

LibTopoART  
v0.44

Generated by Doxygen 1.8.4

Wed Jun 5 2013 21:42:16



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	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 . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	LibTopoART.F2_output Class Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Constructor & Destructor Documentation . . . . .	9
5.1.2.1	F2_output . . . . .	9
5.1.3	Member Data Documentation . . . . .	10
5.1.3.1	bm_cluster_ID . . . . .	10
5.1.3.2	bm_node_activation . . . . .	10
5.1.3.3	bm_node_ID . . . . .	10
5.1.3.4	bm_permanent_cluster_ID . . . . .	10
5.1.3.5	bm_permanent_node_activation . . . . .	10
5.1.3.6	bm_permanent_node_ID . . . . .	10
5.2	LibTopoART.Fast_TopoART Class Reference . . . . .	10
5.2.1	Detailed Description . . . . .	12
5.2.2	Constructor & Destructor Documentation . . . . .	12
5.2.2.1	Fast_TopoART . . . . .	12
5.2.2.2	Fast_TopoART . . . . .	12
5.2.3	Member Function Documentation . . . . .	12
5.2.3.1	ComputeClusterIDs . . . . .	12
5.2.3.2	GetBMOutput . . . . .	12

5.2.3.3	Learn	12
5.2.3.4	Save	13
5.2.3.5	SaveText	13
5.2.4	Member Data Documentation	13
5.2.4.1	file_format_version	13
5.2.4.2	float_base_type	13
5.2.4.3	integer_base_type	13
5.2.4.4	x_F0	13
5.2.4.5	x_F0_dim	13
5.3	LibTopoART.ITopoART Interface Reference	13
5.3.1	Detailed Description	14
5.3.2	Member Function Documentation	15
5.3.2.1	ComputeClusterIDs	15
5.3.2.2	GetBMOutput	15
5.3.2.3	Learn	15
5.3.2.4	Save	15
5.3.2.5	SaveText	15
5.4	LibTopoART.LibTopoART_info Struct Reference	15
5.4.1	Detailed Description	16
5.4.2	Member Data Documentation	16
5.4.2.1	networks	16
5.4.2.2	version	16
5.5	LibTopoART.TopoART Class Reference	16
5.5.1	Detailed Description	18
5.5.2	Constructor & Destructor Documentation	18
5.5.2.1	TopoART	18
5.5.2.2	TopoART	18
5.5.3	Member Function Documentation	18
5.5.3.1	ComputeClusterIDs	18
5.5.3.2	GetBMOutput	18
5.5.3.3	Learn	18
5.5.3.4	Save	19
5.5.3.5	SaveText	19
5.5.4	Member Data Documentation	19
5.5.4.1	file_format_version	19
5.5.4.2	float_base_type	19
5.5.4.3	integer_base_type	19
5.5.4.4	x_F0	19
5.5.4.5	x_F0_dim	19
5.6	LibTopoART.TopoART_R Class Reference	19

5.6.1	Detailed Description	20
5.6.2	Constructor & Destructor Documentation	20
5.6.2.1	TopoART_R	20
5.6.2.2	TopoART_R	21
5.6.3	Member Function Documentation	21
5.6.3.1	Learn	21
5.6.3.2	Predict	21
5.6.4	Member Data Documentation	21
5.6.4.1	file_format_version	21
5.7	LibTopoART.TopoART_R_Prediction Class Reference	21
5.7.1	Detailed Description	22
5.7.2	Constructor & Destructor Documentation	22
5.7.2.1	TopoART_R_Prediction	22
5.7.3	Member Function Documentation	22
5.7.3.1	PrintPredictions	22
5.7.4	Member Data Documentation	22
5.7.4.1	d_vec_prediction	22
5.7.4.2	i_vec_prediction	22
5.7.4.3	NO_PREDICTION	22
5.8	LibTopoART_samples.TopoART_R_sample1 Class Reference	23
5.8.1	Detailed Description	23
5.9	LibTopoART_samples.TopoART_sample1 Class Reference	23
5.9.1	Detailed Description	23
5.10	LibTopoART_samples.TopoART_sample2 Class Reference	23
5.10.1	Detailed Description	23



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

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





## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

LibTopoART.F2_output . . . . .	9
LibTopoART.ITopoART . . . . .	13
LibTopoART.Fast_TopoART . . . . .	10
LibTopoART.TopoART . . . . .	16
LibTopoART.TopoART_R . . . . .	19
LibTopoART.LibTopoART_info . . . . .	15
LibTopoART.TopoART_R_Prediction . . . . .	21
LibTopoART_samples.TopoART_R_sample1 . . . . .	23
LibTopoART_samples.TopoART_sample1 . . . . .	23
LibTopoART_samples.TopoART_sample2 . . . . .	23



## Chapter 3

# Class Index

### 3.1 Class List

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

<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 <code>y</code> and <code>c</code> . . . . .	9
<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." . . . . .	10
<a href="#">LibTopoART.ITopoART</a>	Interface summarising the basic <a href="#">TopoART</a> functionality. . . . .	13
<a href="#">LibTopoART.LibTopoART_info</a>	Struct <a href="#">LibTopoART_info</a> provides some metainformation regarding the respective implementation of <a href="#">LibTopoART</a> . . . . .	15
<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, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer." . . . . .	16
<a href="#">LibTopoART.TopoART_R</a>	Class <a href="#">TopoART_R</a> provides an implementation of the TopoART-R neural network as proposed in "Tscherepanow, Marko (2011). An Extended <a href="#">TopoART</a> Network for the Stable On-Line Learning of Regression Functions. In Proceedings of the International Conference on Neural Information Processing, LNCS 7063 (pp. 562–571). Berlin, Germany: Springer . . . . .	19
<a href="#">LibTopoART.TopoART_R_Prediction</a>	Class <a href="#">F2_output</a> contains a prediction made by a TopoART-R network. . . . .	21
<a href="#">LibTopoART_samples.TopoART_R_sample1</a>	Function approximation using TopoART-R . . . . .	23
<a href="#">LibTopoART_samples.TopoART_sample1</a>	Simple TopoART sample . . . . .	23
<a href="#">LibTopoART_samples.TopoART_sample2</a>	More complex TopoART sample . . . . .	23



## Chapter 4

# Namespace Documentation

### 4.1 Package LibTopoART

#### Classes

- struct [LibTopoART\\_info](#)  
*Struct [LibTopoART\\_info](#) provides some metainformation regarding the respective implementation of [LibTopoART](#).*
- 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 [TopoART\\_R\\_Prediction](#)  
*Class [F2\\_output](#) contains a prediction made by a [TopoART-R](#) network.*
- interface [ITopoART](#)  
*Interface summarising the basic [TopoART](#) functionality.*
- 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, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer."*
- 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 [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, LNCS 7063 (pp. 562–571). Berlin, Germany: Springer.*

### 4.2 Package LibTopoART\_samples

#### Classes

- class [TopoART\\_R\\_sample1](#)  
*Function approximation using [TopoART-R](#).*
- class [TopoART\\_sample1](#)  
*Simple [TopoART](#) sample.*
- class [TopoART\\_sample2](#)  
*More complex [TopoART](#) sample.*



## Chapter 5

# Class Documentation

### 5.1 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.*

- 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.*

- 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.*

#### 5.1.1 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`.

#### 5.1.2 Constructor & Destructor Documentation

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

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

### 5.1.3 Member Data Documentation

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

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

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

Instance variable `bm_node_activation` represents the activation of the best-matching node.

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

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

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

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

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

Instance variable `bm_permanent_node_activation` represents the activation of the best-matching permanent node.

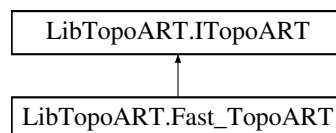
#### 5.1.3.6 long LibTopoART.F2\_output.bm\_permanent\_node\_ID

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

## 5.2 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.*
- 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 `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.*

### Protected Attributes

- readonly long `x_F0_dim`  
*Instance variable `x_F0_dim` represents the input dimension.*
- readonly long[] `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.*

### 5.2.1 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].

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 `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).

#### 5.2.2.2 `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.
-------------	--

### 5.2.3 Member Function Documentation

#### 5.2.3.1 `void LibTopoART.Fast_TopoART.ComputeClusterIDs ( )`

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

#### 5.2.3.2 `F2_output [] LibTopoART.Fast_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](#).

#### 5.2.3.3 `void LibTopoART.Fast_TopoART.Learn ( decimal[] input )`

This method performs a single training step.

## Parameters

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

Implements [LibTopoART.ITopoART](#).

## 5.2.3.4 void LibTopoART.Fast\_TopoART.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](#).

## 5.2.3.5 void LibTopoART.Fast\_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](#).

## 5.2.4 Member Data Documentation

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

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

## 5.2.4.2 readonly string LibTopoART.Fast\_TopoART.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.

## 5.2.4.3 readonly string LibTopoART.Fast\_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.

## 5.2.4.4 readonly long [] LibTopoART.Fast\_TopoART.x\_F0 [protected]

Instance variable `x_F0` represents the current input vector.

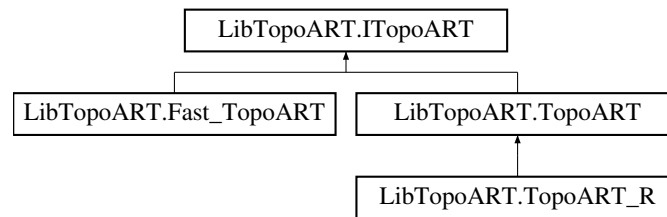
## 5.2.4.5 readonly long LibTopoART.Fast\_TopoART.x\_F0\_dim [protected]

Instance variable `x_F0_dim` represents the input dimension.

## 5.3 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.*

### 5.3.1 Detailed Description

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

### 5.3.2 Member Function Documentation

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

This method computes the cluster IDs for all neurons.

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

#### 5.3.2.2 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.Fast\\_TopoART](#), and [LibTopoART.TopoART](#).

#### 5.3.2.3 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.Fast\\_TopoART](#), and [LibTopoART.TopoART](#).

#### 5.3.2.4 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.Fast\\_TopoART](#), and [LibTopoART.TopoART](#).

#### 5.3.2.5 void LibTopoART.ITopoART.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.
-------------	---

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

## 5.4 LibTopoART.LibTopoART\_info Struct Reference

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

## Static Public Attributes

- static readonly decimal `version` =0.44m

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

- static readonly string[] `networks` ={"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](#).*

### 5.4.1 Detailed Description

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

### 5.4.2 Member Data Documentation

- 5.4.2.1 readonly string [] LibTopoART.LibTopoART\_info.networks ={"TopoART (class [TopoART](#), class [Fast\\_TopoART](#))", "TopoART-R (class [TopoART\\_R](#))"} [static]

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

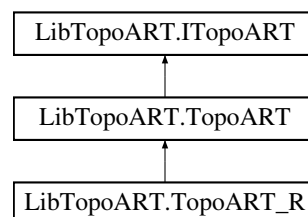
- 5.4.2.2 readonly decimal LibTopoART.LibTopoART\_info.version =0.44m [static]

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

## 5.5 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, 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

- readonly long [x\\_F0\\_dim](#)

*Instance variable `x_F0_dim` represents the input dimension.*

- readonly 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.*

### 5.5.1 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, 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  $[0,1]$ .

### 5.5.2 Constructor & Destructor Documentation

#### 5.5.2.1 `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).

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

### 5.5.3 Member Function Documentation

#### 5.5.3.1 `void LibTopoART.TopoART.ComputeClusterIDs ( )`

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

#### 5.5.3.2 `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](#).

#### 5.5.3.3 `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](#).

5.5.3.4 void LibTopoART.TopoART.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](#).

5.5.3.5 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](#).

## 5.5.4 Member Data Documentation

## 5.5.4.1 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](#).

## 5.5.4.2 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.

## 5.5.4.3 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.

## 5.5.4.4 readonly decimal [] LibTopoART.TopoART.x\_F0 [protected]

Instance variable `x_F0` represents the current input vector.

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

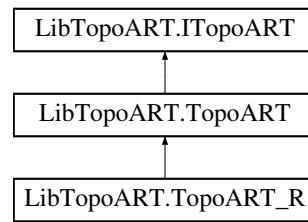
Instance variable `x_F0_dim` represents the input dimension.

## 5.6 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 Pro-

ceedings of the International Conference on Neural Information Processing, 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

- new static 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

### 5.6.1 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, LNCS 7063 (pp. 562–571). Berlin, Germany: Springer.

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

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 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).

#### 5.6.2.2 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.
-------------	--------------------------------------

### 5.6.3 Member Function Documentation

#### 5.6.3.1 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> .

#### 5.6.3.2 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.

### 5.6.4 Member Data Documentation

#### 5.6.4.1 new static 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](#).

## 5.7 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.

### 5.7.1 Detailed Description

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

### 5.7.2 Constructor & Destructor Documentation

#### 5.7.2.1 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`.

### 5.7.3 Member Function Documentation

#### 5.7.3.1 void LibTopoART.TopoART\_R\_Prediction.PrintPredictions ( )

This method prints the predictions on the console.

### 5.7.4 Member Data Documentation

#### 5.7.4.1 readonly decimal [] LibTopoART.TopoART\_R\_Prediction.d\_vec\_prediction

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

#### 5.7.4.2 readonly decimal [] LibTopoART.TopoART\_R\_Prediction.i\_vec\_prediction

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

#### 5.7.4.3 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.

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

Function approximation using TopoART-R.

### 5.8.1 Detailed Description

Function approximation using TopoART-R.

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

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

Simple TopoART sample.

### 5.9.1 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.

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

More complex TopoART sample.

### 5.10.1 Detailed Description

More complex TopoART sample.

Train 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.

# Index

- bm\_cluster\_ID
  - LibTopoART::F2\_output, [10](#)
- bm\_node\_ID
  - LibTopoART::F2\_output, [10](#)
- bm\_node\_activation
  - LibTopoART::F2\_output, [10](#)
- bm\_permanent\_cluster\_ID
  - LibTopoART::F2\_output, [10](#)
- bm\_permanent\_node\_ID
  - LibTopoART::F2\_output, [10](#)
- bm\_permanent\_node\_activation
  - LibTopoART::F2\_output, [10](#)
- ComputeClusterIDs
  - LibTopoART::Fast\_TopoART, [12](#)
  - LibTopoART::ITopoART, [15](#)
  - LibTopoART::TopoART, [18](#)
- d\_vec\_prediction
  - LibTopoART::TopoART\_R\_Prediction, [22](#)
- F2\_output
  - LibTopoART::F2\_output, [9](#)
- Fast\_TopoART
  - LibTopoART::Fast\_TopoART, [12](#)
- file\_format\_version
  - LibTopoART::Fast\_TopoART, [13](#)
  - LibTopoART::TopoART, [19](#)
  - LibTopoART::TopoART\_R, [21](#)
- float\_base\_type
  - LibTopoART::Fast\_TopoART, [13](#)
  - LibTopoART::TopoART, [19](#)
- GetBMOutput
  - LibTopoART::Fast\_TopoART, [12](#)
  - LibTopoART::ITopoART, [15](#)
  - LibTopoART::TopoART, [18](#)
- i\_vec\_prediction
  - LibTopoART::TopoART\_R\_Prediction, [22](#)
- integer\_base\_type
  - LibTopoART::Fast\_TopoART, [13](#)
  - LibTopoART::TopoART, [19](#)
- Learn
  - LibTopoART::Fast\_TopoART, [12](#)
  - LibTopoART::ITopoART, [15](#)
  - LibTopoART::TopoART, [18](#)
  - LibTopoART::TopoART\_R, [21](#)
- LibTopoART, [7](#)
- LibTopoART.F2\_output, [9](#)
- LibTopoART.Fast\_TopoART, [10](#)
- LibTopoART.ITopoART, [13](#)
- LibTopoART.LibTopoART\_info, [15](#)
- LibTopoART.TopoART, [16](#)
- LibTopoART.TopoART\_R, [19](#)
- LibTopoART.TopoART\_R\_Prediction, [21](#)
- LibTopoART::F2\_output
  - bm\_cluster\_ID, [10](#)
  - bm\_node\_ID, [10](#)
  - bm\_node\_activation, [10](#)
  - bm\_permanent\_cluster\_ID, [10](#)
  - bm\_permanent\_node\_ID, [10](#)
  - bm\_permanent\_node\_activation, [10](#)
  - F2\_output, [9](#)
- LibTopoART::Fast\_TopoART
  - ComputeClusterIDs, [12](#)
  - Fast\_TopoART, [12](#)
  - file\_format\_version, [13](#)
  - float\_base\_type, [13](#)
  - GetBMOutput, [12](#)
  - integer\_base\_type, [13](#)
  - Learn, [12](#)
  - Save, [13](#)
  - SaveText, [13](#)
  - x\_F0, [13](#)
  - x\_F0\_dim, [13](#)
- LibTopoART::ITopoART
  - ComputeClusterIDs, [15](#)
  - GetBMOutput, [15](#)
  - Learn, [15](#)
  - Save, [15](#)
  - SaveText, [15](#)
- LibTopoART::LibTopoART\_info
  - networks, [16](#)
  - version, [16](#)
- LibTopoART::TopoART
  - ComputeClusterIDs, [18](#)
  - file\_format\_version, [19](#)
  - float\_base\_type, [19](#)
  - GetBMOutput, [18](#)
  - integer\_base\_type, [19](#)
  - Learn, [18](#)
  - Save, [19](#)
  - SaveText, [19](#)
  - TopoART, [18](#)
  - x\_F0, [19](#)
  - x\_F0\_dim, [19](#)
- LibTopoART::TopoART\_R
  - file\_format\_version, [21](#)

- Learn, [21](#)
- Predict, [21](#)
- TopoART\_R, [20](#), [21](#)
- LibTopoART::TopoART\_R\_Prediction
  - d\_vec\_prediction, [22](#)
  - i\_vec\_prediction, [22](#)
  - PrintPredictions, [22](#)
- LibTopoART\_samples, [7](#)
- LibTopoART\_samples.TopoART\_R\_sample1, [23](#)
- LibTopoART\_samples.TopoART\_sample1, [23](#)
- LibTopoART\_samples.TopoART\_sample2, [23](#)
- NO\_PREDICTION
  - LibTopoART::TopoART\_R\_Prediction, [22](#)
- networks
  - LibTopoART::LibTopoART\_info, [16](#)
- Predict
  - LibTopoART::TopoART\_R, [21](#)
- PrintPredictions
  - LibTopoART::TopoART\_R\_Prediction, [22](#)
- Save
  - LibTopoART::Fast\_TopopART, [13](#)
  - LibTopoART::ITopoART, [15](#)
  - LibTopoART::TopoART, [19](#)
- SaveText
  - LibTopoART::Fast\_TopopART, [13](#)
  - LibTopoART::ITopoART, [15](#)
  - LibTopoART::TopoART, [19](#)
- TopoART
  - LibTopoART::TopoART, [18](#)
- TopoART\_R
  - LibTopoART::TopoART\_R, [20](#), [21](#)
- TopoART\_R\_Prediction
  - LibTopoART::TopoART\_R\_Prediction, [22](#)
- version
  - LibTopoART::LibTopoART\_info, [16](#)
- x\_F0
  - LibTopoART::Fast\_TopopART, [13](#)
  - LibTopoART::TopoART, [19](#)
- x\_F0\_dim
  - LibTopoART::Fast\_TopopART, [13](#)
  - LibTopoART::TopoART, [19](#)