型权重将（附带现实时间）保存在./checkpoints文件夹下
- 测试结果将（附带现实时间）保存在./test_results文件夹下

4. 时间序列插补模型预训练/微调/测试

功能

- 预训练完成时间序列插补通用领域任务的模型
- 微调完成时间序列插补专业领域任务的模型
- 测试特定的时间序列插补任务

用法

代码块

```
1  bash ./scripts/imputation/xw_elec_finetune.sh
2
3  shell文件内容:
4  model_name=Timer_multivariate
5  ckpt_path=checkpoints/Timer_imputation_1.0.ckpt
6  d_model=256
7  d_ff=512
8  e_layers=4
9  seq_len=192
10 patch_len=24
11 data=multivariate
12
13 # set data scarcity ratio
14 for subset_rand_ratio in 1
15 do
16     # set mask rate of imputation
17     for mask_rate in 0.25 0.5
18     do
19         python -u run.py \
20             --task_name imputation \
21             --is_finetuning 1 \
22             --seed 1 \
23             --ckpt_path $ckpt_path \
24             --root_path ./dataset/xw/elec \
25             --data $data \
26             --model_id Timer_multivariate_imputation \
27             --model $model_name \
28             --subset_rand_ratio $subset_rand_ratio \
29             --mask_rate $mask_rate \
```

```

30     --use_norm \
31     --covariate \
32     --n_pred_vars 15 \
33     --seq_len $seq_len \
34     --input_len 0 \
35     --output_len 0 \
36     --patch_len $patch_len \
37     --e_layers $e_layers \
38     --factor 3 \
39     --train_test 0 \
40     --batch_size 128 \
41     --d_model $d_model \
42     --d_ff $d_ff \
43     --des 'Exp' \
44     --itr 1 \
45     --use_ims \
46     --learning_rate 0.001 \
47     --train_epochs 100
48 done
49 done
50
51 bash ./scripts/imputation/xw_elec_test_forecast_pretrain.sh
52
53 shell文件内容:
54 model_name=Timer_multivariate
55 ckpt_path=checkpoints/Timer_multivariate_imputation_multivariate_d256_n8_l4_itr
56   0_25-06-05_16-34-18/subset1.0_mask0.125/checkpoint.pth
57 d_model=256
58 d_ff=512
59 e_layers=4
60 seq_len=192
61 patch_len=24
62 data=multivariate
63
64 # set mask rate of imputation
65 for mask_rate in 0.125 0.25 0.5
66 do
67     python -u run.py \
68         --task_name imputation \
69         --is_finetuning 0 \
70         --seed 1 \
71         --ckpt_path $ckpt_path \
72         --root_path ./dataset/xw/elec \
73         --data $data \
74         --model_id Timer_multivariate_imputation \
75         --model $model_name \
76         --mask_rate $mask_rate \

```

```

76     --use_norm \
77     --covariate \
78     --n_pred_vars 15 \
79     --seq_len $seq_len \
80     --input_len 0 \
81     --output_len 0 \
82     --patch_len $patch_len \
83     --e_layers $e_layers \
84     --factor 3 \
85     --train_test 0 \
86     --batch_size 4 \
87     --d_model $d_model \
88     --d_ff $d_ff \
89     --des 'Exp' \
90     --itr 1 \
91     --test_version auto_process
92 done

```

输入

- task_name: 任务类型, 包含["forecast", "imputation", "anomaly_detection"], 分别代表时间序列预测、数据插补、异常值检测3个任务
- is_finetuning: 是否训练, 1代表进行训练, 0代表仅进行测试
- ckpt_path: 如果进行微调/测试, 指定权重文件所在的路径
- root_path: 训练数据集目录
- data: 数据集类型 (没有特殊情况均为multivariate)
- model_id: 产生结果的目录携带的前缀
- model: 指定使用的模型 (没有特殊情况均为Timer_multivariate)
- seq_len: 模型输入的时间点数量
- label_len: 模型输入与模型输出中overlap的时间点数量 (该任务不使用)
- input_len: 模型输出的起始时间点与模型输入的起始时间点相差的时间点数量 (该任务不使用, 需设为0)
- pred_len: 模型输出的结束时间点与模型输入的结束时间点相差的时间点数量 (该任务不使用, 需设为0)
- output_len: 模型测试时Rolling Forecast会预测的有效时间点数量 (该任务不使用, 需设为0)
- patch_len: 模型以多少个时间点划分为一个patch
- use_norm: 是否在forward入口/出口对数据进行归一化/反归一化

- covariate：启用此标志代表任务为带协变量的多变量插补，否则为普通的多变量插补（**预训练使用普通的多变量插补，否则各文件之间输出变量/协变量数量不同导致报错**），启用此标志则训练、测试数据的损失计算、数据可视化等均只考虑目标变量
- n_pred_vars：带协变量的多变量预测中前n_pred_vars个特征为预测变量，其余的为协变量（**不使用covariate标志则该参数无效**）
- e_layers：Transformer Layer层数
- d_model：模型隐藏层维度
- d_ff：前馈神经网络中间隐藏层的维度
- batch_size：训练时dataloader的batch_size，测试时不管设置多少恒为1
- num_workers：训练时dataloader的num_workers
- learning_rate：学习率
- train_epochs：时间序列插补任务的epochs（时间序列预测为finetune_epochs，其余任务为train_epochs）
- train_test：在训练后是否进行模型测试，1代表进行test，0则直接结束
- test_version：测试任务类型，包含["test", "predict", "auto_process", "prune", "visualize"]，其中predict与visualize在时间序列插补任务中不可用，其余分别代表自生成mask的测试集测试、基于测试集nan数据制作mask的测试集测试、模型剪枝
- use_ims：是否进行自回归训练（进行数据插补时模型输出的起始时间点是否向右移动patch_len，有则移动）

输出

- 训练结果模型权重将（附带现实时间）保存在./checkpoints文件夹下
- 测试结果将（附带现实时间）保存在./test_results文件夹下

5. 时间序列异常检测模型预训练/微调/测试

功能

- 预训练完成时间序列异常值检测通用领域任务的模型
- 微调完成时间序列异常值检测专业领域任务的模型
- 测试特定的时间序列异常值检测任务

用法

代码块

```
1  bash ./scripts/anomaly_detection/xw_elec_finetime_forecast_pretrain.sh
2
3  shell文件内容：
4  model_name=Timer_multivariate
```

```

5  ckpt_path=checkpoints/Timer_forecast_1.0.ckpt
6  seq_len=768
7  d_model=1024
8  d_ff=2048
9  e_layers=8
10 patch_len=96
11 dataset_dir="./dataset/xw/elec"
12 data_type=multivariate_anomaly
13
14 # ergodic datasets
15 python -u run.py \
16     --task_name anomaly_detection \
17     --is_finetuning 1 \
18     --ckpt_path $ckpt_path \
19     --root_path $dataset_dir \
20     --data $data_type \
21     --model_id Timer_multivariate_anomaly_detection \
22     --model $model_name \
23     --seq_len $seq_len \
24     --patch_len $patch_len \
25     --use_norm \
26     --covariate \
27     --n_pred_vars 15 \
28     --d_model $d_model \
29     --d_ff $d_ff \
30     --e_layers $e_layers \
31     --train_test 0 \
32     --batch_size 32 \
33     --use_ims \
34     --train_epochs 100
35
36 bash ./scripts/anomaly_detection/xw_elec_test_forecast_pretrain.sh
37 (特别的对于异常值检测, exp.test仅检测单一文件, 因此被遍历测试集需要在shell文件见上写遍
38 历)
39 shell文件内容:
40 model_name=Timer_multivariate
41 ckpt_path=checkpoints/Timer_multivariate_anomaly_detection_multivariate_anomaly
42   _d1024_n8_l8_itr0_25-06-06_16-47-50/checkpoint.pth
43 seq_len=768
44 d_model=1024
45 d_ff=2048
46 e_layers=8
47 patch_len=96
48 dataset_dir="./dataset/xw/elec"
49 data_type=multivariate_anomaly
50
51 for file_path in "$dataset_dir"/test/*

```

```

50 do
51 data=$(basename "$file_path")
52 python -u run.py \
53     --task_name anomaly_detection \
54     --is_finetuning 0 \
55     --ckpt_path $ckpt_path \
56     --root_path $dataset_dir \
57     --data_path $data \
58     --data $data_type \
59     --model_id Timer_multivariate_anomaly_detection \
60     --model $model_name \
61     --seq_len $seq_len \
62     --patch_len $patch_len \
63     --use_norm \
64     --covariate \
65     --n_pred_vars 15 \
66     --loss_threshold 10 \
67     --d_model $d_model \
68     --d_ff $d_ff \
69     --e_layers $e_layers \
70     --test_version test \
71     --batch_size 128 \
72     --use_ims \
73     --train_epochs 10
74 done

```

输入

- task_name: 任务类型, 包含["forecast", "imputation", "anomaly_detection"], 分别代表时间序列预测、数据插补、异常值检测3个任务
- is_finetuning: 是否训练, 1代表进行训练, 0代表仅进行测试
- ckpt_path: 如果进行微调/测试, 指定权重文件所在的路径
- root_path: 训练数据集目录
- data_path: 测试时具体测试.csv文件的名称
- data: 数据集类型 (异常值检测没有特殊情况为multivariate_anomaly)
- model_id: 产生结果的目录携带的前缀
- model: 指定使用的模型 (没有特殊情况均为Timer_multivariate)
- seq_len: 模型输入的时间点数量
- label_len: 模型输入与模型输出中overlap的时间点数量 (该任务不使用)
- input_len: 模型输出的起始时间点与模型输入的起始时间点相差的时间点数量 (该任务不使用, 需设为0)

- pred_len: 模型输出的结束时间点与模型输入的结束时间点相差的时间点数量 (该任务不使用, 需设为0)
- output_len: 模型测试时Rolling Forecast会预测的有效时间点数量 (该任务不使用, 需设为0)
- patch_len: 模型以多少个时间点划分为一个patch
- use_norm: 是否在forward入口/出口对数据进行归一化/反归一化
- covariate: 启用此标志代表任务为带协变量的多变量异常检测, 否则为普通的多变量异常检测 (预训练使用普通的多变量异常检测, 否则各文件之间输出变量/协变量数量不同导致报错), 启用此标志则训练、测试数据的损失计算、数据可视化等均只考虑目标变量
- n_pred_vars: 带协变量的多变量预测中前n_pred_vars个特征为预测变量, 其余的为协变量 (不使用covariate标志则该参数无效)
- e_layers: Transformer Layer层数
- d_model: 模型隐藏层维度
- d_ff: 前馈神经网络中间隐藏层的维度
- batch_size: 训练时dataloader的batch_size, 测试时不管设置多少恒为1
- num_workers: 训练时dataloader的num_workers
- learning_rate: 学习率
- train_epochs: 时间序列插补任务的epochs (时间序列预测为finetune_epochs, 其余任务为train_epochs)
- train_test: 在训练后是否进行模型测试, 1代表进行test, 0则直接结束 (由于特别的测试设置, 需设为0)
- test_version: 测试任务类型, 包含["test", "predict", "auto_process", "prune", "visualize"], 其中predict、auto_process、visualize在时间序列异常检测任务中无法使用, 其余分别代表分别测试集异常检测、模型剪枝、可解释性
- use_ims: 是否进行自回归训练 (进行数据插补时模型输出的起始时间点是否向右移动patch_len, 有则移动)
- loss_threshold: 测试时被判定为异常patch的归一化损失函数阈值

输出

- 训练结果模型权重将 (附带现实时间) 保存在./checkpoints文件夹下
- 测试结果将 (附带现实时间) 保存在./test_results文件夹下

6. 模型剪枝

功能

- 对已训练的模型权重进行剪枝, 以提升推理速度

用法

代码块

```
1  bash ./scripts/xw_elec_prune.sh
2
3  shell文件内容:
4  model_name=Timer_multivariate
5  seq_len=672
6  label_len=576
7  input_len=96
8  pred_len=96
9  output_len=96
10 patch_len=96
11 ckpt_path=checkpoints/xw_elec_forecast_d1024_n8_l8/checkpoint.pth
12 data=multivariate
13
14 torchrun --nnodes=1 --nproc_per_node=2 run.py \
15     --task_name forecast \
16     --is_finetuning 0 \
17     --seed 1 \
18     --ckpt_path $ckpt_path \
19     --root_path ./dataset/xw/elec \
20     --data $data \
21     --model_id forecast_multivariate_with_covariate \
22     --model $model_name \
23     --seq_len $seq_len \
24     --label_len $label_len \
25     --input_len $input_len \
26     --pred_len $pred_len \
27     --output_len $output_len \
28     --use_norm \
29     --covariate \
30     --n_pred_vars 15 \
31     --e_layers 8 \
32     --factor 3 \
33     --des 'Exp' \
34     --d_model 1024 \
35     --d_ff 2048 \
36     --batch_size 16 \
37     --learning_rate 3e-5 \
38     --finetune_epochs 1 \
39     --num_workers 4 \
40     --patch_len $patch_len \
41     --train_test 0 \
42     --itr 1 \
43     --gpu 0 \
```



```
44     --test_version prune \
45     --train_test 0 \
46     --prune_ratio 0.2 \
47     --remove_mask \
48     --use_multi_gpu
49
```

输入

- task_name: 任务类型, 包含["forecast", "imputation", "anomaly_detection"], 分别代表时间序列预测、数据插补、异常值检测3个任务
- is_finetuning: 是否训练, 1代表进行训练, 0代表仅进行测试 (该任务不使用, 需设为0)
- ckpt_path: 如果进行微调/测试, 指定权重文件所在的路径
- root_path: 训练数据集目录
- data: 数据集类型 (异常值检测没有特殊情况为multivariate_anomaly)
- model_id: 产生结果的目录携带的前缀
- model: 指定使用的模型 (没有特殊情况均为Timer_multivariate)
- seq_len: 模型输入的时间点数量
- label_len: 模型输入与模型输出中overlap的时间点数量
- input_len: 模型输出的起始时间点与模型输入的起始时间点相差的时间点数量
- pred_len: 模型输出的结束时间点与模型输入的结束时间点相差的时间点数量
- output_len: 模型测试时Rolling Forecast会预测的有效时间点数量
- patch_len: 模型以多少个时间点划分为一个patch
- use_norm: 是否在forward入口/出口对数据进行归一化/反归一化
- covariate: 启用此标志代表任务为带协变量的多变量任务, 否则为普通的多变量任务, 启用此标志则训练、测试数据的损失计算、数据可视化等均只考虑目标变量
- n_pred_vars: 带协变量的多变量预测中前n_pred_vars个特征为目标变量, 其余的为协变量 (不使用covariate标志则该参数无效)
- e_layers: Transformer Layer层数
- d_model: 模型隐藏层维度
- d_ff: 前馈神经网络中间隐藏层的维度
- batch_size: 训练时dataloader的batch_size, 测试时不管设置多少恒为1
- num_workers: 训练时dataloader的num_workers
- learning_rate: 学习率

- train_epochs: 时间序列插补任务的epochs（时间序列预测为finetune_epochs，其余任务为train_epochs）
- train_test: 是否进行剪枝-训练迭代，还是一片剪除（1代表进行剪枝-训练迭代，0代表一片剪除）
- test_version: 测试任务类型，包含["test", "predict", "prune", "visualize"]，分别代表单patch预测、Rolling Forecast、模型剪枝、可解释性4个任务（该需设为"prune"）
- use_ims: 是否进行自回归训练
- prune_ratio: 剪枝的目标参数剪除率
- remove_mask: 是否在保存时进行mask的去除（即将mask部分权重置为0，视作正常参数）

输出

- 训练结果模型权重将（附带现实时间）保存在./checkpoints文件夹下
- 测试结果将（附带现实时间）保存在./test_results文件夹下

7. 差分隐私训练

功能

- 预训练完成时间序列预测通用领域任务的模型
- 微调完成时间序列预测专业领域任务的模型
- 测试特定的时间序列预测任务

用法

代码块

```

1  bash ./scripts/forecast/xw_elec_security_finetune.sh
2
3  shell文件内容:
4  model_name=Timer_multivariate
5  seq_len=672
6  label_len=576
7  input_len=96
8  pred_len=96
9  output_len=96
10 patch_len=96
11 ckpt_path=checkpoints/Timer_forecast_1.0.ckpt
12 data=multivariate
13
14 # train
15 torchrun --nnodes=1 --nproc_per_node=2 run.py \
16     --task_name forecast \
17     --is_finetuning 1 \
18     --seed 1 \
19     --ckpt_path $ckpt_path \

```

```

20  --root_path ./dataset/xw/elec \
21  --data $data \
22  --model_id Timer_multivariate_forecast \
23  --model $model_name \
24  --seq_len $seq_len \
25  --label_len $label_len \
26  --input_len $input_len \
27  --pred_len $pred_len \
28  --output_len $output_len \
29  --use_norm \
30  --covariate \
31  --n_pred_vars 15 \
32  --e_layers 8 \
33  --factor 3 \
34  --des 'Exp' \
35  --d_model 1024 \
36  --d_ff 2048 \
37  --batch_size 48 \
38  --learning_rate 3e-5 \
39  --finetune_epochs 1000 \
40  --num_workers 4 \
41  --patch_len $patch_len \
42  --train_test 0 \
43  --itr 1 \
44  --gpu 0 \
45  --use_ims \
46  --use_opacus \
47  --use_multi_gpu

```

输入

- task_name: 任务类型, 包含["forecast", "imputation", "anomaly_detection"], 分别代表时间序列预测、数据插补、异常值检测3个任务
- is_finetuning: 是否训练, 1代表进行训练, 0代表仅进行测试
- ckpt_path: 如果进行微调/测试, 指定权重文件所在的路径
- root_path: 训练数据集目录
- data: 数据集类型 (没有特殊情况均为multivariate)
- model_id: 产生结果的目录携带的前缀
- model: 指定使用的模型 (没有特殊情况均为Timer_multivariate)
- seq_len: 模型输入的时间点数量
- label_len: 模型输入与模型输出中overlap的时间点数量

- input_len: 模型输出的起始时间点与模型输入的起始时间点相差的时间点数量
- pred_len: 模型输出的结束时间点与模型输入的结束时间点相差的时间点数量（一般input_len与output_len相等）
- output_len: 模型测试时Rolling Forecast会预测的有效时间点数量
- patch_len: 模型以多少个时间点划分为一个patch
- use_norm: 是否在forward入口/出口对数据进行归一化/反归一化
- covariate: 启用此标志代表任务为带协变量的多变量任务，否则为普通的多变量任务，启用此标志则训练、测试数据的损失计算、数据可视化等均只考虑目标变量
- n_pred_vars: 带协变量的多变量预测中前n_pred_vars个特征为目标变量，其余的为协变量（不使用covariate标志则该参数无效）
- e_layers: Transformer Layer层数
- d_model: 模型隐藏层维度
- d_ff: 前馈神经网络中间隐藏层的维度
- batch_size: 训练时dataloader的batch_size，测试时不管设置多少恒为1
- num_workers: 训练时dataloader的num_workers
- learning_rate: 学习率
- finetune_epochs: 时间序列预测任务的epochs（时间序列预测为finetune_epochs，其余任务为train_epochs）
- train_test: 在训练后是否进行模型测试，1代表进行test，0则直接结束
- test_version: 测试任务类型，包含["test", "predict", "prune", "visualize"]，分别代表单patch预测、Rolling Forecast、模型剪枝、可解释性4个任务
- use_ims: 是否进行自回归训练（使用则计算损失函数时，取最后[:seq_len]做比较，否则取最后[:pred_len]）
- use_multi_gpu: 是否使用多卡训练
- use_opacus: 使用opacus差分隐私注入训练

输出

- 训练结果模型权重将（附带现实时间）保存在./checkpoints文件夹下
- 测试结果将（附带现实时间）保存在./test_results文件夹下

8. 模型可解释性

功能

- 对Transformer模型的注意力权重进行建模
- 建模模型的时间patch-通道热力图

用法

代码块

```
1  bash ./scripts/forecast/xw_elec_visualize.sh
2
3  shell文件内容:
4  model_name=Timer_multivariate
5  seq_len=672
6  label_len=576
7  input_len=96
8  pred_len=96
9  output_len=96
10 patch_len=96
11 ckpt_path=checkpoints/xw_elec_forecast_d1024_n8_l8/checkpoint.pth
12 data=multivariate
13
14 # test
15 torchrun --nnodes=1 --nproc_per_node=2 run.py \
16     --task_name forecast \
17     --is_finetuning 0 \
18     --seed 1 \
19     --ckpt_path $ckpt_path \
20     --root_path ./dataset/xw/elec \
21     --data $data \
22     --model_id forecast_multivariate_with_covariate \
23     --model $model_name \
24     --seq_len $seq_len \
25     --label_len $label_len \
26     --input_len $input_len \
27     --pred_len $pred_len \
28     --output_len $output_len \
29     --use_norm \
30     --covariate \
31     --n_pred_vars 15 \
32     --e_layers 8 \
33     --factor 3 \
34     --des 'Exp' \
35     --d_model 1024 \
36     --d_ff 2048 \
37     --batch_size 1 \
38     --learning_rate 3e-5 \
39     --finetune_epochs 1 \
40     --num_workers 4 \
41     --patch_len $patch_len \
42     --train_test 0 \
43     --itr 1 \
```

```
44     --gpu 0 \  
45     --test_version visualize \  
46     --output_attention \  
47     --use_multi_gpu
```

输入

- task_name: 任务类型, 包含["forecast", "imputation", "anomaly_detection"], 分别代表时间序列预测、数据插补、异常值检测3个任务
- is_finetuning: 是否训练, 1代表进行训练, 0代表仅进行测试
- ckpt_path: 如果进行微调/测试, 指定权重文件所在的路径
- root_path: 训练数据集目录
- data: 数据集类型 (没有特殊情况均为multivariate)
- model_id: 产生结果的目录携带的前缀
- model: 指定使用的模型 (没有特殊情况均为Timer_multivariate)
- seq_len: 模型输入的时间点数量
- label_len: 模型输入与模型输出中overlap的时间点数量
- input_len: 模型输出的起始时间点与模型输入的起始时间点相差的时间点数量
- pred_len: 模型输出的结束时间点与模型输入的结束时间点相差的时间点数量 (一般input_len与output_len相等)
- output_len: 模型测试时Rolling Forecast会预测的有效时间点数量
- patch_len: 模型以多少个时间点划分为一个patch
- use_norm: 是否在forward入口/出口对数据进行归一化/反归一化
- covariate: 启用此标志代表任务为带协变量的多变量任务, 否则为普通的多变量任务, 启用此标志则训练、测试数据的损失计算、数据可视化等均只考虑目标变量
- n_pred_vars: 带协变量的多变量预测中前n_pred_vars个特征为目标变量, 其余的为协变量 (不使用covariate标志则该参数无效)
- e_layers: Transformer Layer层数
- d_model: 模型隐藏层维度
- d_ff: 前馈神经网络中间隐藏层的维度
- batch_size: 训练时dataloader的batch_size, 测试时不管设置多少恒为1
- num_workers: 训练时dataloader的num_workers
- learning_rate: 学习率

- finetune_epochs: 时间序列预测任务的epochs（时间序列预测为finetune_epochs，其余任务为train_epochs）
- train_test: 在训练后是否进行模型测试，1代表进行test，0则直接结束
- test_version: 测试任务类型，包含["test", "predict", "prune", "visualize"]，分别代表单patch预测、Rolling Forecast、模型剪枝、可解释性4个任务（该任务需设置为"visualize"）
- use_ims: 是否进行自回归训练（使用则计算损失函数时，取最后[:seq_len]做比较，否则取最后[:pred_len]）
- use_multi_gpu: 是否使用多卡训练
- output_attention: 模型在forward时是否输出注意力权重（该任务需输出）

输出

- 训练结果模型权重将（附带现实时间）保存在./checkpoints文件夹下
- 测试结果将（附带现实时间）保存在./test_results文件夹下

9.时间序列预测模型微调/测试前端

代码块

```
1  训练前端
2  streamlit run app_general_train.py
3
4  测试前端
5  streamlit run app_general_predict.py
```

输出结果说明

- 存储结果的子文件夹名称如下所示

代码块

```
1  训练
2  {args.model}_{args.task_name}_{args.data}_d{args.d_model}_n{args.n_heads}_l{args.e_layers}_itr{args.itr}_{datetime.now().strftime("%y-%m-%d_%H-%M-%S")}
3  测试
4  {args.model}_{args.task_name}_{args.data}_d{args.d_model}_n{args.n_heads}_l{args.e_layers}_{args.test_version}_{datetime.now().strftime("%y-%m-%d_%H-%M-%S")}
```

- ./checkpoints的子文件夹下存储的权重文件名称固定为checkpoint.pth

常见问题

1. CUDA内存不足

问题：运行训练脚本时出现CUDA内存不足错误

解决方案：

- 减小批处理大小 (batch_size)
- 减小模型维度 (d_model) 或注意力头数 (nhead)
- 减小序列长度 (seq_length)
- 减少特征数量 (n_feat)
- 使用CPU进行训练 (在代码中将device设为'cpu')

2. 数据加载失败

问题：无法正确载入数据集

解决方案：

- 检查配置文件中的路径是否正确
- 确认数据文件格式是否符合要求
- 检查原始数据是否存在于指定位置
- 确保有足够的磁盘空间

3. 训练不收敛

问题：模型训练损失不下降或波动很大

解决方案：

- 调整学习率 (lr) ，通常可以尝试更小的值如1e-5
- 增加训练轮次 (num_epochs)
- 检查数据预处理是否正确
- 尝试使用不同的参数设置
- 使用学习率调度器

4. 预测结果不准确

问题：模型预测结果与实际值偏差较大

解决方案：

- 改进特征选择，添加更多相关参数
- 增加训练数据量，使用更多卫星或更长时间范围的数据
- 调整模型复杂度，增加或减少层数

5. 多种不同数据维度的数据如何进行预训练

问题：各个数据文件的特征数量不同

解决方案：

- 禁用covariate协变量选项，模型的预训练全部采用多变量形式

注：使用前请确保修改配置文件中的路径参数，使其符合您的实际环境。