

Contents

Introduction	2
Function Introduction of eegGA for project_EEG	3
1 Functions that require adjustment of parameters	3
1.1 Functions for run properties	3
1.1.1 run_clear	3
1.1.2 run_featureShow	3
1.1.3 run_featureAnalysisShow	3
1.1.4 run_featureClassification	3
1.2 Functions for parameter setting properties	3
1.2.1 setup_basicInfo	3
1.2.2 setup_featureExtraction	3
1.2.3 setup_decomSignal	3
1.2.4 setup_featureAnalysis	3
1.2.5 setup_nosvmClassifier	3
2 Functions that require no adjustment of parameters	3
2.1 Functions for various functional implementations	3
2.1.1 featureExtraction	4
2.1.2 featureShow	4
2.1.3 featureAnalysis	4
2.1.4 featureAnalysisShow	4
2.1.5 featureClassification	4
2.1.6 leaveOneOut_svm	4
2.1.7 nosvmClassifier	4
2.1.8 leaveOneIn_hmm	4
2.1.9 classificationShowHmmEst	4
2.1.10 classificationShowTimeSeq	4
2.1.11 decomSignal	4
2.1.12 gatherReshape	4
2.1.13 violin	4
2.1.14 num2code	4
2.1.15 wave_erode	4
2.1.16 wave_dilate	4
2.1.17 wave_open	4
2.1.18 wave_close	4
2.1.19 wave_PS	4
2.1.20 wave_se	4
2.1.21 variable: featureSet	4
2.1.22 folder: scalp	4
2.2 Functions for processing specific instructions	4
2.2.1 process_featureExtraction	4
2.2.2 process_featureShowEEGblock	4
2.2.3 process_featureShowEEGfeature	4
2.2.4 process_featureAnalysisGather	4
2.2.5 process_featureAnalysisTest	4
2.2.6 process_featureAnalysisShowBarweb	4
2.2.7 process_featureAnalysisShowAnova	4
2.2.8 process_featureAnalysisShowViolin	4
2.2.9 process_featureClassificationVector	4
2.2.10 process_featureClassificationClusterShow	4
2.3 Functions in other folders	4
2.3.1 folder: toolbox	4
2.3.2 folder: matlabUpdate	4
Function Introduction for project_eegLSTM	5
1 Functions that require adjustment of parameters	5
1.1 Functions for run properties	5
1.1.1 run.py	5
1.2 Functions for parameter setting properties	5
1.2.1 config.json	5
2 Functions that require no adjustment of parameters	5
2.1 Functions for various functional implementations	5
2.1.1 process_data.py	5
2.1.2 process_model.py	5
References	6

Introduction

This Introduction is temporary, mainly for the editors and reviewers of the journal ‘Computers in Biology and Medicine’ which has been currently reviewing our manuscript. After our manuscript is officially published, this document will be improved and standardized.

The main part of these codes (especially eegGA) is independently developed by Xiaolong Wu (Copyright) of Beijing University of Science and Technology. Please pay attention to our subsequent updates, and reasonably cite our research results (e.g., Ref. [1]) and use our codes.

The codes in the eegGA part implement feature extraction, group analysis, and subsequent cluster analysis, time series analysis, and establishment of classification models for EEG. In addition, the generated feature set file can also be used for further deep learning in Python language. These two parts correspond to the test_matlab and test_pycharm folders respectively.

For the content of the manuscript that we recently submitted to the journal, the results of these *.m files in the folder test_matlab\project_EEG\data_2019\ correspond to the Fig.* in the manuscript. When using these codes, just copy the test_matlab and test_pycharm folders into the D drive (that is, D:\test_matlab\... and D:\test_pycharm\..., otherwise the user has to modify the path settings in these codes before using them), configure the environment as required in our manuscript, and install the correct version of the program (MATLAB R2014a, PyCharm 2020.1.1, Community Edition; the compilation environment and configuration, Python 3.5, tensorflow-gpu 1.10.0, keras 2.2.2, etc.).

And in order to achieve mental fatigue assessment based on any single-channel, the general process is as follows:

The first is the MATLAB part, which can directly run test_matlab\project_EEG\data_2019\run_all.m. This code (run_all.m) will repeatedly call eegGA\run_featureClassification.m. And through the redefined variable featureSet, eegGA\run_featureClassification.m will repeatedly call eegGA\setup_featureExtraction.m and eegGA\setup_featureAnalysis.m. Finally, feature extraction, analysis, visualization, and feature set construction are completed.

Next is the Python part. By running test_matlab\project_EEG\data_2019\run_result2LSTM_test7_LT_epochAllCh_LSTMCNN.m, the feature sets obtained at the end of the process in MATLAB part are used for LSTM-CNN training. And then by running test_pycharm\project_eegLSTM\data_2019\test7_LT_epochAllCh_LSTMCNN\run.py, the final results of assessment models can be saved in folder test_pycharm\project_eegLSTM\data_2019\test7_LT_epochAllCh_LSTMCNN\saved_models\results\. These final results can be visualized by running test_matlab\project_EEG\data_2019\run_result2LSTM_test7_LT_epochAllCh_LSTMCNN.m again.

Function Introduction of eegGA for project_EEG

1 Functions that require adjustment of parameters

File path: eegGA\...

1.1 Functions for run properties

1.1.1 run_clear

Close figures, clear variables, clear command window, reset default path settings, etc.

1.1.2 run_featureShow

The visualization during feature extraction and in each space and time, for the original signal, time-frequency spectrum diagram, and feature values.

1.1.3 run_featureAnalysisShow

Visualization of the results of feature analysis.

1.1.4 run_featureClassification

Cluster analysis, time series analysis, classification using simple artificial neural networks or support vector machines, and feature set construction and visualization.

1.2 Functions for parameter setting properties

1.2.1 setup_basicInfo

Some basic information settings.

1.2.2 setup_featureExtraction

Setting of feature extraction parameters, including which feature extraction method to use.

1.2.3 setup_decomSignal

Method of signal decomposition (if involved) during feature extraction.

1.2.4 setup_featureAnalysis

Parameter setting of feature analysis mainly refers to hypothesis testing.

1.2.5 setup_nosvmClassifier

The setting of the classifier in function run_featureClassification.

2 Functions that require no adjustment of parameters

2.1 Functions for various functional implementations

File path: eegGA\function\...

Among them, files beginning with “wave_” are algorithms related to mathematical morphology, and another part related to mathematical morphology (generation of structural elements) is located under the path of eegGA\toolbox\decomSignal\MP\.

- 2.1.1 featureExtraction
- 2.1.2 featureShow
- 2.1.3 featureAnalysis
- 2.1.4 featureAnalysisShow
- 2.1.5 featureClassification
- 2.1.6 leaveOneOut_svm
- 2.1.7 nosvmClassifier
- 2.1.8 leaveOneIn_hmm
- 2.1.9 classificationShowHmmEst
- 2.1.10 classificationShowTimeSeq
- 2.1.11 decomSignal
- 2.1.12 gatherReshape
- 2.1.13 violin
- 2.1.14 num2code
- 2.1.15 wave_erode
- 2.1.16 wave_dilate
- 2.1.17 wave_open
- 2.1.18 wave_close
- 2.1.19 wave_PS
- 2.1.20 wave_se
- 2.1.21 variable: featureSet
- 2.1.22 folder: scalp

Some files related to drawing the scalp topographic map.

2.2 Functions for processing specific instructions

File path: eegGA\function\processMeth\...

- 2.2.1 process_featureExtraction
- 2.2.2 process_featureShowEEGblock
- 2.2.3 process_featureShowEEGfeature
- 2.2.4 prosesess_featureAnalysisGather
- 2.2.5 prosesess_featureAnalysisTest
- 2.2.6 prosesess_featureAnalysisShowBarweb
- 2.2.7 prosesess_featureAnalysisShowAnova
- 2.2.8 prosesess_featureAnalysisShowViolin
- 2.2.9 process_featureClassificationVector
- 2.2.10 process_featureClassificationClusterShow

2.3 Functions in other folders

File path: eegGA\...

- 2.3.1 folder: toolbox
- 2.3.2 folder: matlabUpdate

Function Introduction for project_eegLSTM

1 Functions that require adjustment of parameters

File path: project_eegLSTM\data_2019\test7_LT_epochAllCh_LSTMCNN\...

1.1 Functions for run properties

1.1.1 run.py

Establish and train the models of mental fatigue assessment based on single/any-single channel.

1.2 Functions for parameter setting properties

1.2.1 config.json

Set models' architecture.

2 Functions that require no adjustment of parameters

File path: project_eegLSTM\data_2019\test7_LT_epochAllCh_LSTMCNN\core\...

2.1 Functions for various functional implementations

2.1.1 process_data.py

Extract and allocate training data and test data from the *.mat files.

2.1.2 process_model.py

Establish models according to config.json and train the models according to the requirements of run.py.

References

1. Wu, X. and J. Yang, *The superiority verification of morphological features in the EEG-based assessment of depression*. Journal of Neuroscience Methods, 2022. **381**: p. 109690.