

## How to use computer codes

##### Environment#####

Ubuntu: 16.04 (or 14.04)

Python: 2.7.12

naoqi: 2.1.4

opencv: 2.4.9

#####

`Experiment' folder consists of programs used to train S-CTRNN model (in the `/learning\_local\_functional\_disconnection/' or `/learning\_long\_functional\_disconnection/') and those used in the robot experiment (in the `/nao/').

### Learning

Recurrent neural network package for problems of learning time-series is in the `/learning\_local\_functional\_disconnection/' or `/learning\_long\_functional\_disconnection/' (hereinafter called '/learning/').

=== Installation ===

First, type `./autogen.sh' in the `/learning/' to create configure file.

Next, type `./configure' and when it finishes, type `make'. This will create `rnn-learn', `rnn-generate' and other utility programs.

Run them with the argument `-h' to show the usages of them.

=== Requirements ===

Building this package requires a C compiler supporting C99 and Autotools (GNU Autoconf, Automake and Libtool).

In addition, utility scripts in the `src/python' directory require python version 2.5 or later (but not python-3.x).

=====

#### 1) Set severity of functional disconnection

Modify `rnn.c' in the `/learning/src/common/', then type `make clean' and `make' in the `/learning/'

Line 63: #define DYS\_CON 0.015 //severity of functional disconnection k

## 2) Start learning

Type `./start.sh &` in the `/learning/example/tst/seed1/`, then wait until finish.  
Optimized parameters are saved in `rnn.dat`.

## 3) Graph

Type `python ./bin/each_print_open.py` in the `/learning/example/tst/seed1/`,  
then python type `python ./bin/plot_rnn.py -f open_orbitXXXXXX.log`. `open_orbitXXXXXX.pdf` is created. Gnuplot is also needed.

## Robot experiment

Programs for robot experiment is in the `/nao/`. Recurrent neural network package for problems of time-series prediction and generation is in the `/nao/rnn_pb_assign_threshold/`.

### === Installation ===

First, type `./autogen.sh` in the `/nao/rnn_pb_assign_threshold/` to create configure file.

Next, type `./configure` and when it finishes, type `make`. This will create `rnn-learn`, `rnn-generate` and other utility programs.

Run them with the argument `-h` to show the usages of them.

### === Requirements ===

Building this package requires a C compiler supporting C99 and Autotools (GNU Autoconf, Automake and Libtool).

In addition, utility scripts in the `/src/python` directory require python version 2.5 or later (but not python-3.x).

=====

## 1) Put `rnn.dat` (optimized parameters) in the `/nao/`.

You can set optimized parameters by modifying `nao_rnn_runner2_2.1.4.py` in the `/nao/`.

Line 381: `rnn_file = './rnn.dat'`

## 2) Start

Type “python nao\_rnn\_runner2\_2.1.4.py IPAddressOfNao” in the `/nao/`. You will see an image window on the computer display which corresponds to vision of Nao, and at that time, Nao will not move. Right-click on the image will trigger the real-time interaction task of Nao controlled by S-CTRNN. Right-click on the image window again will stop Nao.

## 3) Graph

Type `python ./bin/plot.py -f generateXXXXXX.log` in the `/nao/`. `generateXXXXXX.pdf` is created.