**ECG-MoCo-Classfication**

**Background**

In this project, we use all ECGs to train a Siamese network through the contrast learning framework MoCo, and then transfer the pre-trained weights to the downstream classification network to improve the generalization performance of the model.

Wearable ECGs are more prone to myoelectric artifacts, motion artifacts and electrode shedding than standard resting-state ECGs. To improve the robustness of the model, we designed four data augmentation operations for 1D digital multilead ECG signals. These augmentation operations are not only applied in contrast learning, but also effective in improving the classification performance and robustness of the model.

**Programming languages and packages**

This project is based on python 3.7.11. Python packages used are h5py (3.6.0), scipy (1.7.3), numpy (1.21.5), pandas (1.3.5), sklearn (1.0), matplotlib (3.5.1), neurokit2 (0.1.7), pytorch (1.10.1+cu102) and torchvision (0.11.2+cu102).

**Instructions**

The code consists of two parts: pre-training in the first stage and classification in the second stage.

The “stage1\_pretraining” folder contains the following files, and just run “python main.py” in python environment after loading "your own" data to train the MoCo:

“**augmentation.py**”: Four ECG augmentation operations.

“**builder.py**”: Build a MoCo.

“**MSDNN.py**”: Architecture of our multi-scale convolutional networks.

“**main.py**”: Training the MoCo.

The “stage2\_classfication” folder contains the following files, and just run “python main.py” in python environment after loading "your own" data to train the classification network:

“**augmentation.py**”: Four ECG augmentation operations.

“**dataset.py**”: Build your own dataset.

“**function.py**”: Train and test functions.

“**logs.py**”: Save training logs.

“**loss.py**”: Pairwise ranking loss function.

“**DNN.py**”: Architecture of the baseline convolutional networks.

“**MSDNN.py**”: Architecture of our multi-scale convolutional networks.

“**main.py**”: Training the classification network.

**Maintainer**

Jiewei Lai.

**License**

MIT license.