

Notes — HerkulesIDE

November 28, 2018; rev. December 12, 2018

Max Daiber-Huppert

1 Introduction

HerkulesIDE has a modular structure which is connected to the folder structure. In the style of the Unix philosophy “everything is a file”. A module can contain the process to calculate the propagation of light like the F2 program. Also a neuronal network can be designed, trained and evaluated. Each module has its own folder structure, which consists of an input, output and an intermediate data folder.

2 ToDo

-

3 Notes

3.1 Program flow

First, the data is loaded from the module and stored in the input folder. Including the data to be calculated and also the configuration data. Then the main purpose of the module can be executed. Afterwards a test and a evaluation of the calculation should be performed automatically. This procedure is shown in figure 1. This process can be executed multiple times in dependency of the number of data records.

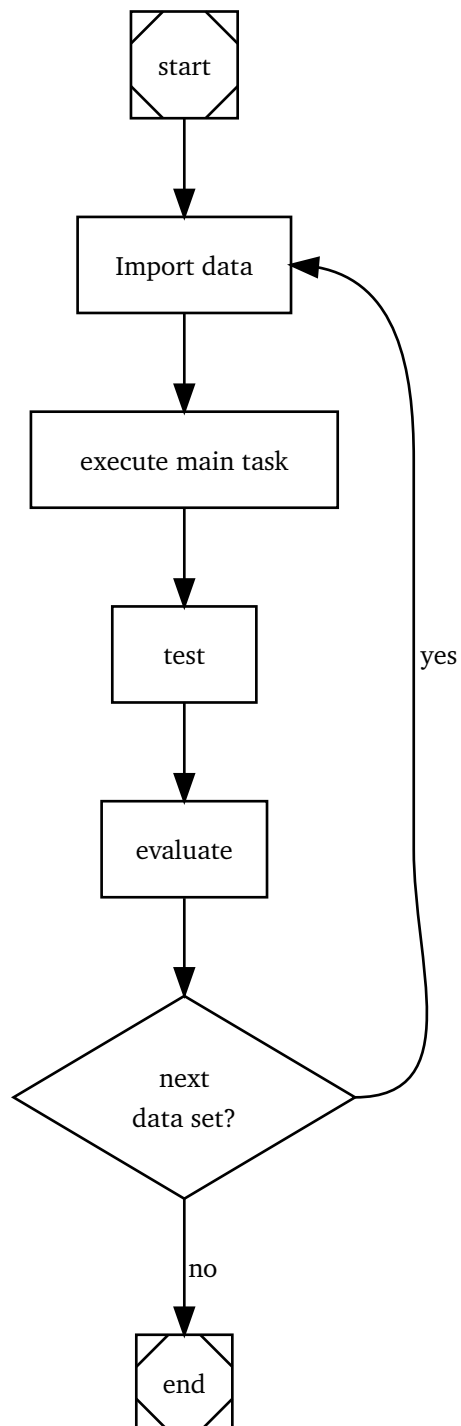


Figure 1: flowchart: module

3.2 Folder structure

3.2.1 HerkulesIDE

Listing 1: Herkules data example folder structure

```
./data
|-- f2
|   |-- input
|   |-- intermediate_data
|   |   |-- scatter_plate
|   |   |-- script
|   |-- output
|   |-- speckle
|   |-- layer_0001
|-- neuronal_network
|   |-- 100_meter // optional subfolder
|   |   |-- input
|   |   |   |-- training_data
|   |   |   |   |-- ground_truth
|   |   |   |-- validation_data
|   |   |   |   |-- ground_truth
|   |   |-- intermediate_data
|   |   |-- history
|   |   |-- trained_weights
|   |   |-- output
|   |   |-- test_data_predictions
|   |-- 10_meter // optional subfolder
|   |   |-- input
|   |   |   |-- training_data
|   |   |   |   |-- ground_truth
|   |   |   |-- validation_data
|   |   |   |   |-- ground_truth
|   |   |-- intermediate_data
|   |   |-- history
|   |   |-- trained_weights
|   |   |-- output
|   |   |-- test_data_predictions
```

3.2.2 Dataset

- 1st layer: beam [gaussian, plane]
- 2nd layer: diameter [d_%2d] // unit: μm
- 3rd layer: fog density [rohn_%3d] // unit: mg/m^3
- 4th layer: sampling rate [sam_%5d]
- 5th layer: fog layer [layer_%4d] // distance is divided by number of layers

Listing 2: Dataset example folder structure

```
./dataset
|-- gaussian
|   |-- d_30
|   |   |-- rohn_200
|   |       |-- sam_04096
|   |           |-- layer_0001
|   |               |-- layer_0100
|   |-- d_40
|   |   |-- rohn_200
|   |       |-- sam_04096
|   |           |-- layer_0100
|-- plane
|   |-- d_30
|   |   |-- rohn_200
|   |       |-- sam_04096
|   |           |-- layer_0001
|   |               |-- layer_0100
|   |-- d_40
|   |   |-- rohn_200
|   |       |-- sam_04096
|   |           |-- layer_0100
```

The filename contains the information about the object name, an optional field that can specify the position of the Gaussian beam and end area to determine whether it represents the intensity, electrical field or argument of the light field.

- filename: [object name]_[,<position of the Gaussian beam>]_[intensity, electrical field, argument].[bmp]
- example: ampelmann1_x_0_y_0_intensity.bmp

3.3 UML Diagrams

The *IModule* interface offers basic functions for data processing. The modules *F2* and *NeuronalNetwork* implements these, see figure 2.

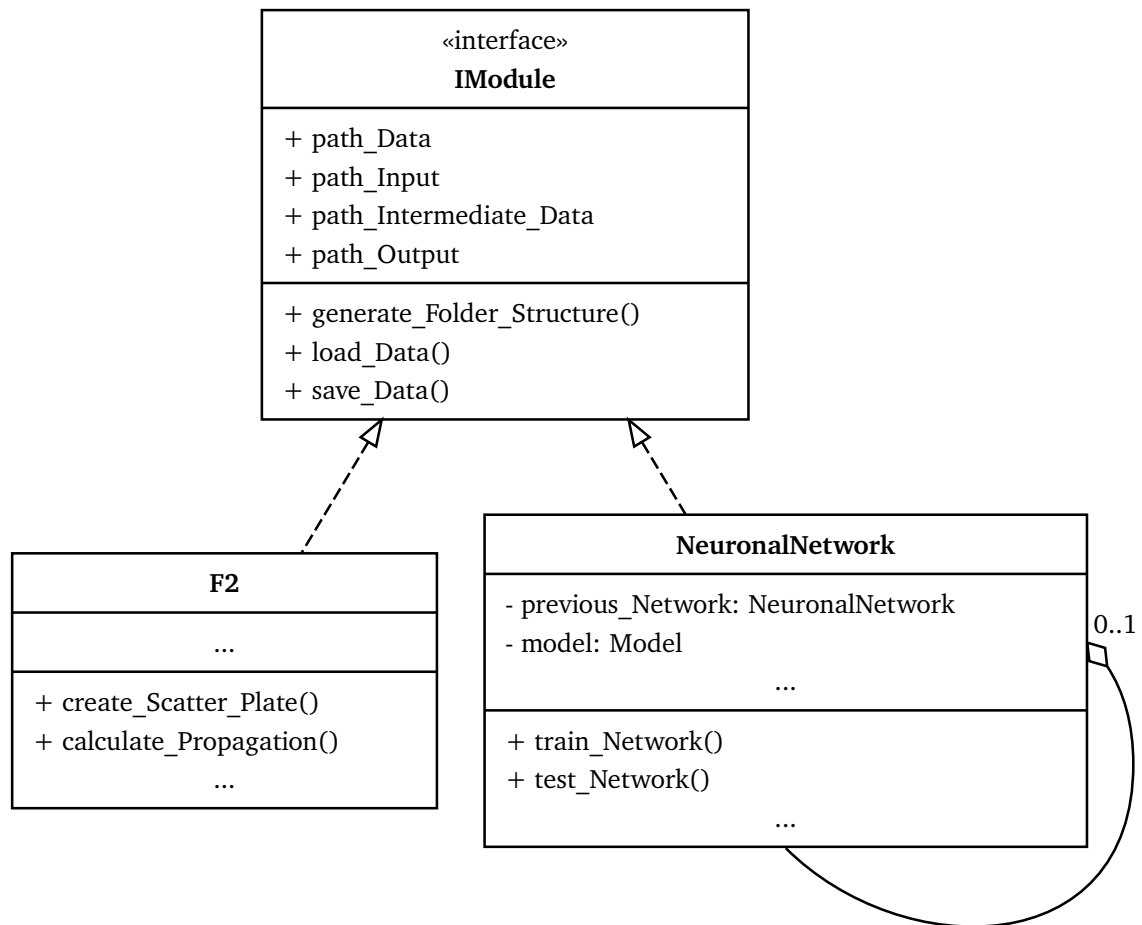


Figure 2: class diagram: HerkulesLib Modules

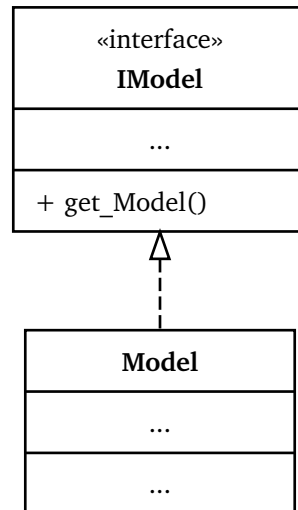



Figure 3: class diagram: HerkulesLib Model

3.4 Constrains

Each module provides its own library. Except a global toolbox which offers functions for highly exchangeable data (e.g. bmp, jpeg, csv, xml, json, pdf, ...) in the folder structure.

3.5 Style guide

- !!! DOCUMENTATION !!!
- code style
 - python
- git
 - gitflow: atlassian, danielKummer
 - one commit contains only one module / project
 - git cheat sheet 
 - frontend
 - * Windows / Mac: Sourcetree

The style of the name of a file, folder, Python function or Python variable is the snake case.

4 Questions

- Shall we setup a own gitlab server?
 - no
- Shall we use github?
 - yes
- Organization with Kanban?
 - Wekan
 - Github porject
- Advanced

- How to save disk space if one module copies files out of another?
 - possible solutions:
 - * create text files with relative path instead of copying all files (inconsistencies?)
 - * file system that can handle copies
 - * zip the project folder (implemented and under test)
- Shall we create a simple GUI, e.g. with Qt?
 - load (test, validation) data; train, validate and test the neuronal network

5 Good to know

Automatically apply PEP8 with *autopep8*.

Listing 3: install autopep8

```
pip install autopep8
```

5.1 Results

Reference