
SOM Documentation

Release 1.0.0

Guilherme Neri

Nov 07, 2023

CONTENTS:

1	kohonen	1
1.1	main module	1
1.2	som package	1
2	Indices and tables	3
	Python Module Index	5
	Index	7

KOHONEN

1.1 main module

<https://github.com/sagnb/OCA>

`main.args()`

Return args

Return arguments passed to the program

`argparse.Namespace`

1.2 som package

1.2.1 Submodules

1.2.2 som.kohonen module

Kohonen Map or SOM(Self Organizing Maps)

class `som.kohonen.SOM`(*input_matrix: ndarray, start_point: ndarray, end_point: ndarray*)

Bases: `object`

Self Organizing Maps

find_winner(*seed: ndarray*)

Find winner in neurons vector

Parameters

seed – `numpy.ndarray`

fit(*max_time: int, max_sigma: float*)

Adjust neuron weights

Parameters

- **max_time** – `int`

- **max_sigma** – `float`

plot_path()

Plot the fit path using `matplotlib`

`som.kohonen.dissimilarity(a: ndarray, b: ndarray, p: int = 2) → float64`

Return the dissimilarity between a and b

Parameters

- **a** – numpy.ndarray
- **b** – numpy.ndarray

Return dissimilarity

numpy.float64

`som.kohonen.gaussian(current_index: int, winner_index: int, current_time: int, max_time: int, max_sigma: float) → float`

Returns the result of the Gaussian function taking into account the topology of the winning neuron and the neuron currently being recalculated

Parameters

- **current_index** – int
- **winner_index** – int
- **current_time** – int
- **max_time** – int
- **max_sigma** – float

Return Gaussian value

float

`som.kohonen.vet(a: ndarray, b: ndarray) → ndarray`

Generates vector between a and b

Parameters

- **a** – numpy.ndarray
- **b** – numpy.ndarray

Return vector

numpy.ndarray

1.2.3 Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

m

`main`, [1](#)

s

`som`, [2](#)

`som.kohonen`, [1](#)

INDEX

A

`args()` (*in module main*), 1

D

`dissimilarity()` (*in module som.kohonen*), 1

F

`find_winner()` (*som.kohonen.SOM method*), 1

`fit()` (*som.kohonen.SOM method*), 1

G

`gaussian()` (*in module som.kohonen*), 2

M

`main`

module, 1

`module`

main, 1

som, 2

som.kohonen, 1

P

`plot_path()` (*som.kohonen.SOM method*), 1

S

`som`

module, 2

`SOM` (*class in som.kohonen*), 1

`som.kohonen`

module, 1

V

`vet()` (*in module som.kohonen*), 2