

# Machine Learning Models for Time Series Classification

---

This project focuses on the implementation and evaluation of various machine learning models for time series classification. It includes logistic regression, SimCLR with InceptionTime, and SimCLR with ResNet models, each tailored for specific datasets. The project structure is designed to facilitate easy experimentation with different models and configurations.

## Project Structure

- `python/`: Main directory for Python scripts and models.
  - `dataset.py`: Utility functions for data loading and preprocessing.
  - `graph.py`: Script for generating evaluation graphs from log files.
  - `LR/`: Logistic Regression model and training/testing scripts.
    - `train_testLR.py`: Script for training and testing the Logistic Regression model.
  - `simCLR+InceptionTime/`: Directory for the SimCLR model with InceptionTime.
    - `dataset.py`, `inception.py`, `module_simCLR_IT.py`: Modules for the SimCLR+InceptionTime model.
    - `testIT.py`, `trainIT.py`: Scripts for testing and training the SimCLR+InceptionTime model.
  - `simCLR+resnet/`: Directory for the SimCLR model with ResNet.
    - `dataset.py`, `module_simCLR_RN.py`: Modules for the SimCLR+ResNet model.
    - `testRN.py`, `trainRN.py`: Scripts for testing and training the SimCLR+ResNet model.
  - `main.py`: Main script for the project (if applicable).
- `weights/`: Directory containing datasets and weights for models.
- `testIT.log`, `testLR.log`, `testRN.log`: Log files for model evaluations.
- `readme.md`: This README file.

## Setup

To set up the project, ensure you have Python 3.11 or later installed. It's recommended to use a virtual environment:

```
conda create -n ml -c rapidsai -c conda-forge -c nvidia rapids=24.04
python=3.11 cuda-version=12.2 aeon

pip install tensorflow[and-cuda]

conda activate ml
```

## X\_train, y\_train, X\_test, y\_test format

The input data should be in the following format:

- `X_train`: Training data with shape `(n_samples, n_pixel_per_sample, n_timestamps, n_channels)`.
- `y_train`: Training labels with shape `(n_samples, )`.
- `X_test`: Testing data with shape `(n_samples, n_pixel_per_sample, n_timestamps, n_channels)`.
- `y_test`: Testing labels with shape `(n_samples, )`.

## Training and Testing

To train and test the models, just run the main

```
python main.py
```

This script will train and test the models on the specified datasets. The results will be saved in the log files.

## Evaluation

To generate evaluation graphs from the log files, run the following command:

```
python graph.py
```

This script will generate graphs for the Logistic Regression, SimCLR+InceptionTime, and SimCLR+ResNet models.

## built with

- [pytorch](#) - PyTorch is an open source machine learning library based on the Torch library.
- [InceptionTime-Pytorch](#) - PyTorch implementation of the InceptionTime model.
- [sklearn](#) - Scikit-learn is a free software machine learning library for the Python programming language.
- [SimCLR](#) - Official implementation of SimCLR in PyTorch.
- [ResNet](#) - PyTorch implementation of the ResNet model.
- [matplotlib](#) - Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
- [numpy](#) - NumPy is the fundamental package for scientific computing in Python.
- [tqdm](#) - A fast, extensible progress bar for loops and CLI.
- [pandas](#) - Pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and data manipulation library built on top of the Python programming language.
- [lightly SLL](#) - Lightly is a Python library that helps you to train self-supervised learning models on image data.
- [torchvision](#) - The torchvision package consists of popular datasets, model architectures, and common image transformations for computer vision.

## Authors

- [gygggggggh](#)