

LibTopoART  
v0.55

Generated by Doxygen 1.8.8

Wed Sep 24 2014 19:13:23



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	Package LibTopoART . . . . .	7
4.2	Package LibTopoART_samples . . . . .	8
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	LibTopoART.Episodic_TopoART Class Reference . . . . .	9
5.1.1	Detailed Description . . . . .	10
5.1.2	Constructor & Destructor Documentation . . . . .	10
5.1.2.1	Episodic_TopoART . . . . .	10
5.1.2.2	Episodic_TopoART . . . . .	10
5.1.3	Member Function Documentation . . . . .	10
5.1.3.1	Learn . . . . .	10
5.1.4	Member Data Documentation . . . . .	10
5.1.4.1	file_format_version . . . . .	10
5.2	LibTopoART_samples.Episodic_TopoART_sample1 Class Reference . . . . .	11
5.2.1	Detailed Description . . . . .	11
5.3	LibTopoART.F2_output Class Reference . . . . .	11
5.3.1	Detailed Description . . . . .	11
5.3.2	Constructor & Destructor Documentation . . . . .	12
5.3.2.1	F2_output . . . . .	12
5.3.3	Member Data Documentation . . . . .	12
5.3.3.1	bm_cluster_ID . . . . .	12
5.3.3.2	bm_node_activation . . . . .	12
5.3.3.3	bm_node_ID . . . . .	12

5.3.3.4	<a href="#">bm_permanent_cluster_ID</a>	12
5.3.3.5	<a href="#">bm_permanent_node_activation</a>	12
5.3.3.6	<a href="#">bm_permanent_node_ID</a>	12
5.4	<a href="#">LibTopoART.Fast_TopoART Class Reference</a>	12
5.4.1	<a href="#">Detailed Description</a>	13
5.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	13
5.4.2.1	<a href="#">Fast_TopoART</a>	13
5.4.2.2	<a href="#">Fast_TopoART</a>	13
5.4.3	<a href="#">Member Function Documentation</a>	13
5.4.3.1	<a href="#">Learn</a>	13
5.5	<a href="#">LibTopoART.Fast_TopoART_Base Class Reference</a>	14
5.5.1	<a href="#">Detailed Description</a>	15
5.5.2	<a href="#">Member Function Documentation</a>	15
5.5.2.1	<a href="#">ComputeClusterIDs</a>	15
5.5.2.2	<a href="#">GetBMOutput</a>	15
5.5.2.3	<a href="#">Learn</a>	15
5.5.2.4	<a href="#">Save</a>	15
5.5.2.5	<a href="#">SaveText</a>	16
5.5.3	<a href="#">Member Data Documentation</a>	16
5.5.3.1	<a href="#">file_format_version</a>	16
5.5.3.2	<a href="#">float_base_type</a>	16
5.5.3.3	<a href="#">integer_base_type</a>	16
5.6	<a href="#">LibTopoART.Hypersphere_TopoART Class Reference</a>	16
5.6.1	<a href="#">Detailed Description</a>	17
5.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	17
5.6.2.1	<a href="#">Hypersphere_TopoART</a>	17
5.6.2.2	<a href="#">Hypersphere_TopoART</a>	17
5.6.2.3	<a href="#">Hypersphere_TopoART</a>	18
5.6.3	<a href="#">Member Data Documentation</a>	18
5.6.3.1	<a href="#">file_format_version</a>	18
5.7	<a href="#">LibTopoART.ITopoART Interface Reference</a>	18
5.7.1	<a href="#">Detailed Description</a>	19
5.7.2	<a href="#">Member Function Documentation</a>	19
5.7.2.1	<a href="#">ComputeClusterIDs</a>	19
5.7.2.2	<a href="#">GetBMOutput</a>	19
5.7.2.3	<a href="#">Learn</a>	19
5.7.2.4	<a href="#">Save</a>	20
5.7.2.5	<a href="#">SaveText</a>	20
5.8	<a href="#">LibTopoART.ITopoART_R Interface Reference</a>	20
5.8.1	<a href="#">Detailed Description</a>	20

5.8.2	Member Function Documentation	20
5.8.2.1	Learn	20
5.8.2.2	Predict	21
5.9	LibTopoART.LibTopoART_info Struct Reference	21
5.9.1	Detailed Description	21
5.9.2	Member Data Documentation	21
5.9.2.1	networks	21
5.9.2.2	version	22
5.10	LibTopoART.TopoART Class Reference	22
5.10.1	Detailed Description	23
5.10.2	Constructor & Destructor Documentation	23
5.10.2.1	TopoART	23
5.10.2.2	TopoART	24
5.10.3	Member Function Documentation	24
5.10.3.1	ComputeClusterIDs	24
5.10.3.2	GetBMOOutput	24
5.10.3.3	Learn	24
5.10.3.4	Save	24
5.10.3.5	SaveText	24
5.10.4	Member Data Documentation	25
5.10.4.1	file_format_version	25
5.10.4.2	float_base_type	25
5.10.4.3	integer_base_type	25
5.10.4.4	x_F0	25
5.10.4.5	x_F0_dim	25
5.11	LibTopoART.TopoART_R Class Reference	25
5.11.1	Detailed Description	26
5.11.2	Constructor & Destructor Documentation	26
5.11.2.1	TopoART_R	26
5.11.2.2	TopoART_R	26
5.11.3	Member Function Documentation	26
5.11.3.1	Learn	26
5.11.3.2	Predict	27
5.11.4	Member Data Documentation	27
5.11.4.1	file_format_version	27
5.12	LibTopoART.TopoART_R_Prediction Class Reference	27
5.12.1	Detailed Description	28
5.12.2	Constructor & Destructor Documentation	28
5.12.2.1	TopoART_R_Prediction	28
5.12.3	Member Function Documentation	28

---

5.12.3.1	<a href="#">PrintPredictions</a>	28
5.12.4	<a href="#">Member Data Documentation</a>	28
5.12.4.1	<a href="#">d_vec_prediction</a>	28
5.12.4.2	<a href="#">i_vec_prediction</a>	28
5.12.4.3	<a href="#">NO_PREDICTION</a>	28
5.13	<a href="#">LibTopoART_samples.TopoART_R_sample1 Class Reference</a>	28
5.13.1	<a href="#">Detailed Description</a>	28
5.14	<a href="#">LibTopoART_samples.TopoART_sample1 Class Reference</a>	28
5.14.1	<a href="#">Detailed Description</a>	29
5.15	<a href="#">LibTopoART_samples.TopoART_sample2 Class Reference</a>	29
5.15.1	<a href="#">Detailed Description</a>	29

# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">LibTopoART</a> . . . . .	7
<a href="#">LibTopoART_samples</a> . . . . .	8





## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

LibTopoART.samples.Episodic_TopoART_sample1 . . . . .	11
LibTopoART.F2_output . . . . .	11
LibTopoART.ITopoART . . . . .	18
LibTopoART.Fast_TopoART_Base . . . . .	14
LibTopoART.Episodic_TopoART . . . . .	9
LibTopoART.Fast_TopoART . . . . .	12
LibTopoART.TopoART . . . . .	22
LibTopoART.Hypersphere_TopoART . . . . .	16
LibTopoART.TopoART_R . . . . .	25
LibTopoART.ITopoART_R . . . . .	20
LibTopoART.TopoART_R . . . . .	25
LibTopoART.LibTopoART_info . . . . .	21
LibTopoART.TopoART_R_Prediction . . . . .	27
LibTopoART.samples.TopoART_R_sample1 . . . . .	28
LibTopoART.samples.TopoART_sample1 . . . . .	28
LibTopoART.samples.TopoART_sample2 . . . . .	29



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">LibTopoART.Episodic_TopoART</a>	
Class <a href="#">Episodic_TopoART</a> provides an implementation of the Episodic <a href="#">TopoART</a> neural network as proposed in "Marko Tscherepanow, Sina Kühnel, and Sören Riechers, 'Episodic Clustering of Data Streams Using a Topology-Learning Neural Network', European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29, 2012."	9
<a href="#">LibTopoART_samples.Episodic_TopoART_sample1</a>	
Sample using artificial two-dimensional data	11
<a href="#">LibTopoART.F2_output</a>	
Class <a href="#">F2_output</a> provides the output of a single <a href="#">TopoART</a> module. It is a compressed version of the output vectors y and c.	11
<a href="#">LibTopoART.Fast_TopoART</a>	
Class <a href="#">Fast_TopoART</a> provides an implementation of the <a href="#">TopoART</a> neural network as proposed in "Tscherepanow, Marko (2010). <a href="#">TopoART</a> : A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer."	12
<a href="#">LibTopoART.Fast_TopoART_Base</a>	
Base class providing functionality common to several <a href="#">TopoART</a> networks.	14
<a href="#">LibTopoART.Hypersphere_TopoART</a>	
Class <a href="#">Hypersphere_TopoART</a> provides an implementation of the Hypersphere <a href="#">TopoART</a> neural network as proposed in "Tscherepanow, Marko (2012). Incremental On-line Clustering with a Topology-Learning Hierarchical ART Neural Network Using Hyperspherical Categories. In Poster and Industry Proceedings of the Industrial Conference on Data Mining (ICDM), pp. 22–34. Fockendorf, Germany: ibai-publishing."	16
<a href="#">LibTopoART.ITopoART</a>	
Interface summarising the basic <a href="#">TopoART</a> functionality.	18
<a href="#">LibTopoART.ITopoART_R</a>	
Interface summarising the basic TopoART-R functionality.	20
<a href="#">LibTopoART.LibTopoART_info</a>	
Struct <a href="#">LibTopoART_info</a> provides some metainformation regarding the respective implementation of <a href="#">LibTopoART</a> .	21
<a href="#">LibTopoART.TopoART</a>	
Class <a href="#">TopoART</a> provides an implementation of the <a href="#">TopoART</a> neural network as proposed in "Tscherepanow, Marko (2010). <a href="#">TopoART</a> : A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), LNCS 6354, pp. 157–167. Berlin, Germany: Springer."	22

[LibTopoART.TopoART\\_R](#)

Class [TopoART\\_R](#) provides an implementation of the TopoART-R neural network as proposed in "Tscherepanow, Marko (2011). An Extended [TopoART](#) Network for the Stable On-Line Learning of Regression Functions. In Proceedings of the International Conference on Neural Information Processing (ICONIP), LNCS 7063, pp. 562–571. Berlin, Germany: Springer . . . . . 25

[LibTopoART.TopoART\\_R\\_Prediction](#)

Class [F2\\_output](#) contains a prediction made by a TopoART-R network. . . . . 27

[LibTopoART\\_samples.TopoART\\_R\\_sample1](#)

Function approximation using TopoART-R . . . . . 28

[LibTopoART\\_samples.TopoART\\_sample1](#)

Simple TopoART sample . . . . . 28

[LibTopoART\\_samples.TopoART\\_sample2](#)

Sample using artificial two-dimensional data . . . . . 29

## Chapter 4

# Namespace Documentation

### 4.1 Package LibTopoART

#### Classes

- class [Episodic\\_Top ART](#)

Class [Episodic\\_Top ART](#) provides an implementation of the [Episodic TopoART](#) neural network as proposed in "Marko Tscherepanow, Sina Kühnel, and Sören Riechers, 'Episodic Clustering of Data Streams Using a Topology-Learning Neural Network', European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29, 2012."

- class [F2\\_output](#)

Class [F2\\_output](#) provides the output of a single [TopoART](#) module. It is a compressed version of the output vectors  $y$  and  $c$ .

- class [Fast\\_Top ART](#)

Class [Fast\\_Top ART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Tscherepanow, Marko (2010). [TopoART](#): A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer."

- class [Fast\\_Top ART\\_Base](#)

Base class providing functionality common to several [TopoART](#) networks.

- class [Hypersphere\\_Top ART](#)

Class [Hypersphere\\_Top ART](#) provides an implementation of the [Hypersphere TopoART](#) neural network as proposed in "Tscherepanow, Marko (2012). Incremental On-line Clustering with a Topology-Learning Hierarchical ART Neural Network Using Hyperspherical Categories. In Poster and Industry Proceedings of the Industrial Conference on Data Mining (ICDM), pp. 22–34. Fockendorf, Germany: ibai-publishing."

- interface [ITopoART](#)

Interface summarising the basic [TopoART](#) functionality.

- interface [ITopoART\\_R](#)

Interface summarising the basic [TopoART-R](#) functionality.

- struct [LibTopoART\\_info](#)

Struct [LibTopoART\\_info](#) provides some meta-information regarding the respective implementation of [LibTopoART](#).

- class [TopoART](#)

Class [TopoART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Tscherepanow, Marko (2010). [TopoART](#): A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), LNCS 6354, pp. 157–167. Berlin, Germany: Springer."

- class [TopoART\\_R](#)

Class [TopoART\\_R](#) provides an implementation of the [TopoART-R](#) neural network as proposed in "Tscherepanow, Marko (2011). An Extended [TopoART](#) Network for the Stable On-Line Learning of Regression Functions. In Proceedings of the International Conference on Neural Information Processing (ICONIP), LNCS 7063, pp. 562–571. Berlin, Germany: Springer.

- class [TopoART\\_R\\_Prediction](#)

*Class [F2\\_output](#) contains a prediction made by a TopoART-R network.*

## 4.2 Package LibTopoART\_samples

### Classes

- class [Episodic\\_Top ART\\_sample1](#)  
*Sample using artificial two-dimensional data.*
- class [TopoART\\_R\\_sample1](#)  
*Function approximation using TopoART-R.*
- class [TopoART\\_sample1](#)  
*Simple TopoART sample.*
- class [TopoART\\_sample2](#)  
*Sample using artificial two-dimensional data.*

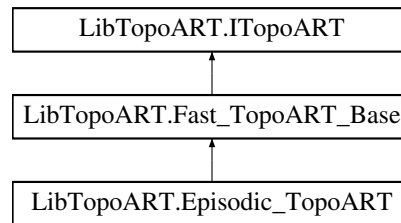
## Chapter 5

# Class Documentation

### 5.1 LibTopoART.Episodic\_TopoART Class Reference

Class [Episodic\\_TopoART](#) provides an implementation of the Episodic [TopoART](#) neural network as proposed in "Marko Tscherepanow, Sina Kühnel, and Sören Riechers, 'Episodic Clustering of Data Streams Using a Topology-Learning Neural Network', European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29, 2012."

Inheritance diagram for LibTopoART.Episodic\_TopoART:



#### Public Member Functions

- [Episodic\\_TopoART](#) (long input\_dimension, long module\_number, decimal rho\_a\_value, long t\_max\_value)  
*This constructor initialises an Episodic [TopoART](#) network.*
- [Episodic\\_TopoART](#) (string path)  
*This constructor loads a saved Episodic [TopoART](#) network.*
- override void [Learn](#) (decimal[] input)  
*This method performs a single training step.*

#### Static Public Attributes

- new static readonly decimal [file\\_format\\_version](#) = 0.01m  
*Instance variable `file_format_version` represents the version of the file format used by class [Episodic\\_TopoART](#).*

#### Properties

- long [T\\_max](#) [get]  
*Property `T_max` represents the considered time frame.*

### 5.1.1 Detailed Description

Class [Episodic\\_TopoART](#) provides an implementation of the Episodic [TopoART](#) neural network as proposed in "Marko Tscherepanow, Sina Kühnel, and Sören Riechers, 'Episodic Clustering of Data Streams Using a Topology-Learning Neural Network', European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29, 2012."

### 5.1.2 Constructor & Destructor Documentation

#### 5.1.2.1 `LibTopoART.Episodic_TopoART.Episodic_TopoART ( long input_dimension, long module_number, decimal rho_a_value, long t_max_value )`

This constructor initialises an Episodic [TopoART](#) network.

Parameters

<i>input_dimension</i>	The dimension of input vectors to be learnt.
<i>module_number</i>	The number of Episodic <a href="#">TopoART</a> modules.
<i>rho_a_value</i>	The vigilance parameter of the first Episodic <a href="#">TopoART</a> module (ETA a).
<i>t_max_value</i>	The parameter limiting the considered time frame.

#### 5.1.2.2 `LibTopoART.Episodic_TopoART.Episodic_TopoART ( string path )`

This constructor loads a saved Episodic [TopoART](#) network.

Parameters

<i>path</i>	The path of a binary Episodic <a href="#">TopoART</a> file.
-------------	---

### 5.1.3 Member Function Documentation

#### 5.1.3.1 `override void LibTopoART.Episodic_TopoART.Learn ( decimal[] input ) [virtual]`

This method performs a single training step.

The spatial weights are adapted as in the original [TopoART](#) network. In contrast, the adaptation of the temporal weight  $w_{\{j, 2\}^{F2, t}}$  occurring only in Episodic [TopoART](#) is slightly different:  $w_{\{j, 2\}^{F2, t}}(t+1) = \text{Beta}_j * \text{Max}(t_{2^{F1}}(t), w_{\{j, 2\}^{F2, t}}(t)) + (1 - \text{Beta}_j) * w_{\{j, 2\}^{F2, t}}(t)$  for  $j = \text{bm}$  or  $j = \text{sbm}$ . (Note:  $w_{\{j, 1\}^{F2, t}}$  remains constant over the life time of a node.)

Parameters

<i>input</i>	The input vector to be learnt.
--------------	--------------------------------

Implements [LibTopoART.Fast\\_TopoART\\_Base](#).

### 5.1.4 Member Data Documentation

#### 5.1.4.1 `new static readonly decimal LibTopoART.Episodic_TopoART.file_format_version = 0.01m [static]`

Instance variable `file_format_version` represents the version of the file format used by class [Episodic\\_TopoART](#).



## 5.2 LibTopoART\_samples.Episodic\_TopoART\_sample1 Class Reference

Sample using artificial two-dimensional data.

### 5.2.1 Detailed Description

Sample using artificial two-dimensional data.

Like in "Marko Tscherepanow, Sina Kühnel, and Sören Riechers, 'Episodic Clustering of Data Streams Using a Topology-Learning Neural Network', European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29, 2012." an Episodic TopoART network is trained with the well-known Two Spirals dataset. Due to the incorporation of temporal information during learning, Episodic TopoART is capable of creating two clusters each representing one spiral in an unsupervised way. These clusters are formed by the nodes of module b (ETA b).

The resulting network can be visualised using the R script `ShowEpisodicTopoARTResults.R` provided in the subfolder R.

## 5.3 LibTopoART.F2\_output Class Reference

Class `F2_output` provides the output of a single `TopoART` module. It is a compressed version of the output vectors `y` and `c`.

### Public Member Functions

- `F2_output()`

*This constructor sets all instance variables of class `F2_output` to "undefined".*

### Public Attributes

- decimal `bm_node_activation`

*Instance variable `bm_node_activation` represents the activation of the best-matching node (prediction variant).*

- long `bm_node_ID`

*Instance variable `bm_node_ID` represents the ID of the best-matching node.*

- long `bm_cluster_ID`

*Instance variable `bm_cluster_ID` represents the cluster ID of the best-matching node.*

- decimal `bm_permanent_node_activation`

*Instance variable `bm_permanent_node_activation` represents the activation of the best-matching permanent node (prediction variant).*

- long `bm_permanent_node_ID`

*Instance variable `bm_permanent_node_ID` represents the ID of the best-matching permanent node.*

- long `bm_permanent_cluster_ID`

*Instance variable `bm_permanent_cluster_ID` represents the cluster ID of the best-matching permanent node.*

### 5.3.1 Detailed Description

Class `F2_output` provides the output of a single `TopoART` module. It is a compressed version of the output vectors `y` and `c`.

### 5.3.2 Constructor & Destructor Documentation

#### 5.3.2.1 LibTopoART.F2\_output.F2\_output ( )

This constructor sets all instance variables of class `F2_output` to "undefined".

### 5.3.3 Member Data Documentation

#### 5.3.3.1 long LibTopoART.F2\_output.bm\_cluster\_ID

Instance variable `bm_cluster_ID` represents the cluster ID of the best-matching node.

#### 5.3.3.2 decimal LibTopoART.F2\_output.bm\_node\_activation

Instance variable `bm_node_activation` represents the activation of the best-matching node (prediction variant).

#### 5.3.3.3 long LibTopoART.F2\_output.bm\_node\_ID

Instance variable `bm_node_ID` represents the ID of the best-matching node.

#### 5.3.3.4 long LibTopoART.F2\_output.bm\_permanent\_cluster\_ID

Instance variable `bm_permanent_cluster_ID` represents the cluster ID of the best-matching permanent node.

#### 5.3.3.5 decimal LibTopoART.F2\_output.bm\_permanent\_node\_activation

Instance variable `bm_permanent_node_activation` represents the activation of the best-matching permanent node (prediction variant).

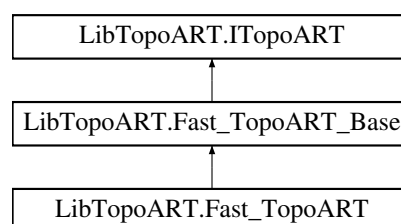
#### 5.3.3.6 long LibTopoART.F2\_output.bm\_permanent\_node\_ID

Instance variable `bm_permanent_node_ID` represents the ID of the best-matching permanent node.

## 5.4 LibTopoART.Fast\_TopoART Class Reference

Class `Fast_TopoART` provides an implementation of the `TopoART` neural network as proposed in "Tscherepanow, Marko (2010). `TopoART`: A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer."

Inheritance diagram for LibTopoART.Fast\_TopoART:



## Public Member Functions

- [Fast\\_TopoART](#) (long input\_dimension, long module\_number, decimal rho\_a\_value)

*This constructor initialises a [TopoART](#) network.*

- [Fast\\_TopoART](#) (string path)

*This constructor loads a saved [TopoART](#) network.*

- override void [Learn](#) (decimal[] input)

*This method performs a single training step.*

## Additional Inherited Members

### 5.4.1 Detailed Description

Class [Fast\\_TopoART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Tscherepanow, Marko (2010). [TopoART](#): A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer."

Internally, real-valued data are mapped to `long` variables. Therefore, computations are accelerated but less accurate. As a consequence, the results may differ slightly from class [TopoART](#).

Class [Fast\\_TopoART](#) requires all input to lie in the interval [0,1].

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 LibTopoART.Fast\_TopoART.Fast\_TopoART ( long input\_dimension, long module\_number, decimal rho\_a\_value )

This constructor initialises a [TopoART](#) network.

##### Parameters

<i>input_dimension</i>	The dimension of input vectors to be learnt.
<i>module_number</i>	The number of <a href="#">TopoART</a> modules.
<i>rho_a_value</i>	The vigilance parameter of the first <a href="#">TopoART</a> module (TA a).

#### 5.4.2.2 LibTopoART.Fast\_TopoART.Fast\_TopoART ( string path )

This constructor loads a saved [TopoART](#) network.

##### Parameters

<i>path</i>	The path of a binary <a href="#">TopoART</a> file.
-------------	--

### 5.4.3 Member Function Documentation

#### 5.4.3.1 override void LibTopoART.Fast\_TopoART.Learn ( decimal[] input ) [virtual]

This method performs a single training step.

##### Parameters

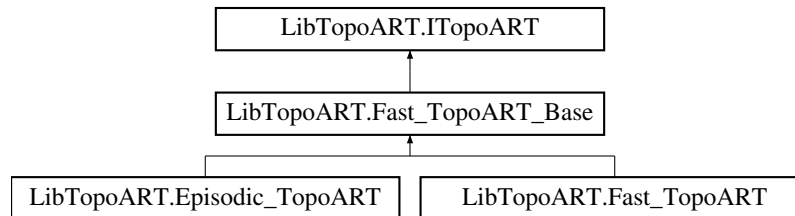
<i>input</i>	The input vector to be learnt.
--------------	--------------------------------

Implements [LibTopoART.Fast\\_TopoART\\_Base](#).

## 5.5 LibTopoART.Fast\_TopoART\_Base Class Reference

Base class providing functionality common to several [TopoART](#) networks.

Inheritance diagram for LibTopoART.Fast\_TopoART\_Base:



### Public Member Functions

- abstract void [Learn](#) (decimal[] input)  
*This method performs a single training step.*
- void [ComputeClusterIDs](#) ()  
*This method computes the cluster IDs for all neurons.*
- [F2\\_output\[\] GetBMOutput](#) (decimal[] input)  
*This method finds the closest category for a given test input.*
- void [SaveText](#) (string path)  
*This method saves the entire network as a text file.*
- void [Save](#) (string path)  
*This method saves the entire network as a binary file.*

### Static Public Attributes

- static readonly decimal [file\\_format\\_version](#) = 0.09m  
*Instance variable `file_format_version` represents the version of the file format used by class [Fast\\_TopoART](#).*
- static readonly string [integer\\_base\\_type](#) = "long"  
*Instance variable `integer_base_type` provides a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.*
- static readonly string [float\\_base\\_type](#) = "long"  
*Instance variable `float_base_type` provides a string containing the data type used for representing floating point variables (input, weights, etc.) internally.*

### Properties

- long [ModuleNum](#) [get]  
*Property `ModuleNum` represents the number of [TopoART](#) modules used. (The original [TopoART](#) uses two modules.)*
- long [LearningSteps](#) [get]  
*Property `LearningSteps` represents the total number of performed learning steps.*
- decimal [Rho\\_a](#) [get]  
*Property `Rho_a` represents the vigilance parameter of the first [TopoART](#) module (TA a).*
- decimal [Beta\\_sbm](#) [get, set]  
*Property `Beta_sbm` represents the learning rate of the second best-matching nodes.*
- long [Tau](#) [get, set]  
*Property `Tau` represents the parameter tau required for the removal of nodes and edges.*

- long [Phi](#) [get, set]  
*Property `Phi` represents the parameter `phi` required for the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules.*
- decimal [Alpha](#) [get, set]  
*Property `Alpha` represents the choice parameter `alpha`.*
- long[] [NodeNum](#) [get]  
*Property `NodeNum` represents the number of [TopoART](#) nodes used by each module.*
- long[] [ClusterNum](#) [get]  
*Property `ClusterNum` represents the number of [TopoART](#) clusters found by each module.*

### 5.5.1 Detailed Description

Base class providing functionality common to several [TopoART](#) networks.

### 5.5.2 Member Function Documentation

#### 5.5.2.1 void LibTopoART.Fast\_TopoART\_Base.ComputeClusterIDs ( )

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

#### 5.5.2.2 F2\_output [] LibTopoART.Fast\_TopoART\_Base.GetBMOutput ( decimal[] input )

This method finds the closest category for a given test input.

Parameters

<i>input</i>	The input vector.
--------------	-------------------

Returns

An array of type [F2\\_output](#). Each entry contains the ID of the best-matching node and the corresponding cluster ID for one [TopoART](#) module.

Implements [LibTopoART.ITopoART](#).

#### 5.5.2.3 abstract void LibTopoART.Fast\_TopoART\_Base.Learn ( decimal[] input ) [pure virtual]

This method performs a single training step.

Parameters

<i>input</i>	The input vector to be learnt.
--------------	--------------------------------

Implements [LibTopoART.ITopoART](#).

Implemented in [LibTopoART.Episodic\\_TopoART](#), and [LibTopoART.Fast\\_TopoART](#).

#### 5.5.2.4 void LibTopoART.Fast\_TopoART\_Base.Save ( string path )

This method saves the entire network as a binary file.

## Parameters

<i>path</i>	A string representing the save file.
-------------	--------------------------------------

Implements [LibTopoART.ITopoART](#).

#### 5.5.2.5 void LibTopoART.Fast\_TopoART\_Base.SaveText ( string path )

This method saves the entire network as a text file.

## Parameters

<i>path</i>	A string representing the save file.
-------------	--------------------------------------

Implements [LibTopoART.ITopoART](#).

### 5.5.3 Member Data Documentation

#### 5.5.3.1 readonly decimal LibTopoART.Fast\_TopoART\_Base.file\_format\_version = 0.09m [static]

Instance variable `file_format_version` represents the version of the file format used by class [Fast\\_TopoART](#).

#### 5.5.3.2 readonly string LibTopoART.Fast\_TopoART\_Base.float\_base\_type = "long" [static]

Instance variable `float_base_type` provides a string containing the data type used for representing floating point variables (input, weights, etc.) internally.

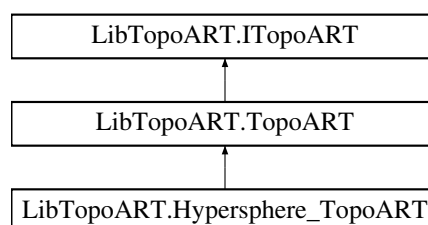
#### 5.5.3.3 readonly string LibTopoART.Fast\_TopoART\_Base.integer\_base\_type = "long" [static]

Instance variable `integer_base_type` provides a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.

## 5.6 LibTopoART.Hypersphere\_TopoART Class Reference

Class [Hypersphere\\_TopoART](#) provides an implementation of the Hypersphere [TopoART](#) neural network as proposed in "Tscherepanow, Marko (2012). Incremental On-line Clustering with a Topology-Learning Hierarchical ART Neural Network Using Hyperspherical Categories. In Poster and Industry Proceedings of the Industrial Conference on Data Mining (ICDM), pp. 22–34. Fockendorf, Germany: ibai-publishing."

Inheritance diagram for LibTopoART.Hypersphere\_TopoART:



### Public Member Functions

- [Hypersphere\\_TopoART](#) (long input\_dimension, long module\_number, decimal rho\_a\_value)

This constructor initialises a Hypersphere [TopoART](#) network and sets the radial extend parameter to  $\text{Math}.\sqrt{\text{input\_dimension}}/2$ .

- [Hypersphere\\_TopoART](#) (long input\_dimension, long module\_number, decimal rho\_a\_value, decimal R\_value)

This constructor initialises a Hypersphere [TopoART](#) network.

- [Hypersphere\\_TopoART](#) (string path)

This constructor loads a saved Hypersphere [TopoART](#) network.

## Static Public Attributes

- static new readonly decimal [file\\_format\\_version](#) = 0.01m

Instance variable `file_format_version` represents the version of the file format used by class [Hypersphere\\_TopoART](#).

## Properties

- decimal [R](#) [get]

Property `R` represents the radial extend parameter `R`.

## Additional Inherited Members

### 5.6.1 Detailed Description

Class [Hypersphere\\_TopoART](#) provides an implementation of the Hypersphere [TopoART](#) neural network as proposed in "Tscherepanow, Marko (2012). Incremental On-line Clustering with a Topology-Learning Hierarchical ART Neural Network Using Hyperspherical Categories. In Poster and Industry Proceedings of the Industrial Conference on Data Mining (ICDM), pp. 22–34. Fockendorf, Germany: ibai-publishing."

In contrast to class [TopoART](#), class [Hypersphere\\_TopoART](#) does not require all input to lie in the interval [0,1]. The input range is controlled by the radial extend parameter `R`

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 LibTopoART.Hypersphere\_TopoART.Hypersphere\_TopoART ( long input\_dimension, long module\_number, decimal rho\_a\_value )

This constructor initialises a Hypersphere [TopoART](#) network and sets the radial extend parameter to  $\text{Math}.\sqrt{\text{input\_dimension}}/2$ .

#### Parameters

<code>input_dimension</code>	The dimension of input vectors to be learnt.
<code>module_number</code>	The number of Hypersphere <a href="#">TopoART</a> modules.
<code>rho_a_value</code>	The vigilance parameter of the first Hypersphere <a href="#">TopoART</a> module (HTA a).

#### 5.6.2.2 LibTopoART.Hypersphere\_TopoART.Hypersphere\_TopoART ( long input\_dimension, long module\_number, decimal rho\_a\_value, decimal R\_value )

This constructor initialises a Hypersphere [TopoART](#) network.

## Parameters

<i>input_dimension</i>	The dimension of input vectors to be learnt.
<i>module_number</i>	The number of Hypersphere <a href="#">TopoART</a> modules.
<i>rho_a_value</i>	The vigilance parameter of the first Hypersphere <a href="#">TopoART</a> module (HTA a).
<i>R_value</i>	The radial extend parameter.

## 5.6.2.3 LibTopoART.Hypersphere\_TopoART.Hypersphere\_TopoART ( string path )

This constructor loads a saved Hypersphere [TopoART](#) network.

## Parameters

<i>path</i>	The path of a binary Hypersphere <a href="#">TopoART</a> file.
-------------	--

## 5.6.3 Member Data Documentation

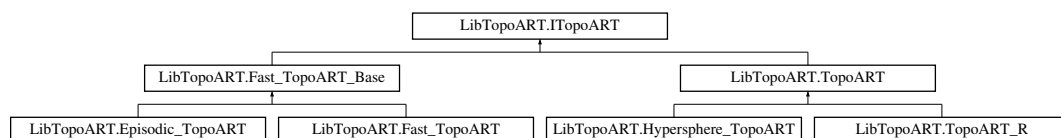
## 5.6.3.1 new readonly decimal LibTopoART.Hypersphere\_TopoART.file\_format\_version = 0.01m [static]

Instance variable `file_format_version` represents the version of the file format used by class [Hypersphere\\_TopoART](#).

## 5.7 LibTopoART.ITopoART Interface Reference

Interface summarising the basic [TopoART](#) functionality.

Inheritance diagram for LibTopoART.ITopoART:



## Public Member Functions

- void [ComputeClusterIDs](#) ()  
*This method computes the cluster IDs for all neurons.*
- [F2\\_output](#)[] [GetBMOutput](#) (decimal[] input)  
*This method finds the closest category for a given test input.*
- void [Learn](#) (decimal[] input)  
*This method performs a single training step.*
- void [SaveText](#) (string path)  
*This method saves the entire network as a text file.*
- void [Save](#) (string path)  
*This method saves the entire network as a binary file.*

## Properties

- long[] [NodeNum](#) [get]  
*Property `NodeNum` represents the number of [TopoART](#) nodes used by each module.*
- long[] [ClusterNum](#) [get]



- Property ClusterNum represents the number of [TopoART](#) clusters found by each module.*

  - long [ModuleNum](#) [get]

*Property ModuleNum represents the number of [TopoART](#) modules used. (The original [TopoART](#) uses two modules.)*
- long [LearningSteps](#) [get]

*Property LearningSteps represents the total number of performed learning steps.*
- decimal [Beta\\_sbm](#) [get, set]

*Property Beta\_sbm represents the learning rate of the second best-matching nodes.*
- decimal [Rho\\_a](#) [get]

*Property Rho\_a represents the vigilance parameter of the first [TopoART](#) module (TA a).*
- long [Tau](#) [get, set]

*Property Tau represents the parameter tau required for the removal of nodes and edges.*
- long [Phi](#) [get, set]

*Property Phi represents the parameter phi required for the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules.*
- decimal [Alpha](#) [get, set]

*Property Alpha represents the choice parameter alpha.*

### 5.7.1 Detailed Description

Interface summarising the basic [TopoART](#) functionality.

### 5.7.2 Member Function Documentation

#### 5.7.2.1 void LibTopoART.ITopoART.ComputeClusterIDs ( )

This method computes the cluster IDs for all neurons.

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast\\_TopoART\\_Base](#).

#### 5.7.2.2 F2\_output [] LibTopoART.ITopoART.GetBMOutput ( decimal[] input )

This method finds the closest category for a given test input.

Parameters

<i>input</i>	The input vector.
--------------	-------------------

Returns

An array of type [F2\\_output](#). Each entry contains the ID of the best-matching node and the corresponding cluster ID for one [TopoART](#) module.

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast\\_TopoART\\_Base](#).

#### 5.7.2.3 void LibTopoART.ITopoART.Learn ( decimal[] input )

This method performs a single training step.

Parameters

<i>input</i>	The input vector to be learnt.
--------------	--------------------------------

Implemented in [LibTopoART.TopoART](#), [LibTopoART.Episodic\\_Top ART](#), [LibTopoART.Fast\\_Top ART](#), and [LibTopoART.Fast\\_Top ART\\_Base](#).

#### 5.7.2.4 void LibTopoART.ITopoART.Save ( string path )

This method saves the entire network as a binary file.

Parameters

<i>path</i>	A string representing the save file.
-------------	--------------------------------------

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast\\_Top ART\\_Base](#).

#### 5.7.2.5 void LibTopoART.ITopoART.SaveText ( string path )

This method saves the entire network as a text file.

Parameters

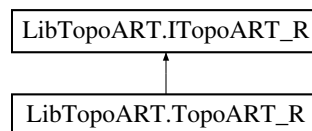
<i>path</i>	A string representing the save file.
-------------	--------------------------------------

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast\\_Top ART\\_Base](#).

## 5.8 LibTopoART.ITopoART\_R Interface Reference

Interface summarising the basic TopoART-R functionality.

Inheritance diagram for LibTopoART.ITopoART\_R:



### Public Member Functions

- void [Learn](#) (decimal[] i\_vec, decimal[] d\_vec)  
*This method performs a single training step.*
- [TopoART\\_R\\_Prediction Predict](#) (decimal[] i\_vec, bool[] m\_i\_vec)  
*This method predicts the dependent variables for a given set of independent variables. Unknown values of independent variables can be signified by setting the corresponding value of m\_i\_vec to true.*

#### 5.8.1 Detailed Description

Interface summarising the basic TopoART-R functionality.

#### 5.8.2 Member Function Documentation

##### 5.8.2.1 void LibTopoART.ITopoART\_R.Learn ( decimal[] i\_vec, decimal[] d\_vec )

This method performs a single training step.

## Parameters

<i>i_vec</i>	The input vector (independent variables) to be learnt.
<i>d_vec</i>	The output vector (dependent variables) corresponding to <i>i_vec</i> .

Implemented in [LibTopoART.TopoART\\_R](#).

### 5.8.2.2 TopoART\_R\_Prediction LibTopoART.ITopoART\_R.Predict ( decimal[] *i\_vec*, bool[] *m\_i\_vec* )

This method predicts the dependent variables for a given set of independent variables. Unknown values of independent variables can be signified by setting the corresponding value of *m\_i\_vec* to true.

## Parameters

<i>i_vec</i>	The input vector (independent variables).
<i>m_i_vec</i>	The mask vector corresponding to <i>i_vec</i> .

## Returns

An object of type [TopoART\\_R\\_Prediction](#) containing the predicted values for the unknown independent variables and all dependent variables.

Implemented in [LibTopoART.TopoART\\_R](#).

## 5.9 LibTopoART.LibTopoART\_info Struct Reference

Struct [LibTopoART\\_info](#) provides some metainformation regarding the respective implementation of [LibTopoART](#).

### Static Public Attributes

- static readonly decimal [version](#) = 0.55m  
*Instance variable `version` represents the version of [LibTopoART](#).*
- static readonly string[] [networks](#)  
*Instance variable `networks` provides a string array containing the networks implemented in the current version of [LibTopoART](#) and the corresponding class names.*

### 5.9.1 Detailed Description

Struct [LibTopoART\\_info](#) provides some metainformation regarding the respective implementation of [LibTopoART](#).

### 5.9.2 Member Data Documentation

#### 5.9.2.1 readonly string [] LibTopoART.LibTopoART\_info.networks [static]

## Initial value:

```
= {
    "Episodic TopoART (class Episodic_TopoART)",
    "Hypersphere TopoART (class Hypersphere_TopoART)",
    "TopoART (class TopoART, class Fast_TopoART)",
    "TopoART-R (class TopoART_R)" }
```

Instance variable `networks` provides a string array containing the networks implemented in the current version of [LibTopoART](#) and the corresponding class names.

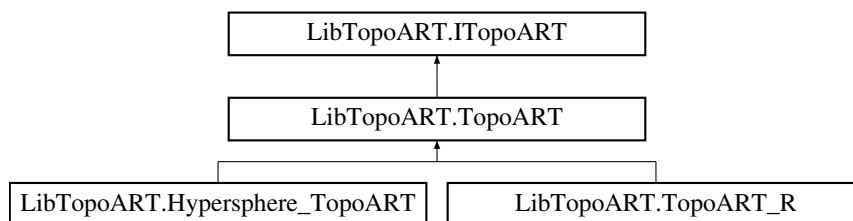
### 5.9.2.2 readonly decimal LibTopoART.LibTopoART\_info.version = 0.55m [static]

Instance variable `version` represents the version of [LibTopoART](#).

## 5.10 LibTopoART.TopoART Class Reference

Class [TopoART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Tscherepanow, Marko (2010). [TopoART](#): A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), LNCS 6354, pp. 157–167. Berlin, Germany: Springer."

Inheritance diagram for LibTopoART.TopoART:



### Public Member Functions

- [TopoART](#) (long input\_dimension, long module\_number, decimal rho\_a\_value)

*This constructor initialises a [TopoART](#) network.*

- [TopoART](#) (string path)

*This constructor loads a saved [TopoART](#) network.*

- void [ComputeClusterIDs](#) ()

*This method computes the cluster IDs for all neurons.*

- [F2\\_output\[\] GetBMOutput](#) (decimal[] input)

*This method finds the closest category for a given test input.*

- void [Learn](#) (decimal[] input)

*This method performs a single training step.*

- void [SaveText](#) (string path)

*This method saves the entire network as a text file.*

- void [Save](#) (string path)

*This method saves the entire network as a binary file.*

### Static Public Attributes

- static readonly decimal [file\\_format\\_version](#) = 0.09m

*Instance variable `file_format_version` represents the version of the file format used by class [TopoART](#).*

- static readonly string [integer\\_base\\_type](#) = "long"

*Instance variable `integer_base_type` provides a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.*

- static readonly string [float\\_base\\_type](#) = "decimal"

*Instance variable `float_base_type` provides a string containing the data type used for representing floating point variables (input, weights, etc.) internally.*

## Protected Attributes

- long `x_F0_dim`  
*Instance variable `x_F0_dim` represents the input dimension.*
- decimal[] `x_F0`  
*Instance variable `x_F0` represents the current input vector.*

## Properties

- long[] `NodeNum` [get]  
*Property `NodeNum` represents the number of [TopoART](#) nodes used by each module.*
- long[] `ClusterNum` [get]  
*Property `ClusterNum` represents the number of [TopoART](#) clusters found by each module.*
- long `ModuleNum` [get]  
*Property `ModuleNum` represents the number of [TopoART](#) modules used. (The original [TopoART](#) uses two modules.)*
- long `LearningSteps` [get]  
*Property `LearningSteps` represents the total number of performed learning steps.*
- decimal `Rho_a` [get]  
*Property `Rho_a` represents the vigilance parameter of the first [TopoART](#) module (TA a).*
- decimal `Beta_sbm` [get, set]  
*Property `Beta_sbm` represents the learning rate of the second best-matching nodes.*
- long `Tau` [get, set]  
*Property `Tau` represents the parameter tau required for the removal of nodes and edges.*
- long `Phi` [get, set]  
*Property `Phi` represents the parameter phi required for the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules.*
- decimal `Alpha` [get, set]  
*Property `Alpha` represents the choice parameter alpha.*

### 5.10.1 Detailed Description

Class [TopoART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Tscherepanow, Marko (2010). [TopoART](#): A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), LNCS 6354, pp. 157–167. Berlin, Germany: Springer."

Internally, real-valued data are stored in `decimal` variables. Hence, computations are rather slow but very accurate.

Class [TopoART](#) requires all input to lie in the interval [0,1].

### 5.10.2 Constructor & Destructor Documentation

#### 5.10.2.1 LibTopoART.TopoART.TopoART ( long `input_dimension`, long `module_number`, decimal `rho_a_value` )

This constructor initialises a [TopoART](#) network.

#### Parameters

<code>input_dimension</code>	The dimension of input vectors to be learnt.
<code>module_number</code>	The number of <a href="#">TopoART</a> modules.

<i>rho_a_value</i>	The vigilance parameter of the first <a href="#">TopoART</a> module (TA a).
--------------------	---

#### 5.10.2.2 LibTopoART.TopoART.TopoART ( string path )

This constructor loads a saved [TopoART](#) network.

Parameters

<i>path</i>	The path of a binary <a href="#">TopoART</a> file.
-------------	--

### 5.10.3 Member Function Documentation

#### 5.10.3.1 void LibTopoART.TopoART.ComputeClusterIDs ( )

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

#### 5.10.3.2 F2\_output [] LibTopoART.TopoART.GetBMOutput ( decimal[] input )

This method finds the closest category for a given test input.

Parameters

<i>input</i>	The input vector.
--------------	-------------------

Returns

An array of type [F2\\_output](#). Each entry contains the ID of the best-matching node and the corresponding cluster ID for one [TopoART](#) module.

Implements [LibTopoART.ITopoART](#).

#### 5.10.3.3 void LibTopoART.TopoART.Learn ( decimal[] input )

This method performs a single training step.

Parameters

<i>input</i>	The input vector to be learnt.
--------------	--------------------------------

Implements [LibTopoART.ITopoART](#).

#### 5.10.3.4 void LibTopoART.TopoART.Save ( string path )

This method saves the entire network as a binary file.

Parameters

<i>path</i>	A <i>string</i> representing the save file.
-------------	---

Implements [LibTopoART.ITopoART](#).

#### 5.10.3.5 void LibTopoART.TopoART.SaveText ( string path )

This method saves the entire network as a text file.

## Parameters

<i>path</i>	A string representing the save file.
-------------	--------------------------------------

Implements [LibTopoART.ITopoART](#).

### 5.10.4 Member Data Documentation

5.10.4.1 readonly decimal `LibTopoART.TopoART.file_format_version = 0.09m` `[static]`

Instance variable `file_format_version` represents the version of the file format used by class [TopoART](#).

5.10.4.2 readonly string `LibTopoART.TopoART.float_base_type = "decimal"` `[static]`

Instance variable `float_base_type` provides a string containing the data type used for representing floating point variables (input, weights, etc.) internally.

5.10.4.3 readonly string `LibTopoART.TopoART.integer_base_type = "long"` `[static]`

Instance variable `integer_base_type` provides a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.

5.10.4.4 decimal [] `LibTopoART.TopoART.x_F0` `[protected]`

Instance variable `x_F0` represents the current input vector.

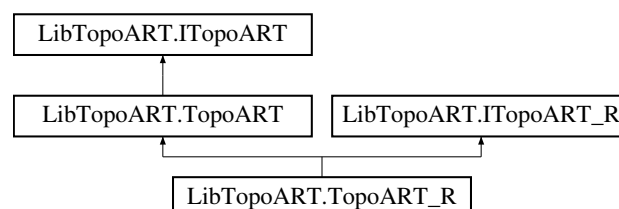
5.10.4.5 long `LibTopoART.TopoART.x_F0_dim` `[protected]`

Instance variable `x_F0_dim` represents the input dimension.

## 5.11 LibTopoART.TopoART\_R Class Reference

Class [TopoART\\_R](#) provides an implementation of the TopoART-R neural network as proposed in "Tscherepanow, Marko (2011). An Extended [TopoART](#) Network for the Stable On-Line Learning of Regression Functions. In Proceedings of the International Conference on Neural Information Processing (ICONIP), LNCS 7063, pp. 562–571. Berlin, Germany: Springer.

Inheritance diagram for `LibTopoART.TopoART_R`:



### Public Member Functions

- [TopoART\\_R](#) (long `i_dimension`, long `d_dimension`, long `module_number`, decimal `rho_a_value`)  
*This constructor initialises a TopoART-R network.*

- [TopoART\\_R](#) (string path)  
*This constructor loads a saved TopoART-R network.*
- void [Learn](#) (decimal[] i\_vec, decimal[] d\_vec)  
*This method performs a single training step.*
- [TopoART\\_R\\_Prediction Predict](#) (decimal[] i\_vec, bool[] m\_i\_vec)  
*This method predicts the dependent variables for a given set of independent variables. Unknown values of independent variables can be signified by setting the corresponding value of m\_i\_vec to true.*

## Static Public Attributes

- static new readonly decimal [file\\_format\\_version](#) = 0.01m  
*Instance variable file\_format\_version represents the version of the file format used by class [TopoART\\_R](#).*

## Additional Inherited Members

### 5.11.1 Detailed Description

Class [TopoART\\_R](#) provides an implementation of the TopoART-R neural network as proposed in "Tscherepanow, Marko (2011). An Extended [TopoART](#) Network for the Stable On-Line Learning of Regression Functions. In Proceedings of the International Conference on Neural Information Processing (ICONIP), LNCS 7063, pp. 562–571. Berlin, Germany: Springer.

Class [TopoART\\_R](#) requires all input and output to lie in the interval [0,1].

### 5.11.2 Constructor & Destructor Documentation

#### 5.11.2.1 LibTopoART.TopoART\_R.TopoART\_R ( long i\_dimension, long d\_dimension, long module\_number, decimal rho\_a\_value )

This constructor initialises a TopoART-R network.

##### Parameters

<i>i_dimension</i>	The dimension of the input vector (independent variables) to be learnt.
<i>d_dimension</i>	The dimension of the output vector (dependent variables) to be learnt.
<i>module_number</i>	The number of TopoART-R modules.
<i>rho_a_value</i>	The vigilance parameter of the first TopoART-R module (TopoART-R a).

#### 5.11.2.2 LibTopoART.TopoART\_R.TopoART\_R ( string path )

This constructor loads a saved TopoART-R network.

##### Parameters

<i>path</i>	The path of a binary TopoART-R file.
-------------	--------------------------------------

### 5.11.3 Member Function Documentation

#### 5.11.3.1 void LibTopoART.TopoART\_R.Learn ( decimal[] i\_vec, decimal[] d\_vec )

This method performs a single training step.



## Parameters

<i>i_vec</i>	The input vector (independent variables) to be learnt.
<i>d_vec</i>	The output vector (dependent variables) corresponding to <i>i_vec</i> .

Implements [LibTopoART.ITopoART\\_R](#).

### 5.11.3.2 TopoART\_R\_Prediction LibTopoART.TopoART\_R.Predict ( decimal[] i\_vec, bool[] m\_i\_vec )

This method predicts the dependent variables for a given set of independent variables. Unknown values of independent variables can be signified by setting the corresponding value of *m\_i\_vec* to true.

## Parameters

<i>i_vec</i>	The input vector (independent variables).
<i>m_i_vec</i>	The mask vector corresponding to <i>i_vec</i> .

## Returns

An object of type [TopoART\\_R\\_Prediction](#) containing the predicted values for the unknown independent variables and all dependent variables.

Implements [LibTopoART.ITopoART\\_R](#).

## 5.11.4 Member Data Documentation

### 5.11.4.1 new readonly decimal LibTopoART.TopoART\_R.file\_format\_version = 0.01m [static]

Instance variable *file\_format\_version* represents the version of the file format used by class [TopoART\\_R](#).

## 5.12 LibTopoART.TopoART\_R\_Prediction Class Reference

Class [F2\\_output](#) contains a prediction made by a TopoART-R network.

### Public Member Functions

- [TopoART\\_R\\_Prediction](#) (decimal[] *i\_vec\_prediction*, decimal[] *d\_vec\_prediction*)  
*This constructor sets the instance variables *i\_vec\_prediction* and *d\_vec\_prediction* of class [TopoART\\_R\\_Prediction](#).*
- void [PrintPredictions](#) ()  
*This method prints the predictions on the console.*

### Public Attributes

- readonly decimal[] [i\\_vec\\_prediction](#)  
*Instance variable *i\_vec\_prediction* represents predictions for unknown independent variables.*
- readonly decimal[] [d\\_vec\\_prediction](#)  
*Instance variable *d\_vec\_prediction* provides the predictions for the dependent variables.*

### Static Public Attributes

- static readonly decimal [NO\\_PREDICTION](#) = -1.0m  
*Instance variable *NO\_PREDICTION* provides a default prediction to signify variables that are presented to the network; i.e., these variables are known and no prediction is computed for them.*

### 5.12.1 Detailed Description

Class `F2_output` contains a prediction made by a TopoART-R network.

### 5.12.2 Constructor & Destructor Documentation

5.12.2.1 `LibTopoART.TopoART_R_Prediction.TopoART_R_Prediction ( decimal[] i_vec_prediction, decimal[] d_vec_prediction )`

This constructor sets the instance variables `i_vec_prediction` and `d_vec_prediction` of class `TopoART_R_Prediction`.

### 5.12.3 Member Function Documentation

5.12.3.1 `void LibTopoART.TopoART_R_Prediction.PrintPredictions ( )`

This method prints the predictions on the console.

### 5.12.4 Member Data Documentation

5.12.4.1 `readonly decimal [] LibTopoART.TopoART_R_Prediction.d_vec_prediction`

Instance variable `d_vec_prediction` provides the predictions for the dependent variables.

5.12.4.2 `readonly decimal [] LibTopoART.TopoART_R_Prediction.i_vec_prediction`

Instance variable `i_vec_prediction` represents predictions for unknown independent variables.

5.12.4.3 `readonly decimal LibTopoART.TopoART_R_Prediction.NO_PREDICTION = -1.0m [static]`

Instance variable `NO_PREDICTION` provides a default prediction to signify variables that are presented to the network; i.e., these variables are known and no prediction is computed for them.

## 5.13 LibTopoART\_samples.TopoART\_R\_sample1 Class Reference

Function approximation using TopoART-R.

### 5.13.1 Detailed Description

Function approximation using TopoART-R.

This sample trains a TopoART-R network with 100 points sampled from a sine function. Then, sine values are predicted for 25 random values.

## 5.14 LibTopoART\_samples.TopoART\_sample1 Class Reference

Simple TopoART sample.

### 5.14.1 Detailed Description

Simple TopoART sample.

First, a dataset comprised of 10 samples is learned by a TopoART network. Afterwards, the training samples are slightly modified by random values and used for predicting cluster labels.

## 5.15 LibTopoART\_samples.TopoART\_sample2 Class Reference

Sample using artificial two-dimensional data.

### 5.15.1 Detailed Description

Sample using artificial two-dimensional data.

Train TopoART or Hypersphere TopoART with a two-dimensional dataset similar to the one used in "Marko Tscherepanow and Sören Riechers, 'An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data', European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23, 2012." This dataset comprises six clusters (each containing 15,000 samples) as well as 10,000 noise samples. These samples were mixed randomly.

The resulting neural network can be visualised using the R script `ShowTopoARTResults.R` or the R script `ShowHypersphereTopoARTResults.R`, respectively. Both R scripts are provided in the subfolder `R`.