

---

# **art-python Documentation**

***Release 0.0.1***

**Birk Jones**

**Sep 06, 2019**



---

## Source Code:

---

<b>1</b>	<b>Code</b>	<b>3</b>
1.1	Training Code . . . . .	3
1.2	Testing Code . . . . .	5
1.3	Installation . . . . .	5
1.4	Need Help . . . . .	5
1.5	License . . . . .	5
<b>2</b>	<b>Example</b>	<b>7</b>
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



Fuzzy ART is a ANN architecture that can learn without forgetting. It is similar to human memory where people can recognize their parents even if they have not seen them in a while and have learned many new faces since. The theory was developed by Grossberg and Carpenter and includes various types such as ART 1, ART 2, ART 3, and Fuzzy ART. ART 1 is an architecture that can be used for clustering of binary inputs only. ART 2 improved upon the ART 1 architecture to support continuous inputs. Fuzzy ART, used in the present work, incorporates fuzzy set theory into the pattern recognition process.



Run training code to learn patterns:

```
python train.py
```

Weights are stored (csv or mysql)

Run testing code to identify anomalies:

```
python test.py
```

## 1.1 Training Code

The training code includes the following class and functions:

**class** `train.FuzzyArt` (*x, T, rho, beta, alpha, nep, update*)  
Train using ART Neural Network

### Parameters

- **x** – Input data
- **rho** – Free parameter
- **beta** – Choice Parameter
- **alpha** – Learning Rate
- **nep** – Number of epochs

**create** (*I, T, nc, j*)

Resonance did not occur - create new template

### Parameters

- **I** – Input
- **T** – Template

- **nc** – Number of Categories
- **j** – Input matrix iteration value

**Return T** Template matrix with new template

**update** (*I, T, j, cmax*)

Resonance did occur - update new template

**Parameters**

- **I** – Input
- **T** – Template
- **cmax** – Maximum choice template location
- **j** – Input matrix iteration value

**Return T** Template matrix with update template

**match\_choice** (*c, norm, normI, normT*)

Checks match criterion Compute choice equation Discovers best choice

**Parameters**

- **norm** – minimum of input and templates
- **normI** – norm of input

**Returns** returns category choice location

**template\_options\_loop** (*cmax, chmax, ch, nc*)

Match Criterion

**Parameters**

- **cmax** – Maximum choice (initialized to be -1)
- **chmax** – Match Criterion (initialized to be -1)
- **ch** – Template choice
- **nc** – Number of Categories

**Return cmax** Maximum choice template location

while loop end when ->

**art\_train** (*I, T, T\_length*)

Train ART - Create Template Matrix

**Parameters**

- **I** – Input Matrix
- **T** – Template Matrix
- **cmax** – Max choice (initialized to be -1)
- **chmax** – Match Criterion (initialized to be -1)

**Return T** Final Template



## 1.2 Testing Code

## 1.3 Installation

art-python requires Python (2.7, 3.4, or 3.5) along with several Python package dependencies. Information on installing and using Python can be found at <https://www.python.org/>. Python distributions, such as Anaconda, are recommended to manage the Python interface.

art-python can be installed using pip, git, or a downloaded zip file. Note that the pip installation will NOT include the examples folder referenced in this manual.

**pip:** To install art-python using pip:

```
pip install art-python
```

Required Python package dependencies include:

- Pandas [Mcki13]: used to analyze and store time series data, <http://pandas.pydata.org/>
- Numpy [VaCV11]: used to support large, multi-dimensional arrays and matrices, <http://www.numpy.org/>

## 1.4 Need Help

If you have questions email [cbirkjones@gmail.com](mailto:cbirkjones@gmail.com)

## 1.5 License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



## CHAPTER 2

---

Example

---



## CHAPTER 3

---

### Indices and tables

---

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



### t

test, 5  
train, 3





## A

`art_train()` (*train.FuzzyArt method*), 4

## C

`create()` (*train.FuzzyArt method*), 3

## F

`FuzzyArt` (*class in train*), 3

## M

`match_choice()` (*train.FuzzyArt method*), 4

## T

`template_options_loop()` (*train.FuzzyArt method*), 4

`test` (*module*), 5

`train` (*module*), 3

## U

`update()` (*train.FuzzyArt method*), 4