

Mimicking the CA1 Encoding Process in Rats During Open-field Behavior Using Deep Learning

This data comes from the Buzsaki Lab. The link is:

<https://buzsakilab.com/wp/projects/entry/11031/>

<https://crcns.org/data-sets/hc/hc-2/about-hc-2>

Code execution is divided into two stages.

Stage One: MATLAB Phase

In this phase, the eegGA toolbox (<https://github.com/Xiaolong-Wu/eegGA>) located in the folder 'toolbox\eegGA' of the 'encodeCA1' directory for EEG analysis is needed. The spikeSorting toolbox in the same folder 'toolbox\spikeSorting' is used for spike extraction-related computations.

1. **Step One:**

Download the data and save it in the folder 'test_matlab\raw\A(source)'.

2. **Step Two:**

Copy the data from folder 'test_matlab\raw\A(source)' to folder 'test_matlab\raw\B(copy)'.

This step is for securely backing up the original data.

3. **Step Three:**

Unzip the corresponding data from folder 'test_matlab\raw\B(copy)' and copy it to folder 'test_matlab\raw\C(data)'. (Such as the selected option in the PNG file in folder 'test_matlab\raw\C(data)')

Run 'run_epochAll.m' and 'run_videoAll.m' to obtain results in the folder 'test_matlab\raw\C(data)\result(epoch)' and folder 'test_matlab\raw\C(data)\result(video)', respectively.

This step is to organize electrophysiological signal files and video files into a format suitable for batch processing.

4. **Step Four:**

Copy data from folder 'test_matlab\raw\C(data)\result(epoch)' to folder 'test_matlab\raw\D(epoch)'. (Such as the selected option in the PNG file in folder 'test_matlab\raw\D(epoch)')

Run 'run_mmExtractionAll.m' and 'run_spikeExtractionAll.m' to obtain results in the folder 'test_matlab\raw\D(epoch)\result(mm)' and folder 'test_matlab\raw\D(epoch)\result(spike)', respectively.

This step is to preprocess and extract features from the electrophysiological signals.

5. **Step Five:**

Copy data from folder 'test_matlab\raw\D(epoch)\result(mm)' and folder 'test_matlab\raw\D(epoch)\result(spike)' to folder 'test_matlab\raw\E(mm)(spike)'. (Such as the selected option in the PNG file in folder 'test_matlab\raw\E(mm)(spike)')

Run 'run_featureMm.m' and 'run_featureSpike.m' to obtain results in the folder 'test_matlab\raw\E(mm)(spike)\result(mm)' and folder 'test_matlab\raw\E(mm)(spike)\result(spike)', respectively.

This step is to organize MM and spike features into a unified format for time-series coding of electrophysiological signals.

6. **Step Six:**

Copy data from folder 'test_matlab\raw\C(data)\result(video)' to folder 'test_matlab\raw\F(video)'. (Such as the selected option in the PNG file in folder 'test_matlab\raw\F(video)')

First, run the Python code in folder 'test_matlab\raw\F(video)\project_posture' to extract the behavioral coordinates of the rats from the video, and obtain results in the folder 'test_matlab\raw\F(video)\project_posture\data\predictions' folder as '.txt' files.

Then, run 'run_cordAll.m' in folder 'test_matlab\raw\F(video)' folder to obtain results in the folder 'test_matlab\raw\F(video)\result(cord_ori)' and folder 'test_matlab\raw\F(video)\result(cord_smooth)', respectively.

This step uses DeepPoseKit (<https://github.com/Xiaolong-Wu/DeepPoseKit>) in Python to extract behavior coordinates from the video and MATLAB to organize and smooth these time-series coordinates for subsequent analysis with MM and spike features.

7. ****Final Steps:****

Copy 'spikeCluster-ch*.mat' from folder 'test_matlab\raw\D(epoch)' to folder 'test_matlab\result(spikeCluster)'.

Copy folder 'test_matlab\raw\D(epoch)\result(mm)' to folder 'test_matlab\result(mm)'.

Copy folder 'test_matlab\raw\D(epoch)\result(spike)' to folder 'test_matlab\result(spike)'.

Copy folder 'test_matlab\raw\F(video)\result(cord_ori)' to folder 'test_matlab\result(cord_ori)'.

Copy folder 'test_matlab\raw\F(video)\result(cord_smooth)' to folder 'test_matlab\result(cord_smooth)'.

- 'run_show1_feature.m' displays the extracted CA1 activity time-series encoding features MM or spike.

- 'run_show2_cord.m' displays the rat's behavior trajectory in the open-field.

- 'run_show3_cordvFeature.m' synchronously displays the rat's behavior trajectory and CA1 activity time-series encoding.

- 'run_data4train.m' constructs training samples using CA1 activity time-series encoding as samples and rat behavior as prediction targets, saving them in the folder 'test_matlab\result(train)'.

Finally, run 'run_dataResult2dataPython.m' in 'test_matlab\result(train)' to prepare the training data for Python.

Stage Two: Python Phase

In this phase, a deep learning model is built using PyTorch to mimic the function of the rat's CA1 during the open-field task.

The data from Stage One has already been copied to the folder 'test_pycharm\data_train' and folder 'test_pycharm\data_test'. Run 'run.py' to start the modeling process.

The built CA1 model and its results are stored in the folder 'test_pycharm\savedModel'.

Abbreviations

MM, mathematical morphology