

Построение виртуальных анализаторов для производственных процессов на основе генеративных нейронных сетей

Рассмотрим производственную систему, осуществляющую технологический процесс.

- Она характеризуется набором датчиков-контроллеров, показания с которых снимаются периодически.
- Есть переменные "качества", расчет значений которых требует лабораторных исследований, временных и экономических ресурсов. Назовем их виртуальными датчиками.

Виртуальные анализаторы (Soft Sensors) - математические модели лабораторных переменных, производящие расчет показателей качества трудноизмеримых переменных (переменных качества) на основе легкоизмеримых производственных данных.

В данной работе проводится исследование и разработка методов прогнозирования лабораторных переменных по физическим в условиях малого набора данных. Для повышения точности исследуются генеративно-состязательные нейронные сети для аугментации данных.

Структура SoftSensors

```
SoftSensors
├──Baseline.ipynb
├──baseline_results
│   ├──i # in {0, 1, 2, 3}
│   └──models
├──Dataset1.csv
├──Dataset_path.txt
├──dataset_utils.py
├──gan.ipynb
├──gan_model.py
├──gan_results
│   ├──i # in {0, 1, 2, 3}
│   └──models
├──gan_varience.ipynb
├──README.md
├──results
│   ├──plots
│   ├──best_fixed.csv
│   └──best_CV.csv
├──results_calc.py
├──soft_sensors_prepare.ipynb
├──train.py
├──Utils.py
├──visualize.ipynb
└──visualize_pickle.ipynb
```

Данные:

- Основной датасет **Dataset1.csv** (находится на визарде, путь указан в Dataset_path.txt)
- Ноутбук **soft sensors_prepare.ipynb** осуществляющий предобработку данных, фильтрацию лабораторных переменных, выбор стабильных участков. Формируется таблица **Dataset1_nans.csv**, которая далее используется.
- Модуль **dataset_utils** с классом для набора данных и методами для выбора временных окон для целевых переменных.

Модели:

- Ноутбук **Baseline.ipynb** для проведения экспериментов, обучения и применения базовых методов.
- Ноутбук **gan.ipynb** для проведения экспериментов, обучения и применения модифицированных регрессоров, обученных вместе с генеративно-сопоставительными сетями.
- В модуле **gan_model** реализованы базовые классы для регрессора, генератора и дискриминатора с функционалом одного шага обучения и применения. Также реализованы классы моделей, основанных на MLP, GRU и LSTM.

Вспомогательные модули:

- Модуль **train.py** с классом для обучения модифицированного метода RcGAN (регрессор с условным GAN).
- Модуль **Utils.py** с необходимыми метриками, EarlyStopping, схемами разбиения для кросс-валидации, обработка и запись результатов.
- Ноутбук **gan_varience.ipynb** для исследования правдоподобности сгенерированных данных.

Результаты:

- Таблицы с результатами применения модели на тренировочной, валидационной и тестовой выборках, а также все параметры моделей и обучения записываются в каталоги **gan_results/i** и **baseline_results/i**, где *i* - индекс лабораторной переменной.
- В каталоги **gan_results/i/models** и **baseline_results/i/models** сохраняются веса моделей.
- Для визуализации и подсчета метрик по кросс-валидации используются ноутбуки **visualize.ipynb** и **visualuze_pickle.ipynb** и функции из модуля **results_calc.py**. Последний сохраняет графики в каталог **results/plots**, а второй ноутбук рассчитывает сводную таблицу по всем моделям, сохраняет лучшие. Таблицы лежат в **results/best_CV.csv** и **results/best_fixed.csv** соответственно.

Система:

Ubuntu 20.04.6 LTS