

SimCLR and Logistic Regression for Time Series Classification

This project implements SimCLR (Simple Framework for Contrastive Learning of Visual Representations) and Logistic Regression for time series classification tasks.

Project Structure

- `weights/`: Data directory
- `graphs.py`: Performance visualization
- `main.py`: Main script for training and testing
- `simclr+resnet/`: SimCLR with ResNet backbone
- `simclr+inceptiontime/`: SimCLR with InceptionTime backbone
- `LR/`: Logistic Regression implementation

Key Components

1. **SimCLRModuleRN**: PyTorch Lightning module for SimCLR with ResNet backbone
2. **SimCLRModuleIT**: PyTorch Lightning module for SimCLR with InceptionTime backbone
3. **NPYDataset** and **NPYDatasetAll**: Custom dataset classes for NumPy array data
4. **Logistic Regression**: GPU-accelerated implementation using cuML

Data

- Training: `x_train` (40000, 100, 60, 12), `y_train` (40000,)
- Testing: `x_test` (10000, 100, 60, 12), `y_test` (10000,)

Setup and Execution

1. Install dependencies: PyTorch, PyTorch Lightning, NumPy, scikit-learn, cuML, tqdm, matplotlib, pandas
2. Place data in `weights/` directory
3. Run: `python main.py`

Configuration

Key parameters (adjust in respective scripts):

- Batch size: 32
- Max epochs: 50
- Learning rate: 0.002
- N values for few-shot learning: [5, 10, 50, 100]

Evaluation

- Few-shot learning performance (5, 10, 50, 100 samples per class)
- Majority voting over 100 time series per sample (5, 10, 50, 100 samples per class)

Logging

- `testRN.log`: SimCLR+ResNet testing
- `testLR.log`: Logistic Regression testing
- `testIT.log`: SimCLR+InceptionTime testing

Results

Check log files for accuracy results. Use `graphs.py` for performance visualizations.

Note

Designed for research on high-dimensional time series data. Requires significant computational resources, especially GPU memory.

References

- [SimCLR: A Simple Framework for Contrastive Learning of Visual Representations](#)
- [InceptionTime: Finding AlexNet for Time Series Classification](#)
- [InceptionTime-Pytorch](#)
- [Lightly SSL](#)