

LibTopoART

v0.64

Generated by Doxygen 1.8.11



# 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	LibTopoART Namespace Reference	7
4.2	LibTopoART_samples Namespace Reference	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(long input_dimension, long module_number, decimal rho_↔ a_value, long t_max_value)	10
5.1.2.2	Episodic_TopoART(string path)	10
5.1.3	Member Function Documentation	10
5.1.3.1	BeginRecall(decimal[] stimulus)	10
5.1.3.2	EndRecall()	11
5.1.3.3	InterEpisodeRecallStep(out decimal[] recall_result, out decimal F3_activation)	11
5.1.3.4	IntraEpisodeRecallStep(out decimal[] recall_result)	11
5.1.3.5	Learn(decimal[] input)	12

5.1.4	Member Data Documentation . . . . .	12
5.1.4.1	file_format_version . . . . .	12
5.2	LibTopoART_samples.Episodic_TopoART_sample1 Class Reference . . . . .	12
5.2.1	Detailed Description . . . . .	12
5.3	LibTopoART_samples.Episodic_TopoART_sample2 Class Reference . . . . .	12
5.3.1	Detailed Description . . . . .	13
5.4	LibTopoART.F2_output Class Reference . . . . .	13
5.4.1	Detailed Description . . . . .	13
5.4.2	Constructor & Destructor Documentation . . . . .	13
5.4.2.1	F2_output() . . . . .	13
5.4.3	Member Data Documentation . . . . .	14
5.4.3.1	bm_cluster_ID . . . . .	14
5.4.3.2	bm_node_activation . . . . .	14
5.4.3.3	bm_node_ID . . . . .	14
5.4.3.4	bm_permanent_cluster_ID . . . . .	14
5.4.3.5	bm_permanent_node_activation . . . . .	14
5.4.3.6	bm_permanent_node_ID . . . . .	14
5.5	LibTopoART.Fast_TopoART Class Reference . . . . .	14
5.5.1	Detailed Description . . . . .	15
5.5.2	Constructor & Destructor Documentation . . . . .	15
5.5.2.1	Fast_TopoART(long input_dimension, long module_number, decimal rho_a_value) . . . . .	15
5.5.2.2	Fast_TopoART(string path) . . . . .	15
5.5.3	Member Function Documentation . . . . .	15
5.5.3.1	Learn(decimal[] input) . . . . .	16
5.6	LibTopoART.Fast_TopoART_Base Class Reference . . . . .	16
5.6.1	Detailed Description . . . . .	17
5.6.2	Member Function Documentation . . . . .	17
5.6.2.1	ComputeClusterIDs() . . . . .	17
5.6.2.2	GetBMOutput(decimal[] input) . . . . .	17
5.6.2.3	Learn(decimal[] input) . . . . .	18

5.6.2.4	Save(string path)	18
5.6.2.5	SaveText(string path)	18
5.6.3	Member Data Documentation	18
5.6.3.1	file_format_version	18
5.6.3.2	float_base_type	18
5.6.3.3	integer_base_type	19
5.6.4	Property Documentation	19
5.6.4.1	ModuleNum	19
5.6.4.2	Phi	19
5.7	LibTopoART.Hypersphere_TopoART Class Reference	19
5.7.1	Detailed Description	20
5.7.2	Constructor & Destructor Documentation	20
5.7.2.1	Hypersphere_TopoART(long input_dimension, long module_number, decimal rho_a_value)	20
5.7.2.2	Hypersphere_TopoART(long input_dimension, long module_number, decimal rho_a_value, decimal R_value)	20
5.7.2.3	Hypersphere_TopoART(string path)	21
5.7.3	Member Data Documentation	21
5.7.3.1	file_format_version	21
5.8	LibTopoART.IEpisodic_Recall Interface Reference	21
5.8.1	Detailed Description	21
5.8.2	Member Function Documentation	21
5.8.2.1	BeginRecall(decimal[] stimulus)	21
5.8.2.2	EndRecall()	22
5.8.2.3	InterEpisodeRecallStep(out decimal[] recall_result, out decimal F3_activation)	22
5.8.2.4	IntraEpisodeRecallStep(out decimal[] recall_result)	22
5.9	LibTopoART.ITopoART Interface Reference	23
5.9.1	Detailed Description	23
5.9.2	Member Function Documentation	24
5.9.2.1	ComputeClusterIDs()	24
5.9.2.2	GetBMOutput(decimal[] input)	24

5.9.2.3	Learn(decimal[] input)	24
5.9.2.4	Save(string path)	24
5.9.2.5	SaveText(string path)	25
5.9.3	Property Documentation	25
5.9.3.1	ModuleNum	25
5.9.3.2	Phi	25
5.10	LibTopoART.ITopoART_R Interface Reference	25
5.10.1	Detailed Description	25
5.10.2	Member Function Documentation	25
5.10.2.1	Learn(decimal[] i_vec, decimal[] d_vec)	25
5.10.2.2	Predict(decimal[] i_vec, bool[] m_i_vec)	26
5.11	LibTopoART.LibTopoART_info Struct Reference	26
5.11.1	Detailed Description	26
5.11.2	Member Data Documentation	27
5.11.2.1	networks	27
5.11.2.2	UNDEFINED	27
5.11.2.3	version	27
5.12	LibTopoART.TopoART Class Reference	27
5.12.1	Detailed Description	28
5.12.2	Constructor & Destructor Documentation	28
5.12.2.1	TopoART(long input_dimension, long module_number, decimal rho_a_value)	28
5.12.2.2	TopoART(string path)	29
5.12.3	Member Function Documentation	29
5.12.3.1	ComputeClusterIDs()	29
5.12.3.2	GetBMOutput(decimal[] input)	29
5.12.3.3	Learn(decimal[] input)	29
5.12.3.4	Save(string path)	30
5.12.3.5	SaveText(string path)	30
5.12.4	Member Data Documentation	30
5.12.4.1	file_format_version	30

5.12.4.2	<a href="#">float_base_type</a>	30
5.12.4.3	<a href="#">integer_base_type</a>	30
5.12.4.4	<a href="#">x_F0</a>	30
5.12.4.5	<a href="#">x_F0_dim</a>	31
5.12.5	<a href="#">Property Documentation</a>	31
5.12.5.1	<a href="#">ModuleNum</a>	31
5.12.5.2	<a href="#">Phi</a>	31
5.13	<a href="#">LibTopoART.TopoART_R Class Reference</a>	31
5.13.1	<a href="#">Detailed Description</a>	32
5.13.2	<a href="#">Constructor &amp; Destructor Documentation</a>	32
5.13.2.1	<a href="#">TopoART_R(long i_dimension, long d_dimension, long module_number, decimal rho_a_value)</a>	32
5.13.2.2	<a href="#">TopoART_R(string path)</a>	32
5.13.3	<a href="#">Member Function Documentation</a>	32
5.13.3.1	<a href="#">Learn(decimal[] i_vec, decimal[] d_vec)</a>	32
5.13.3.2	<a href="#">Predict(decimal[] i_vec, bool[] m_i_vec)</a>	33
5.13.4	<a href="#">Member Data Documentation</a>	33
5.13.4.1	<a href="#">file_format_version</a>	33
5.14	<a href="#">LibTopoART.TopoART_R_Prediction Class Reference</a>	33
5.14.1	<a href="#">Detailed Description</a>	34
5.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	34
5.14.2.1	<a href="#">TopoART_R_Prediction(decimal[] i_vec_prediction, decimal[] d_vec_prediction)</a>	34
5.14.3	<a href="#">Member Function Documentation</a>	34
5.14.3.1	<a href="#">PrintPredictions()</a>	34
5.14.4	<a href="#">Member Data Documentation</a>	34
5.14.4.1	<a href="#">d_vec_prediction</a>	34
5.14.4.2	<a href="#">i_vec_prediction</a>	34
5.14.4.3	<a href="#">NO_PREDICTION</a>	34
5.15	<a href="#">LibTopoART_samples.TopoART_R_sample1 Class Reference</a>	35
5.15.1	<a href="#">Detailed Description</a>	35
5.16	<a href="#">LibTopoART_samples.TopoART_sample1 Class Reference</a>	35
5.16.1	<a href="#">Detailed Description</a>	35
5.17	<a href="#">LibTopoART_samples.TopoART_sample2 Class Reference</a>	35
5.17.1	<a href="#">Detailed Description</a>	35





# Chapter 1

## Namespace Index

### 1.1 Packages

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

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



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

LibTopoART_samples.Episodic_TopoART_sample1 . . . . .	12
LibTopoART_samples.Episodic_TopoART_sample2 . . . . .	12
LibTopoART.F2_output . . . . .	13
LibTopoART.IEpisodic_Recall . . . . .	21
LibTopoART.Episodic_TopoART . . . . .	9
LibTopoART.ITopoART . . . . .	23
LibTopoART.Fast_TopoART_Base . . . . .	16
LibTopoART.Episodic_TopoART . . . . .	9
LibTopoART.Fast_TopoART . . . . .	14
LibTopoART.TopoART . . . . .	27
LibTopoART.Hypersphere_TopoART . . . . .	19
LibTopoART.TopoART_R . . . . .	31
LibTopoART.ITopoART_R . . . . .	25
LibTopoART.TopoART_R . . . . .	31
LibTopoART.LibTopoART_info . . . . .	26
LibTopoART.TopoART_R_Prediction . . . . .	33
LibTopoART_samples.TopoART_R_sample1 . . . . .	35
LibTopoART_samples.TopoART_sample1 . . . . .	35
LibTopoART_samples.TopoART_sample2 . . . . .	35



## 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 synthetic two-dimensional data	12
<a href="#">LibTopoART_samples.Episodic_TopoART_sample2</a>	
Sample using real-world video data	12
<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.	13
<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."	14
<a href="#">LibTopoART.Fast_TopoART_Base</a>	
Base class providing functionality common to several <a href="#">TopoART</a> networks.	16
<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."	19
<a href="#">LibTopoART.IEpisodic_Recall</a>	
Interface summarising the episodic recall functionality.	21
<a href="#">LibTopoART.ITopoART</a>	
Interface summarising the basic <a href="#">TopoART</a> functionality.	23
<a href="#">LibTopoART.ITopoART_R</a>	
Interface summarising the basic TopoART-R functionality.	25
<a href="#">LibTopoART.LibTopoART_info</a>	
Struct <a href="#">LibTopoART_info</a> provides some metainformation regarding the respective implementation of <a href="#">LibTopoART</a> .	26

[LibTopoART.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." . . . . . 27

[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 . . . . . 31

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

Class [F2\\_output](#) contains a prediction made by a TopoART-R network. . . . . 33

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

Regression using TopoART-R . . . . . 35

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

Simple TopoART sample . . . . . 35

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

Sample using artificial two-dimensional data . . . . . 35

## Chapter 4

# Namespace Documentation

### 4.1 LibTopoART Namespace Reference

#### Classes

- class [Episodic\\_TopoART](#)  
*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."*
- 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\\_TopoART](#)  
*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."*
- class [Fast\\_TopoART\\_Base](#)  
*Base class providing functionality common to several [TopoART](#) networks.*
- class [Hypersphere\\_TopoART](#)  
*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."*
- interface [IEpisodic\\_Recall](#)  
*Interface summarising the episodic recall functionality.*
- 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 metainformation 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 LibTopoART\_samples Namespace Reference

### Classes

- class [Episodic\\_TopoART\\_sample1](#)  
*Sample using synthetic two-dimensional data.*
- class [Episodic\\_TopoART\\_sample2](#)  
*Sample using real-world video data.*
- class [TopoART\\_R\\_sample1](#)  
*Regression using TopoART-R.*
- class [TopoART\\_sample1](#)  
*Simple TopoART sample.*
- class [TopoART\\_sample2](#)  
*Sample using artificial two-dimensional data.*



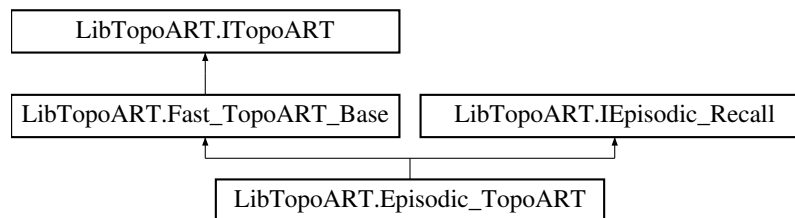
## Chapter 5

# 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.*
- long [BeginRecall](#) (decimal[] stimulus)  
*This method starts the recall process.*
- bool [InterEpisodeRecallStep](#) (out decimal[] recall\_result, out decimal F3\_activation)  
*This method performs a single inter-episode recall step and sets the starting point for intra-episode recall.*
- bool [IntraEpisodeRecallStep](#) (out decimal[] recall\_result)  
*This method performs a single intra-episode recall step.*
- void [EndRecall](#) ()  
*This method stops the recall process and frees temporary resources.*

## Public Attributes

- new const 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 <code>TopoART</code> modules.
<i>rho_a_value</i>	The vigilance parameter of the first Episodic <code>TopoART</code> 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 <code>TopoART</code> file.
-------------	--

### 5.1.3 Member Function Documentation

#### 5.1.3.1 long LibTopoART.Episodic\_TopoART.BeginRecall ( decimal[] *stimulus* )

This method starts the recall process.

**Parameters**

<i>stimulus</i>	The stimulus (input) which is used to trigger recall.
-----------------	---

**Returns**

The number of F3 nodes created.

Implements [LibTopoART.IEpisodic\\_Recall](#).

**5.1.3.2 void LibTopoART.Episodic\_TopoART.EndRecall ( )**

This method stops the recall process and frees temporary resources.

Implements [LibTopoART.IEpisodic\\_Recall](#).

**5.1.3.3 bool LibTopoART.Episodic\_TopoART.InterEpisodeRecallStep ( out decimal[] *recall\_result*, out decimal *F3\_activation* )**

This method performs a single inter-episode recall step and sets the starting point for intra-episode recall.

**Parameters**

<i>recall_result</i>	Returns the recall output vector for the current step.
<i>F3_activation</i>	Returns the activation of the current F3 node.

**Returns**

A boolean result indicating whether the recall step was successfully completed, or not.

Implements [LibTopoART.IEpisodic\\_Recall](#).

**5.1.3.4 bool LibTopoART.Episodic\_TopoART.IntraEpisodeRecallStep ( out decimal[] *recall\_result* )**

This method performs a single intra-episode recall step.

**Parameters**

<i>recall_result</i>	Returns the recall output vector for the current step.
----------------------	--

**Returns**

A boolean result indicating whether the recall step was successfully completed, or not.

Implements [LibTopoART.IEpisodic\\_Recall](#).

### 5.1.3.5 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  $w_{\{j, 2\}^{F2, t}}(t+1)$  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 const decimal LibTopoART.Episodic\_TopoART.file\_format\_version = 0.01m

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 synthetic two-dimensional data.

### 5.2.1 Detailed Description

Sample using synthetic two-dimensional data.

Like in Section 4.1 of "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\_samples.Episodic\_TopoART\_sample2 Class Reference

Sample using real-world video data.

### 5.3.1 Detailed Description

Sample using real-world video data.

Like in Section 4.2 of "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 real-world video data. Each image has a size of 64x36 pixels. As each pixel comprises 3 color channels (RGB), the input dimension equals 6912. After finishing training, recall is performed for a single input stimulus.

The recall results can be visualised using the R script `ShowEpisodicTopoARTRecallResults.R` provided in the subfolder R.

## 5.4 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 `LibTopoART_info.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.*

### 5.4.1 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`.

### 5.4.2 Constructor & Destructor Documentation

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

This constructor sets all instance variables of class `F2_output` to `LibTopoART_info.UNDEFINED`.

### 5.4.3 Member Data Documentation

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

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

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

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

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

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

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

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

#### 5.4.3.5 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).

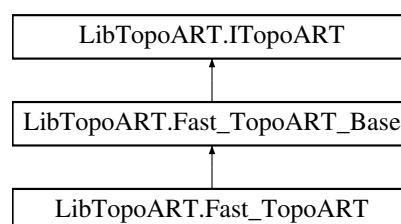
#### 5.4.3.6 long LibTopoART.F2\_output.bm\_permanent\_node\_ID

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

## 5.5 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

### 5.5.1 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].

### 5.5.2 Constructor & Destructor Documentation

#### 5.5.2.1 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).

#### 5.5.2.2 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.
-------------	--

### 5.5.3 Member Function Documentation

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

This method performs a single training step.

#### Parameters

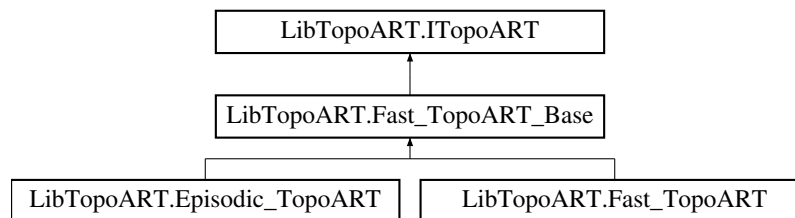
<i>input</i>	The input vector to be learnt.
--------------	--------------------------------

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

## 5.6 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.*

### Public Attributes

- const decimal [file\\_format\\_version](#) = 0.09m  
*Instance variable `file_format_version` represents the version of the file format used by class [Fast\\_TopoART](#).*
- const 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.*
- const 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]
- 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]
- 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.*

### 5.6.1 Detailed Description

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

### 5.6.2 Member Function Documentation

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

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

#### 5.6.2.2 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](#).

**5.6.2.3** `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](#).

**5.6.2.4** `void LibTopoART.Fast_TopoART_Base.Save ( string path )`

This method saves the entire network as a binary file.

#### Parameters

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

Implements [LibTopoART.ITopoART](#).

**5.6.2.5** `void LibTopoART.Fast_TopoART_Base.SaveText ( string path )`

This method saves the entire network as a text file.

#### Parameters

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

Implements [LibTopoART.ITopoART](#).

### 5.6.3 Member Data Documentation

**5.6.3.1** `const decimal LibTopoART.Fast_TopoART_Base.file_format_version = 0.09m`

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

**5.6.3.2** `const string LibTopoART.Fast_TopoART_Base.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.

### 5.6.3.3 `const string LibTopoART.Fast_TopoART_Base.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.

## 5.6.4 Property Documentation

### 5.6.4.1 `long LibTopoART.Fast_TopoART_Base.ModuleNum` `[get]`

Property `ModuleNum` represents the number of [TopoART](#) modules used. (The original [TopoART](#) uses two modules.)

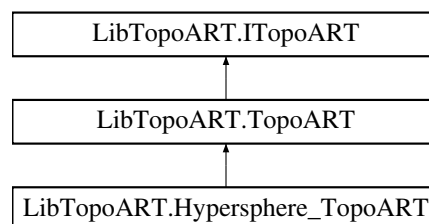
### 5.6.4.2 `long LibTopoART.Fast_TopoART_Base.Phi` `[get]`, `[set]`

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.

## 5.7 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.*

## Public Attributes

- new const 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

### 5.7.1 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`

### 5.7.2 Constructor & Destructor Documentation

#### 5.7.2.1 `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 `Math.Sqrt(input_dimension)/2`.

##### Parameters

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

#### 5.7.2.2 `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

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

### 5.7.2.3 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.
-------------	--

## 5.7.3 Member Data Documentation

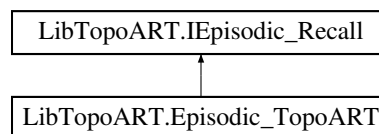
### 5.7.3.1 new const decimal LibTopoART.Hypersphere\_TopoART.file\_format\_version = 0.01m

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

## 5.8 LibTopoART.IEpisodic\_Recall Interface Reference

Interface summarising the episodic recall functionality.

Inheritance diagram for LibTopoART.IEpisodic\_Recall:



### Public Member Functions

- long [BeginRecall](#) (decimal[ ] stimulus)  
*This method starts the recall process.*
- bool [InterEpisodeRecallStep](#) (out decimal[ ] recall\_result, out decimal F3\_activation)  
*This method performs a single inter-episode recall step and sets the starting point for intra-episode recall.*
- bool [IntraEpisodeRecallStep](#) (out decimal[ ] recall\_result)  
*This method performs a single intra-episode recall step.*
- void [EndRecall](#) ()  
*This method stops the recall process and frees temporary resources.*

### 5.8.1 Detailed Description

Interface summarising the episodic recall functionality.

### 5.8.2 Member Function Documentation

#### 5.8.2.1 long LibTopoART.IEpisodic\_Recall.BeginRecall ( decimal[ ] stimulus )

This method starts the recall process.

**Parameters**

<i>stimulus</i>	The stimulus (input) which is used to trigger recall.
-----------------	---

**Returns**

The number of F3 nodes created.

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

**5.8.2.2 void LibTopoART.IEpisodic\_Recall.EndRecall ( )**

This method stops the recall process and frees temporary resources.

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

**5.8.2.3 bool LibTopoART.IEpisodic\_Recall.InterEpisodeRecallStep ( out decimal[] *recall\_result*, out decimal *F3\_activation* )**

This method performs a single inter-episode recall step and sets the starting point for intra-episode recall.

**Parameters**

<i>recall_result</i>	Returns the recall output vector for the current step.
<i>F3_activation</i>	Returns the activation of the current F3 node.

**Returns**

A boolean result indicating whether the recall step was successfully completed, or not.

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

**5.8.2.4 bool LibTopoART.IEpisodic\_Recall.IntraEpisodeRecallStep ( out decimal[] *recall\_result* )**

This method performs a single intra-episode recall step.

**Parameters**

<i>recall_result</i>	Returns the recall output vector for the current step.
----------------------	--

**Returns**

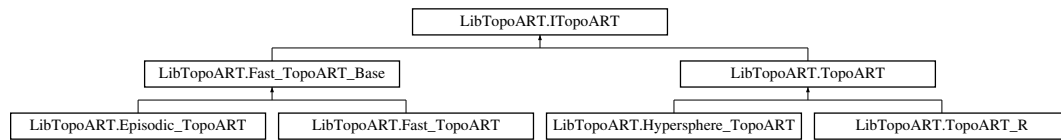
A boolean result indicating whether the recall step was successfully completed, or not.

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

## 5.9 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]
- 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]
- decimal [Alpha](#) [get, set]  
*Property Alpha represents the choice parameter alpha.*

#### 5.9.1 Detailed Description

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

## 5.9.2 Member Function Documentation

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

This method computes the cluster IDs for all neurons.

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

### 5.9.2.2 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](#).

### 5.9.2.3 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\\_TopoART](#), [LibTopoART.Fast\\_TopoART](#), and [LibTopoART.Fast\\_TopoART\\_Base](#).

### 5.9.2.4 void LibTopoART.ITopoART.Save ( string *path* )

This method saves the entire network as a binary file.

#### Parameters

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

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



## 5.9.2.5 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\\_TopoART\\_Base](#).

## 5.9.3 Property Documentation

## 5.9.3.1 long LibTopoART.ITopoART.ModuleNum [get]

Property `ModuleNum` represents the number of [TopoART](#) modules used. (The original [TopoART](#) uses two modules.)

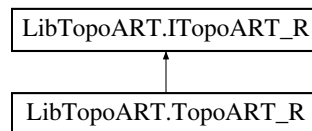
## 5.9.3.2 long LibTopoART.ITopoART.Phi [get], [set]

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.

## 5.10 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`.*

## 5.10.1 Detailed Description

Interface summarising the basic TopoART-R functionality.

## 5.10.2 Member Function Documentation

## 5.10.2.1 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](#).

#### 5.10.2.2 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](#).

## 5.11 LibTopoART.LibTopoART\_info Struct Reference

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

## Public Attributes

- const decimal [version](#) = 0.64m  
*Instance variable `version` represents the version of [LibTopoART](#).*
- const long [UNDEFINED](#) = -1  
*Instance variable `UNDEFINED` gives the value used for indicating undefined and uninitialised variables.*

## Static Public Attributes

- 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.*

### 5.11.1 Detailed Description

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

### 5.11.2 Member Data Documentation

#### 5.11.2.1 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.

#### 5.11.2.2 const long LibTopoART.LibTopoART\_info.UNDEFINED = -1

Instance variable `UNDEFINED` gives the value used for indicating undefined and uninitialised variables.

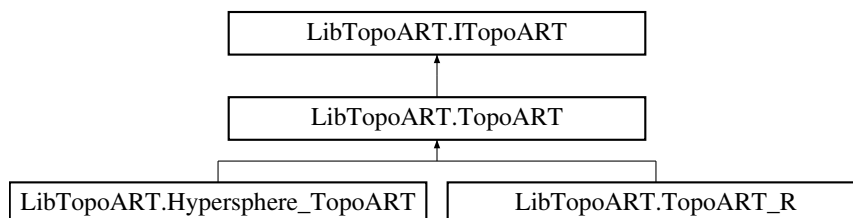
#### 5.11.2.3 const decimal LibTopoART.LibTopoART\_info.version = 0.64m

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

## 5.12 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.*

## Public Attributes

- const decimal `file_format_version` = 0.09m  
*Instance variable `file_format_version` represents the version of the file format used by class `TopoART`.*
- const 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.*
- const 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]
- 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]
- decimal `Alpha` [get, set]  
*Property `Alpha` represents the choice parameter alpha.*

### 5.12.1 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].

### 5.12.2 Constructor & Destructor Documentation

#### 5.12.2.1 `LibTopoART.TopoART.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).

## 5.12.2.2 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.
-------------	--

## 5.12.3 Member Function Documentation

## 5.12.3.1 void LibTopoART.TopoART.ComputeClusterIDs ( )

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

## 5.12.3.2 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](#).

## 5.12.3.3 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](#).

#### 5.12.3.4 void LibTopoART.TopoART.Save ( string *path* )

This method saves the entire network as a binary file.

##### Parameters

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

Implements [LibTopoART.ITopoART](#).

#### 5.12.3.5 void LibTopoART.TopoART.SaveText ( string *path* )

This method saves the entire network as a text file.

##### Parameters

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

Implements [LibTopoART.ITopoART](#).

### 5.12.4 Member Data Documentation

#### 5.12.4.1 const decimal LibTopoART.TopoART.file\_format\_version = 0.09m

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

#### 5.12.4.2 const string LibTopoART.TopoART.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.

#### 5.12.4.3 const string LibTopoART.TopoART.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.

#### 5.12.4.4 decimal [ ] LibTopoART.TopoART.x\_F0 [protected]

Instance variable `x_F0` represents the current input vector.

## 5.12.4.5 long LibTopoART.TopoART.x\_F0\_dim [protected]

Instance variable `x_F0_dim` represents the input dimension.

## 5.12.5 Property Documentation

## 5.12.5.1 long LibTopoART.TopoART.ModuleNum [get]

Property `ModuleNum` represents the number of [TopoART](#) modules used. (The original [TopoART](#) uses two modules.)

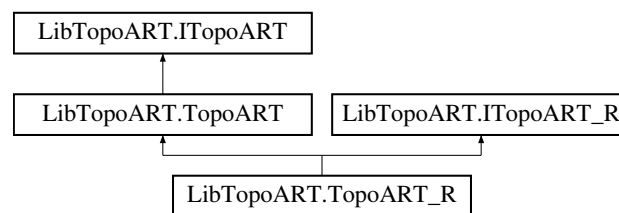
## 5.12.5.2 long LibTopoART.TopoART.Phi [get],[set]

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.

## 5.13 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`.*

## Public Attributes

- new const 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

### 5.13.1 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].

### 5.13.2 Constructor & Destructor Documentation

#### 5.13.2.1 `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).

#### 5.13.2.2 `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.
-------------	--------------------------------------

### 5.13.3 Member Function Documentation

#### 5.13.3.1 `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](#).

### 5.13.3.2 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](#).

## 5.13.4 Member Data Documentation

### 5.13.4.1 new const decimal LibTopoART.TopoART\_R.file\_format\_version = 0.01m

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

## 5.14 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

- const decimal `NO_PREDICTION` = `LibTopoART_info.UNDEFINED`  
*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.*
- 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.*

### 5.14.1 Detailed Description

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

### 5.14.2 Constructor & Destructor Documentation

5.14.2.1 `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`.

### 5.14.3 Member Function Documentation

5.14.3.1 `void LibTopoART.TopoART_R_Prediction.PrintPredictions ( )`

This method prints the predictions on the console.

### 5.14.4 Member Data Documentation

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

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

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

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

5.14.4.3 `const decimal LibTopoART.TopoART_R_Prediction.NO_PREDICTION = LibTopoART_info.UNDEFINED`

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.

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

Regression using TopoART-R.

### 5.15.1 Detailed Description

Regression 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.

The predicted results can be visualised using the R script `ShowTopoARTResults.R` provided in the subfolder `R`.

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

Simple TopoART sample.

### 5.16.1 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.

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

Sample using artificial two-dimensional data.

### 5.17.1 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`.



# Index

- BeginRecall
  - LibTopoART::Episodic\_TopoART, [10](#)
  - LibTopoART::IEpisodic\_Recall, [21](#)
- bm\_cluster\_ID
  - LibTopoART::F2\_output, [14](#)
- bm\_node\_ID
  - LibTopoART::F2\_output, [14](#)
- bm\_node\_activation
  - LibTopoART::F2\_output, [14](#)
- bm\_permanent\_cluster\_ID
  - LibTopoART::F2\_output, [14](#)
- bm\_permanent\_node\_ID
  - LibTopoART::F2\_output, [14](#)
- bm\_permanent\_node\_activation
  - LibTopoART::F2\_output, [14](#)
- ComputeClusterIDs
  - LibTopoART::Fast\_TopoART\_Base, [17](#)
  - LibTopoART::ITopoART, [24](#)
  - LibTopoART::TopoART, [29](#)
- d\_vec\_prediction
  - LibTopoART::TopoART\_R\_Prediction, [34](#)
- EndRecall
  - LibTopoART::Episodic\_TopoART, [11](#)
  - LibTopoART::IEpisodic\_Recall, [22](#)
- Episodic\_TopoART
  - LibTopoART::Episodic\_TopoART, [10](#)
- F2\_output
  - LibTopoART::F2\_output, [13](#)
- Fast\_TopoART
  - LibTopoART::Fast\_TopoART, [15](#)
- file\_format\_version
  - LibTopoART::Episodic\_TopoART, [12](#)
  - LibTopoART::Fast\_TopoART\_Base, [18](#)
  - LibTopoART::Hypersphere\_TopoART, [21](#)
  - LibTopoART::TopoART\_R, [33](#)
  - LibTopoART::TopoART, [30](#)
- float\_base\_type
  - LibTopoART::Fast\_TopoART\_Base, [18](#)
  - LibTopoART::TopoART, [30](#)
- GetBMOutput
  - LibTopoART::Fast\_TopoART\_Base, [17](#)
  - LibTopoART::ITopoART, [24](#)
  - LibTopoART::TopoART, [29](#)
- Hypersphere\_TopoART
  - LibTopoART::Hypersphere\_TopoART, [20](#), [21](#)
- i\_vec\_prediction
  - LibTopoART::TopoART\_R\_Prediction, [34](#)
- integer\_base\_type
  - LibTopoART::Fast\_TopoART\_Base, [18](#)
  - LibTopoART::TopoART, [30](#)
- InterEpisodeRecallStep
  - LibTopoART::Episodic\_TopoART, [11](#)
  - LibTopoART::IEpisodic\_Recall, [22](#)
- IntraEpisodeRecallStep
  - LibTopoART::Episodic\_TopoART, [11](#)
  - LibTopoART::IEpisodic\_Recall, [22](#)
- Learn
  - LibTopoART::Episodic\_TopoART, [11](#)
  - LibTopoART::Fast\_TopoART\_Base, [17](#)
  - LibTopoART::Fast\_TopoART, [15](#)
  - LibTopoART::ITopoART\_R, [25](#)
  - LibTopoART::ITopoART, [24](#)
  - LibTopoART::TopoART\_R, [32](#)
  - LibTopoART::TopoART, [29](#)
- LibTopoART.Episodic\_TopoART, [9](#)
- LibTopoART.F2\_output, [13](#)
- LibTopoART.Fast\_TopoART\_Base, [16](#)
- LibTopoART.Fast\_TopoART, [14](#)
- LibTopoART.Hypersphere\_TopoART, [19](#)
- LibTopoART.IEpisodic\_Recall, [21](#)
- LibTopoART.ITopoART\_R, [25](#)
- LibTopoART.ITopoART, [23](#)
- LibTopoART.LibTopoART\_info, [26](#)
- LibTopoART.TopoART\_R\_Prediction, [33](#)
- LibTopoART.TopoART\_R, [31](#)
- LibTopoART.TopoART, [27](#)
- LibTopoART::Episodic\_TopoART
  - BeginRecall, [10](#)
  - EndRecall, [11](#)
  - Episodic\_TopoART, [10](#)
  - file\_format\_version, [12](#)
  - InterEpisodeRecallStep, [11](#)
  - IntraEpisodeRecallStep, [11](#)
  - Learn, [11](#)
- LibTopoART::F2\_output
  - bm\_cluster\_ID, [14](#)
  - bm\_node\_ID, [14](#)
  - bm\_node\_activation, [14](#)
  - bm\_permanent\_cluster\_ID, [14](#)
  - bm\_permanent\_node\_ID, [14](#)
  - bm\_permanent\_node\_activation, [14](#)
  - F2\_output, [13](#)
- LibTopoART::Fast\_TopoART\_Base
  - ComputeClusterIDs, [17](#)

- file\_format\_version, 18
- float\_base\_type, 18
- GetBMOOutput, 17
- integer\_base\_type, 18
- Learn, 17
- ModuleNum, 19
- Phi, 19
- Save, 18
- SaveText, 18
- LibTopoART::Fast\_TopoART
  - Fast\_TopoART, 15
  - Learn, 15
- LibTopoART::Hypersphere\_TopoART
  - file\_format\_version, 21
  - Hypersphere\_TopoART, 20, 21
- LibTopoART::IEpisodic\_Recall
  - BeginRecall, 21
  - EndRecall, 22
  - InterEpisodeRecallStep, 22
  - IntraEpisodeRecallStep, 22
- LibTopoART::ITopoART\_R
  - Learn, 25
  - Predict, 26
- LibTopoART::ITopoART
  - ComputeClusterIDs, 24
  - GetBMOOutput, 24
  - Learn, 24
  - ModuleNum, 25
  - Phi, 25
  - Save, 24
  - SaveText, 24
- LibTopoART::LibTopoART\_info
  - networks, 27
  - UNDEFINED, 27
  - version, 27
- LibTopoART::TopoART\_R\_Prediction
  - d\_vec\_prediction, 34
  - i\_vec\_prediction, 34
  - NO\_PREDICTION, 34
  - PrintPredictions, 34
  - TopoART\_R\_Prediction, 34
- LibTopoART::TopoART\_R
  - file\_format\_version, 33
  - Learn, 32
  - Predict, 33
  - TopoART\_R, 32
- LibTopoART::TopoART
  - ComputeClusterIDs, 29
  - file\_format\_version, 30
  - float\_base\_type, 30
  - GetBMOOutput, 29
  - integer\_base\_type, 30
  - Learn, 29
  - ModuleNum, 31
  - Phi, 31
  - Save, 30
  - SaveText, 30
  - TopoART, 28, 29
- x\_F0, 30
- x\_F0\_dim, 30
- LibTopoART\_samples, 8
- LibTopoART\_samples.Episodic\_TopoART\_sample1, 12
- LibTopoART\_samples.Episodic\_TopoART\_sample2, 12
- LibTopoART\_samples.TopoART\_R\_sample1, 35
- LibTopoART\_samples.TopoART\_sample1, 35
- LibTopoART\_samples.TopoART\_sample2, 35
- LibTopoART, 7
- ModuleNum
  - LibTopoART::Fast\_TopoART\_Base, 19
  - LibTopoART::ITopoART, 25
  - LibTopoART::TopoART, 31
- NO\_PREDICTION
  - LibTopoART::TopoART\_R\_Prediction, 34
- networks
  - LibTopoART::LibTopoART\_info, 27
- Phi
  - LibTopoART::Fast\_TopoART\_Base, 19
  - LibTopoART::ITopoART, 25
  - LibTopoART::TopoART, 31
- Predict
  - LibTopoART::ITopoART\_R, 26
  - LibTopoART::TopoART\_R, 33
- PrintPredictions
  - LibTopoART::TopoART\_R\_Prediction, 34
- Save
  - LibTopoART::Fast\_TopoART\_Base, 18
  - LibTopoART::ITopoART, 24
  - LibTopoART::TopoART, 30
- SaveText
  - LibTopoART::Fast\_TopoART\_Base, 18
  - LibTopoART::ITopoART, 24
  - LibTopoART::TopoART, 30
- TopoART\_R\_Prediction
  - LibTopoART::TopoART\_R\_Prediction, 34
- TopoART\_R
  - LibTopoART::TopoART\_R, 32
- TopoART
  - LibTopoART::TopoART, 28, 29
- UNDEFINED
  - LibTopoART::LibTopoART\_info, 27
- version
  - LibTopoART::LibTopoART\_info, 27
- x\_F0
  - LibTopoART::TopoART, 30
- x\_F0\_dim
  - LibTopoART::TopoART, 30