

LibTopoART
v0.78

Generated by Doxygen 1.8.13

Wed Jan 3 2018 11:43:52

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	2
2.1	Class Hierarchy	2
3	Class Index	3
3.1	Class List	3
4	Namespace Documentation	5
4.1	LibTopoART Namespace Reference	5
4.1.1	Enumeration Type Documentation	7
4.2	LibTopoART_samples Namespace Reference	7
5	Class Documentation	8
5.1	LibTopoART.Episodic_TopoART Class Reference	8
5.1.1	Detailed Description	9
5.1.2	Constructor & Destructor Documentation	9
5.1.3	Member Function Documentation	9
5.1.4	Property Documentation	11
5.2	LibTopoART_samples.Episodic_TopoART_sample1 Class Reference	11
5.2.1	Detailed Description	12
5.3	LibTopoART_samples.Episodic_TopoART_sample2 Class Reference	12
5.3.1	Detailed Description	12
5.4	LibTopoART.F2_output Class Reference	12
5.4.1	Detailed Description	13
5.4.2	Constructor & Destructor Documentation	13
5.4.3	Member Data Documentation	13
5.5	LibTopoART.Fast_TopoART Class Reference	14
5.5.1	Detailed Description	15
5.5.2	Constructor & Destructor Documentation	15

5.5.3	Member Function Documentation	15
5.6	LibTopoART.Fast_TopoART_base Class Reference	16
5.6.1	Detailed Description	17
5.6.2	Member Function Documentation	17
5.6.3	Property Documentation	21
5.7	LibTopoART.Fast_TopoART_C Class Reference	21
5.7.1	Detailed Description	22
5.7.2	Constructor & Destructor Documentation	22
5.7.3	Member Function Documentation	23
5.7.4	Member Data Documentation	25
5.7.5	Property Documentation	25
5.8	LibTopoART.Hypersphere_TopoART Class Reference	25
5.8.1	Detailed Description	26
5.8.2	Constructor & Destructor Documentation	26
5.8.3	Property Documentation	27
5.9	LibTopoART.Hypersphere_TopoART_C Class Reference	27
5.9.1	Detailed Description	28
5.9.2	Constructor & Destructor Documentation	28
5.9.3	Member Function Documentation	30
5.9.4	Member Data Documentation	31
5.9.5	Property Documentation	31
5.10	LibTopoART.IAdaptationStateCheck Interface Reference	32
5.10.1	Detailed Description	32
5.10.2	Member Function Documentation	32
5.11	LibTopoART.IEpisodic_recall Interface Reference	33
5.11.1	Detailed Description	33
5.11.2	Member Function Documentation	33
5.12	LibTopoART.InvalidClassIDException Class Reference	34
5.12.1	Detailed Description	34
5.13	LibTopoART.InvalidFileException Class Reference	35

5.13.1 Detailed Description	35
5.14 LibTopoART.InvalidStateException Class Reference	35
5.14.1 Detailed Description	35
5.15 LibTopoART.ITopoART Interface Reference	35
5.15.1 Detailed Description	36
5.15.2 Member Function Documentation	36
5.15.3 Property Documentation	39
5.16 LibTopoART.ITopoART_C Interface Reference	39
5.16.1 Detailed Description	40
5.16.2 Member Function Documentation	40
5.17 LibTopoART.ITopoART_R Interface Reference	41
5.17.1 Detailed Description	42
5.17.2 Member Function Documentation	42
5.18 LibTopoART.LibTopoART_info Struct Reference	43
5.18.1 Detailed Description	43
5.18.2 Member Data Documentation	44
5.19 LibTopoART.TopoART Class Reference	44
5.19.1 Detailed Description	46
5.19.2 Constructor & Destructor Documentation	46
5.19.3 Member Function Documentation	47
5.19.4 Property Documentation	50
5.20 LibTopoART.TopoART_C Class Reference	50
5.20.1 Detailed Description	51
5.20.2 Constructor & Destructor Documentation	51
5.20.3 Member Function Documentation	52
5.20.4 Member Data Documentation	54
5.20.5 Property Documentation	54
5.21 LibTopoART.TopoART_C_Prediction Struct Reference	54
5.21.1 Detailed Description	54
5.21.2 Constructor & Destructor Documentation	55

5.21.3 Member Data Documentation	56
5.22 LibTopoART_samples.TopoART_C_sample1 Class Reference	56
5.22.1 Detailed Description	56
5.23 LibTopoART_samples.TopoART_C_sample2 Class Reference	56
5.23.1 Detailed Description	56
5.24 LibTopoART.TopoART_R Class Reference	57
5.24.1 Detailed Description	57
5.24.2 Constructor & Destructor Documentation	57
5.24.3 Member Function Documentation	58
5.24.4 Property Documentation	60
5.25 LibTopoART.TopoART_R_Prediction Struct Reference	60
5.25.1 Detailed Description	60
5.25.2 Constructor & Destructor Documentation	60
5.25.3 Member Function Documentation	61
5.25.4 Member Data Documentation	61
5.26 LibTopoART_samples.TopoART_R_sample1 Class Reference	61
5.26.1 Detailed Description	62
5.27 LibTopoART_samples.TopoART_R_sample2 Class Reference	62
5.27.1 Detailed Description	62
5.28 LibTopoART_samples.TopoART_sample1 Class Reference	62
5.28.1 Detailed Description	62
5.29 LibTopoART_samples.TopoART_sample2 Class Reference	62
5.29.1 Detailed Description	62

Index**63****1 Namespace Index****1.1 Packages**

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

LibTopoART**5**

LibTopoART_samples	7
------------------------------------	-------------------

2 Hierarchical Index

2.1 Class Hierarchy

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

LibTopoART_samples.Episodic_TopoART_sample1	11
LibTopoART_samples.Episodic_TopoART_sample2	12
Exception	
LibTopoART.InvalidClassIDException	34
LibTopoART.InvalidFileException	35
LibTopoART.InvalidStateException	35
LibTopoART.F2_output	12
LibTopoART.IAdaptationStateCheck	32
LibTopoART.ITopoART	35
LibTopoART.Fast_TopoART_base	16
LibTopoART.Episodic_TopoART	8
LibTopoART.Fast_TopoART	14
LibTopoART.Fast_TopoART_C	21
LibTopoART.ITopoART_C	39
LibTopoART.Fast_TopoART_C	21
LibTopoART.Hypersphere_TopoART_C	27
LibTopoART.TopoART_C	50
LibTopoART.ITopoART_R	41
LibTopoART.TopoART_R	57
LibTopoART.TopoART	44
LibTopoART.Hypersphere_TopoART	25
LibTopoART.Hypersphere_TopoART_C	27
LibTopoART.TopoART_C	50
LibTopoART.TopoART_R	57
IDisposable	
LibTopoART.Fast_TopoART_base	16
LibTopoART.TopoART	44

LibTopoART.Episodic_recall	33
LibTopoART.Episodic_TopoART	8
LibTopoART.LibTopoART_info	43
LibTopoART.TopoART_C_Prediction	54
LibTopoART_samples.TopoART_C_sample1	56
LibTopoART_samples.TopoART_C_sample2	56
LibTopoART.TopoART_R_Prediction	60
LibTopoART_samples.TopoART_R_sample1	61
LibTopoART_samples.TopoART_R_sample2	62
LibTopoART_samples.TopoART_sample1	62
LibTopoART_samples.TopoART_sample2	62

3 Class Index

3.1 Class List

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

[LibTopoART.Episodic_TopoART](#)

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

8

[LibTopoART_samples.Episodic_TopoART_sample1](#)

Episodic clustering sample using synthetic two-dimensional data

11

[LibTopoART_samples.Episodic_TopoART_sample2](#)

Episodic clustering sample using real-world video data

12

[LibTopoART.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 .

12

[LibTopoART.Fast_TopoART](#)

Class [Fast_TopoART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Marko Tscherepanow (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."

14

[LibTopoART.Fast_TopoART_base](#)

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

16

[LibTopoART.Fast_TopoART_C](#)

Class [Fast_TopoART_C](#) provides an implementation of the TopoART-C neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

21

LibTopoART.Hypersphere_TopoART

Class **Hypersphere_TopoART** provides an implementation of the Hypersphere **TopoART** neural network as proposed in "Marko Tscherepanow (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."

25

LibTopoART.Hypersphere_TopoART_C

Class **Hypersphere_TopoART_C** provides an implementation of the Hypersphere **TopoART-C** neural network. Hypersphere **TopoART-C** is a combination of Hypersphere **TopoART** as proposed in "Marko Tscherepanow (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." and **TopoART-C** as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

27

LibTopoART.IAdaptationStateCheck

Interface enabling checks whether certain adaptations of a network occurred.

32

LibTopoART.IEpisodic_recall

Interface summarising the episodic recall functionality.

33

LibTopoART.InvalidClassIDException

Exception signalling an invalid class ID.

34

LibTopoART.InvalidFileException

Exception signalling an invalid file.

35

LibTopoART.InvalidStateException

Exception signalling an invalid state of the neural network.

35

LibTopoART.ITopoART

Interface summarising the basic **TopoART** functionality.

35

LibTopoART.ITopoART_C

Interface summarising the **TopoART-C** functionality.

39

LibTopoART.ITopoART_R

Interface summarising the **TopoART-R** functionality.

41

LibTopoART.LibTopoART_info

Struct **LibTopoART_info** provides some metainformation regarding the respective implementation of **LibTopoART**.

43

LibTopoART.TopoART

Class **TopoART** provides an implementation of the **TopoART** neural network as proposed in "Marko Tscherepanow (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."

44

LibTopoART.TopoART_C

Class **TopoART_C** provides an implementation of the **TopoART-C** neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

50

LibTopoART.TopoART_C_Prediction	
Struct TopoART_C_Prediction contains a prediction made by a TopoART-C network.	54
LibTopoART_samples.TopoART_C_sample1	
Simple classification sample	56
LibTopoART_samples.TopoART_C_sample2	
Classification sample using more complex synthetic two-dimensional data with associated class IDs	56
LibTopoART.TopoART_R	
Class TopoART_R provides an implementation of the TopoART-R neural network as proposed in "Marko Tscherepanow (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."	57
LibTopoART.TopoART_R_Prediction	
Struct TopoART_R_Prediction contains a prediction made by a TopoART-R network.	60
LibTopoART_samples.TopoART_R_sample1	
Regression sample using TopoART-R. (simplified version)	61
LibTopoART_samples.TopoART_R_sample2	
Regression sample using TopoART-R. (advanced version)	62
LibTopoART_samples.TopoART_sample1	
Simple clustering sample	62
LibTopoART_samples.TopoART_sample2	
Clustering sample using more complex synthetic two-dimensional data	62

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 (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29. Montpellier, France."
- 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 "Marko Tscherepanow (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 [Fast_TopoART_base](#)
Base class providing functionality common to several [TopoART](#) networks.
- class [Fast_TopoART_C](#)
Class [Fast_TopoART_C](#) provides an implementation of the TopoART-C neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

- class [Hypersphere_TopoART](#)

Class [Hypersphere_TopoART](#) provides an implementation of the Hypersphere [TopoART](#) neural network as proposed in "Marko Tscherepanow (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."

- class [Hypersphere_TopoART_C](#)

Class [Hypersphere_TopoART_C](#) provides an implementation of the Hypersphere TopoART-C neural network. Hypersphere TopoART-C is a combination of Hypersphere [TopoART](#) as proposed in "Marko Tscherepanow (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." and TopoART-C as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

- interface [IAdaptationStateCheck](#)

Interface enabling checks whether certain adaptations of a network occurred.

- interface [IEpisodic_recall](#)

Interface summarising the episodic recall functionality.

- class [InvalidClassIDException](#)

Exception signalling an invalid class ID.

- class [InvalidFileException](#)

Exception signalling an invalid file.

- class [InvalidStateException](#)

Exception signalling an invalid state of the neural network.

- interface [ITopoART](#)

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

- interface [ITopoART_C](#)

Interface summarising the TopoART-C functionality.

- interface [ITopoART_R](#)

Interface summarising the TopoART-R functionality.

- struct [LibTopoART_info](#)

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

- class [TopoART](#)

Class [TopoART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Marko Tscherepanow (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_C](#)

Class [TopoART_C](#) provides an implementation of the TopoART-C neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

- struct [TopoART_C_Prediction](#)

Struct [TopoART_C_Prediction](#) contains a prediction made by a TopoART-C network.

- class [TopoART_R](#)

Class [TopoART_R](#) provides an implementation of the TopoART-R neural network as proposed in "Marko Tscherepanow (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."

- struct [TopoART_R_Prediction](#)

Struct [TopoART_R_Prediction](#) contains a prediction made by a TopoART-R network.

Enumerations

- enum [AdaptationState](#) {
[AdaptationState.NO_ADAPTATION](#) = 0, [AdaptationState.ADDED_NODE_CANDIDATE](#) = 0x0001,
[AdaptationState.ADAPTED_NONPERMANENT_WEIGHT](#) = 0x0002, [AdaptationState.ADDED_EDGE_CANDIDATE](#) = 0x0004,
[AdaptationState.REMOVED_NODE_CANDIDATE](#) = 0x0008, [AdaptationState.REMOVED_EDGE_CANDIDATE](#) = 0x0010, [AdaptationState.ANY_NONPERMANENT_ADAPTATION_MASK](#) = 0x00ff, [AdaptationState.ADDED_PERMANENT_NODE](#) = 0x0100,
[AdaptationState.ADAPTED_PERMANENT_WEIGHT](#) = 0x0200, [AdaptationState.ADDED_PERMANENT_EDGE](#) = 0x0400, [AdaptationState.ANY_PERMANENT_ADAPTATION_MASK](#) = 0xff00 }

Enumeration specifying possible adaptation states.

4.1.1 Enumeration Type Documentation

4.1.1.1 AdaptationState

```
enum LibTopoART.AdaptationState [strong]
```

Enumeration specifying possible adaptation states.

Enumerator

NO_ADAPTATION	No adaptation occurred.
ADDED_NODE_CANDIDATE	Added one or more node candidates.
ADAPTED_NONPERMANENT_WEIGHT	The change of at least a single weight of one node candidate exceeds the given threshold.
ADDED_EDGE_CANDIDATE	Added an edge from/to a node candidate.
REMOVED_NODE_CANDIDATE	Removed one or more node candidates.
REMOVED_EDGE_CANDIDATE	Removed one or more node candidates.
ANY_NONPERMANENT_ADAPTATION_MASK	Mask for all non-permanent adaptations.
ADDED_PERMANENT_NODE	Added one or more permanent nodes.
ADAPTED_PERMANENT_WEIGHT	The change of at least a single weight of one permanent node exceeds the given threshold.
ADDED_PERMANENT_EDGE	Added an edge between two permanent nodes.
ANY_PERMANENT_ADAPTATION_MASK	Mask for all permanent adaptations.

4.2 LibTopoART_samples Namespace Reference

Classes

- class [Episodic_TopoART_sample1](#)
Episodic clustering sample using synthetic two-dimensional data.
- class [Episodic_TopoART_sample2](#)
Episodic clustering sample using real-world video data.
- class [TopoART_C_sample1](#)
Simple classification sample.

- class [TopoART_C_sample2](#)
Classification sample using more complex synthetic two-dimensional data with associated class IDs.
- class [TopoART_R_sample1](#)
Regression sample using TopoART-R. (simplified version)
- class [TopoART_R_sample2](#)
Regression sample using TopoART-R. (advanced version)
- class [TopoART_sample1](#)
Simple clustering sample.
- class [TopoART_sample2](#)
Clustering sample using more complex synthetic two-dimensional data.

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 (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29. Montpellier, France."

Inherits [LibTopoART.Fast_TopoART_base](#), and [LibTopoART.IEpisodic_recall](#).

Public Member Functions

- [Episodic_TopoART](#) (long input_dimension, long module_number, decimal rho_a, long t_max)
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.

Properties

- new decimal [FileFormatVersion](#) [get]
Property `FileFormatVersion` returns the version of the file format used by class [Episodic_TopoART](#).
- long [T_max](#) [get]
Property `T_max` represents the maximum considered time frame.

Additional Inherited Members

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 (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29. Montpellier, France."

5.1.2 Constructor & Destructor Documentation

5.1.2.1 [Episodic_TopoART\(\)](#) [1/2]

```
LibTopoART.Episodic_TopoART.Episodic_TopoART (
    long input_dimension,
    long module_number,
    decimal rho_a,
    long t_max )
```

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 TopoART modules.
<i>rho_a</i>	The vigilance parameter of the first Episodic TopoART module (ETA a).
<i>t_max</i>	The parameter limiting the considered time frame.

5.1.2.2 [Episodic_TopoART\(\)](#) [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 TopoART file.
-------------	---

5.1.3 Member Function Documentation

5.1.3.1 BeginRecall()

```
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 EndRecall()

```
void LibTopoART.Episodic_TopoART.EndRecall ( )
```

This method stops the recall process and frees temporary resources.

Implements [LibTopoART.IEpisodic_recall](#).

5.1.3.3 InterEpisodeRecallStep()

```
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 IntraEpisodeRecallStep()

```
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 Learn()

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

This method performs a single training step.

The spatial weights are adapted as in the original [TopoART](#) network. In contrast, the adaptation of the temporal weight $w_{\{j,2\}^{F2,t}}$ occurring only in Episodic [TopoART](#) is slightly different↵: $w_{\{j,2\}^{F2,t}}(t+1) = \text{beta_j} * \text{Max}(t_2^{F1}(t), w_{\{j,2\}^{F2,t}}(t) + (1 - \text{beta_j}) * w_{\{j,2\}^{F2,t}}(t)$ for $j = \text{bm}$ or $j = \text{sbm}$. (Note: $w_{\{j,1\}^{F2,t}}$ remains constant over the life time of a node.)

Parameters

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

Implements [LibTopoART.Fast_TopoART_base](#).

5.1.4 Property Documentation

5.1.4.1 FileFormatVersion

```
new decimal LibTopoART.Episodic_TopoART.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [Episodic_TopoART](#).

5.2 LibTopoART_samples.Episodic_TopoART_sample1 Class Reference

Episodic clustering sample using synthetic two-dimensional data.

5.2.1 Detailed Description

Episodic clustering sample using synthetic two-dimensional data.

Like in Section 4.1 of "Marko Tscherepanow, Sina Kühnel, and Sören Riechers (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29. Montpellier, France.", 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

Episodic clustering sample using real-world video data.

5.3.1 Detailed Description

Episodic clustering sample using real-world video data.

Like in Section 4.2 of "Marko Tscherepanow, Sina Kühnel, and Sören Riechers (2012). Episodic Clustering of Data Streams Using a Topology-Learning Neural Network. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29. Montpellier, France.", 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 F2_output()

```
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 bm_cluster_ID

```
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 bm_node_activation

```
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 `bm_node_ID`

```
long LibTopoART.F2_output.bm_node_ID
```

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

5.4.3.4 `bm_permanent_cluster_ID`

```
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 `bm_permanent_node_activation`

```
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 `bm_permanent_node_ID`

```
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 "Marko Tscherepanow (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."

Inherits [LibTopoART.Fast_TopoART_base](#).

Inherited by [LibTopoART.Fast_TopoART_C](#).

Public Member Functions

- [Fast_TopoART](#) (long input_dimension, long module_number, decimal rho_a)
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 "Marko Tscherepanow (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 mapped to `int` 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 `Fast_TopoART()` [1/2]

```
LibTopoART.Fast_TopoART.Fast_TopoART (
    long input_dimension,
    long module_number,
    decimal rho_a )
```

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 <code>TopoART</code> modules.
<i>rho_a</i>	The vigilance parameter of the first <code>TopoART</code> module (TA a).

5.5.2.2 `Fast_TopoART()` [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 <code>TopoART</code> file.
-------------	---

5.5.3 Member Function Documentation

5.5.3.1 Learn()

```
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](#).

Reimplemented in [LibTopoART.Fast_TopoART_C](#).

5.6 LibTopoART.Fast_TopoART_base Class Reference

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

Inherits [LibTopoART.ITopoART](#), and [IDisposable](#).

Inherited by [LibTopoART.Episodic_TopoART](#), and [LibTopoART.Fast_TopoART](#).

Public Member Functions

- abstract void [Learn](#) (decimal[] input)
This method performs a single training step.
- void [Dispose](#) ()
Releases all resources used by the [LibTopoART.Fast_TopoART_base](#) object.
- 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.
- [F2_output](#) [] [GetBMOutput](#) (decimal[] input, bool[] mask_vector)
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.
- void [Save](#) (string path, bool compatibility_mode)
This method saves the entire network as a binary file.
- void [ResetAdaptationState](#) ()
This method resets the adaptation state to [AdaptationState.NO_ADAPTATION](#).
- [AdaptationState](#) [GetAdaptationState](#) (decimal epsilon=0.001m)
This method returns the current adaptation state.

Protected Member Functions

- virtual void [Dispose](#) (bool disposing)
Release resources used by the [LibTopoART.Fast_TopoART_base](#) object.

Properties

- decimal [Alpha](#) [get, set]
Property Alpha represents the choice parameter alpha.
- decimal [Beta_sbm](#) [get, set]
Property Beta_sbm represents the learning rate of the second best-matching nodes.
- long [] [ClusterNum](#) [get]
Property ClusterNum represents the number of TopoART clusters found by each module.
- long [LearningSteps](#) [get, protected set]
Property LearningSteps represents the total number of performed learning steps.
- long [ModuleNum](#) [get]
Property ModuleNum represents the number of TopoART modules used. (The original TopoART uses two modules.)
- long [] [NodeNum](#) [get]
Property NodeNum represents the number of TopoART nodes used by each module.
- 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.
- string [IntegerBaseType](#) [get]
Property IntegerBaseType returns a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.
- decimal [FileFormatVersion](#) [get]
Property FileFormatVersion returns the version of the file format used by class Fast_TopoART_base.
- string [FloatBaseType](#) [get]
Property FloatBaseType returns a string containing the data type used for representing floating point variables (input, weights, etc.) internally.
- decimal [TopoARTFileFormatVersion](#) [get]
Property TopoARTFileFormatVersion returns the version of the file format used by class Fast_TopoART_base.

5.6.1 Detailed Description

Base class providing functionality common to several TopoART networks.

5.6.2 Member Function Documentation

5.6.2.1 ComputeClusterIDs()

```
void LibTopoART.Fast_TopoART_base.ComputeClusterIDs ( )
```

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

5.6.2.2 Dispose() [1/2]

```
void LibTopoART.Fast_TopoART_base.Dispose ( )
```

Releases all resources used by the [LibTopoART.Fast_TopoART_base](#) object.

Call [Dispose\(\)](#) when you are finished using the [LibTopoART.Fast_TopoART_base](#). The [Dispose\(\)](#) method leaves the [LibTopoART.Fast_TopoART_base](#) in an unusable state. After calling [Dispose\(\)](#), you must release all references to the [LibTopoART.Fast_TopoART_base](#) so the garbage collector can reclaim the memory that the [LibTopoART.Fast_TopoART_base](#) was occupying.

5.6.2.3 Dispose() [2/2]

```
virtual void LibTopoART.Fast_TopoART_base.Dispose (
    bool disposing ) [protected], [virtual]
```

Release resources used by the [LibTopoART.Fast_TopoART_base](#) object.

Parameters

<i>disposing</i>	If set to <code>true</code> all managed resources are released.
------------------	---

5.6.2.4 GetAdaptationState()

```
AdaptationState LibTopoART.Fast_TopoART_base.GetAdaptationState (
    decimal epsilon = 0.001m )
```

This method returns the current adaptation state.

Parameters

<i>epsilon</i>	The threshold for weight adaptations to be considered.
----------------	--

Returns

An enumeration describing the adaptation state.

Implements [LibTopoART.IAdaptationStateCheck](#).

5.6.2.5 GetBMOutput() [1/2]

```
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 $x(t)$.
--------------	---------------------------

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](#).

5.6.2.6 GetBMOutput() [2/2]

```
F2_output [] LibTopoART.Fast_TopoART_base.GetBMOutput (
    decimal [] input,
    bool [] mask_vector )
```

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

Parameters

<i>input</i>	The input vector $x(t)$.
<i>mask_vector</i>	A mask vector excluding individual dimensions of $x(t)$ from the computation. (Setting an element of the mask vector to <code>true</code> , excludes the corresponding elements of $x(t)$.)

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](#).

5.6.2.7 Learn()

```
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.Fast_TopoART_C](#), [LibTopoART.Episodic_TopoART](#), and [LibTopoART.Fast_TopoART](#).

5.6.2.8 ResetAdaptationState()

```
void LibTopoART.Fast_TopoART_base.ResetAdaptationState ( )
```

This method resets the adaptation state to [AdaptationState.NO_ADAPTATION](#).

Implements [LibTopoART.IAdaptationStateCheck](#).

5.6.2.9 Save() [1/2]

```
void LibTopoART.Fast_TopoART_base.Save (
    string path )
```

This method saves the entire network as a binary file.

Parameters

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

Implements [LibTopoART.ITopoART](#).

5.6.2.10 Save() [2/2]

```
void LibTopoART.Fast_TopoART_base.Save (
    string path,
    bool compatibility_mode )
```

This method saves the entire network as a binary file.

Parameters

<i>path</i>	A string representing the path of the file to save.
<i>compatibility_mode</i>	If true, the file is saved in compatibility mode.

5.6.2.11 SaveText()

```
void LibTopoART.Fast_TopoART_base.SaveText (
    string path )
```

This method saves the entire network as a text file.

Parameters

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

Implements [LibTopoART.ITopoART](#).

5.6.3 Property Documentation

5.6.3.1 FileFormatVersion

```
decimal LibTopoART.Fast_TopoART_base.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [Fast_TopoART_base](#).

5.6.3.2 FloatBaseType

```
string LibTopoART.Fast_TopoART_base.FloatBaseType [get]
```

Property `FloatBaseType` returns a string containing the data type used for representing floating point variables (input, weights, etc.) internally.

5.6.3.3 IntegerBaseType

```
string LibTopoART.Fast_TopoART_base.IntegerBaseType [get]
```

Property `IntegerBaseType` returns a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.

5.6.3.4 TopoARTFileFormatVersion

```
decimal LibTopoART.Fast_TopoART_base.TopoARTFileFormatVersion [get]
```

Property `TopoARTFileFormatVersion` returns the version of the file format used by class [Fast_TopoART_base](#).

5.7 LibTopoART.Fast_TopoART_C Class Reference

Class [Fast_TopoART_C](#) provides an implementation of the TopoART-C neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

Inherits [LibTopoART.Fast_TopoART](#), and [LibTopoART.ITopoART_C](#).

Public Member Functions

- [Fast_TopoART_C](#) (long input_dimension, long module_number, decimal rho_a)
This constructor initialises a TopoART-C network.
- [Fast_TopoART_C](#) (string path)
This constructor loads a saved TopoART-C network.
- override void [Learn](#) (decimal[] input)
This method performs a single training step and sets the class ID corresponding to input to `UNDEFINED_CLASS_ID`.
- void [Learn](#) (decimal[] input, long class_ID)
This method performs a single training step.
- long [Predict](#) (decimal[] input, long nu)
This method predicts the class ID.
- [TopoART_C_Prediction Predict](#) (decimal[] input, bool[] mask_vector, long nu)
This method predicts the class ID.

Public Attributes

- const long [UNDEFINED_CLASS_ID](#) = -2
Instance variable `UNDEFINED_CLASS_ID` gives the value used for indicating that an input sample was predict to belong to the undefined class; i.e, no class ID was provided for such input samples during training.

Properties

- new decimal [FileFormatVersion](#) [get]
Property `FileFormatVersion` returns the version of the file format used by class [Fast_TopoART_C](#).

Additional Inherited Members

5.7.1 Detailed Description

Class [Fast_TopoART_C](#) provides an implementation of the TopoART-C neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

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

Class [Fast_TopoART_C](#) requires all input except the class IDs to lie in the interval [0,1]. The class IDs are signed integer values.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 [Fast_TopoART_C](#)() [1/2]

```
LibTopoART.Fast_TopoART_C.Fast_TopoART_C (
    long input_dimension,
    long module_number,
    decimal rho_a )
```

This constructor initialises a TopoART-C network.

Parameters

<i>input_dimension</i>	The dimension of input vectors to be learnt.
<i>module_number</i>	The number of TopoART-C modules.
<i>rho_a</i>	The vigilance parameter of the first TopoART-C module (TopoART-C a).

5.7.2.2 Fast_TopoART_C() [2/2]

```
LibTopoART.Fast_TopoART_C.Fast_TopoART_C (
    string path )
```

This constructor loads a saved TopoART-C network.

Parameters

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

5.7.3 Member Function Documentation

5.7.3.1 Learn() [1/2]

```
override void LibTopoART.Fast_TopoART_C.Learn (
    decimal [] input ) [virtual]
```

This method performs a single training step and sets the class ID corresponding to *input* to `UNDEFINED_CLASS_ID`.

Parameters

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

Reimplemented from [LibTopoART.Fast_TopoART](#).

5.7.3.2 Learn() [2/2]

```
void LibTopoART.Fast_TopoART_C.Learn (
    decimal [] input,
    long class_ID )
```

This method performs a single training step.

Parameters

<i>input</i>	The input vector to be learnt.
<i>class_ID</i>	The class ID corresponding to <i>input</i> . (must be equal to or larger than 0)

Implements [LibTopoART.ITopoART_C](#).

5.7.3.3 Predict() [1/2]

```
long LibTopoART.Fast_TopoART_C.Predict (
    decimal [ ] input,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

The predicted class ID.

Implements [LibTopoART.ITopoART_C](#).

5.7.3.4 Predict() [2/2]

```
TopoART_C_Prediction LibTopoART.Fast_TopoART_C.Predict (
    decimal [ ] input,
    bool [ ] mask_vector,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>mask_vector</i>	The mask vector corresponding to <i>input</i> .
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

An object of type [TopoART_C_Prediction](#) containing the predicted class ID and a corresponding confidence value.

Implements [LibTopoART.ITopoART_C](#).

5.7.4 Member Data Documentation**5.7.4.1 UNDEFINED_CLASS_ID**

```
const long LibTopoART.Fast_TopoART_C.UNDEFINED_CLASS_ID = -2
```

Instance variable `UNDEFINED_CLASS_ID` gives the value used for indicating that an input sample was predict to belong to the undefined class; i.e, no class ID was provided for such input samples during training.

5.7.5 Property Documentation**5.7.5.1 FileFormatVersion**

```
new decimal LibTopoART.Fast_TopoART_C.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [Fast_TopoART_C](#).

5.8 LibTopoART.Hypersphere_TopoART Class Reference

Class [Hypersphere_TopoART](#) provides an implementation of the Hypersphere [TopoART](#) neural network as proposed in "Marko Tscherepanow (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."

Inherits [LibTopoART.TopoART](#).

Inherited by [LibTopoART.Hypersphere_TopoART_C](#).

Public Member Functions

- [Hypersphere_TopoART](#) (long input_dimension, long module_number, decimal rho_a)
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, decimal R)
This constructor initialises a Hypersphere [TopoART](#) network.
- [Hypersphere_TopoART](#) (string path)
This constructor loads a saved Hypersphere [TopoART](#) network.

Properties

- new decimal [FileFormatVersion](#) [get]
Property `FileFormatVersion` returns the version of the file format used by class `Hypersphere_TopoART`.
- decimal [HypersphereTopoARTFileFormatVersion](#) [get]
Property `HypersphereTopoARTFileFormatVersion` returns the version of the file format used by class `Hypersphere_TopoART`.
- decimal [R](#) [get]
Property `R` represents the radial extend parameter R .

Additional Inherited Members

5.8.1 Detailed Description

Class [Hypersphere_TopoART](#) provides an implementation of the Hypersphere [TopoART](#) neural network as proposed in "Marko Tscherepanow (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 .

5.8.2 Constructor & Destructor Documentation

5.8.2.1 [Hypersphere_TopoART\(\)](#) [1/3]

```
LibTopoART.Hypersphere_TopoART.Hypersphere_TopoART (
    long input_dimension,
    long module_number,
    decimal rho_a )
```

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

Parameters

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

5.8.2.2 [Hypersphere_TopoART\(\)](#) [2/3]

```
LibTopoART.Hypersphere_TopoART.Hypersphere_TopoART (
    long input_dimension,
    long module_number,
```

```
decimal rho_a,
decimal R )
```

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

Parameters

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

5.8.2.3 Hypersphere_TopoART() [3/3]

```
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 TopoART file.
-------------	--

5.8.3 Property Documentation

5.8.3.1 FileFormatVersion

```
new decimal LibTopoART.Hypersphere_TopoART.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [Hypersphere_TopoART](#).

5.8.3.2 HypersphereTopoARTFileFormatVersion

```
decimal LibTopoART.Hypersphere_TopoART.HypersphereTopoARTFileFormatVersion [get]
```

Property `HypersphereTopoARTFileFormatVersion` returns the version of the file format used by class [Hypersphere_TopoART](#).

5.9 LibTopoART.Hypersphere_TopoART_C Class Reference

Class [Hypersphere_TopoART_C](#) provides an implementation of the Hypersphere TopoART-C neural network. Hypersphere TopoART-C is a combination of Hypersphere [TopoART](#) as proposed in "Marko Tscherepanow (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." and TopoART-C as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

Inherits [LibTopoART.Hypersphere_TopoART](#), and [LibTopoART.ITopoART_C](#).

Public Member Functions

- [Hypersphere_TopoART_C](#) (long input_dimension, long module_number, decimal rho_a)
This constructor initialises a Hypersphere TopoART-C network and sets the radial extend parameter to $\text{Math}.\sqrt{\text{input_dimension}}/2$.
- [Hypersphere_TopoART_C](#) (long input_dimension, long module_number, decimal rho_a, decimal R)
This constructor initialises a Hypersphere TopoART-C network.
- [Hypersphere_TopoART_C](#) (string path)
This constructor loads a saved Hypersphere TopoART-C network.
- override void [Learn](#) (decimal[] input)
This method performs a single training step and sets the class ID corresponding to input to `UNDEFINED_CLASS_ID`.
- void [Learn](#) (decimal[] input, long class_ID)
This method performs a single training step.
- long [Predict](#) (decimal[] input, long nu)
This method predicts the class ID.
- [TopoART_C_Prediction_Predict](#) (decimal[] input, bool[] mask_vector, long nu)
This method predicts the class ID.

Public Attributes

- const long [UNDEFINED_CLASS_ID](#) = -2
Instance variable `UNDEFINED_CLASS_ID` gives the value used for indicating that an input sample was predict to belong to the undefined class; i.e, no class ID was provided for such input samples during training.

Properties

- new decimal [FileFormatVersion](#) [get]
Property `FileFormatVersion` returns the version of the file format used by class [Hypersphere_TopoART_C](#).

Additional Inherited Members

5.9.1 Detailed Description

Class [Hypersphere_TopoART_C](#) provides an implementation of the Hypersphere TopoART-C neural network. Hypersphere TopoART-C is a combination of Hypersphere [TopoART](#) as proposed in "Marko Tscherepanow (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." and TopoART-C as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

In contrast to classes [TopoART_C](#) and [Fast_TopoART_C](#), class [Hypersphere_TopoART_C](#) does not require all input to lie in the interval [0,1]. The input range is controlled by the radial extend parameter R.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 Hypersphere_TopoART_C() [1/3]

```
LibTopoART.Hypersphere_TopoART_C.Hypersphere_TopoART_C (
    long input_dimension,
    long module_number,
    decimal rho_a )
```

This constructor initialises a Hypersphere TopoART-C network and sets the radial extend parameter to $\text{Math}.\sqrt{\text{input_dimension}}/2$.

Parameters

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

5.9.2.2 Hypersphere_TopoART_C() [2/3]

```
LibTopoART.Hypersphere_TopoART_C.Hypersphere_TopoART_C (
    long input_dimension,
    long module_number,
    decimal rho_a,
    decimal R )
```

This constructor initialises a Hypersphere TopoART-C network.

Parameters

<i>input_dimension</i>	The dimension of input vectors to be learnt.
<i>module_number</i>	The number of Hypersphere TopoART-C modules.
<i>rho_a</i>	The vigilance parameter of the first Hypersphere TopoART-C module (HTA-C a).
<i>R</i>	The radial extend parameter.

5.9.2.3 Hypersphere_TopoART_C() [3/3]

```
LibTopoART.Hypersphere_TopoART_C.Hypersphere_TopoART_C (
    string path )
```

This constructor loads a saved Hypersphere TopoART-C network.

Parameters

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

5.9.3 Member Function Documentation

5.9.3.1 Learn() [1/2]

```
override void LibTopoART.Hypersphere_TopoART_C.Learn (
    decimal [ ] input )
```

This method performs a single training step and sets the class ID corresponding to *input* to UNDEFINED_CLASS_ID.

Parameters

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

Implements [LibTopoART.ITopoART](#).

5.9.3.2 Learn() [2/2]

```
void LibTopoART.Hypersphere_TopoART_C.Learn (
    decimal [ ] input,
    long class_ID )
```

This method performs a single training step.

Parameters

<i>input</i>	The input vector to be learnt.
<i>class_ID</i>	The class ID corresponding to <i>input</i> . (must be equal to or larger than 0)

Implements [LibTopoART.ITopoART_C](#).

5.9.3.3 Predict() [1/2]

```
long LibTopoART.Hypersphere_TopoART_C.Predict (
    decimal [ ] input,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

The predicted class ID.

Implements [LibTopoART.ITopoART_C](#).

5.9.3.4 Predict() [2/2]

```
TopoART_C_Prediction LibTopoART.Hypersphere_TopoART_C.Predict (
    decimal [ ] input,
    bool [ ] mask_vector,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>mask_vector</i>	The mask vector corresponding to <i>input</i> .
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

An object of type [TopoART_C_Prediction](#) containing the predicted class ID and a corresponding confidence value.

Implements [LibTopoART.ITopoART_C](#).

5.9.4 Member Data Documentation**5.9.4.1 UNDEFINED_CLASS_ID**

```
const long LibTopoART.Hypersphere_TopoART_C.UNDEFINED_CLASS_ID = -2
```

Instance variable `UNDEFINED_CLASS_ID` gives the value used for indicating that an input sample was predict to belong to the undefined class; i.e, no class ID was provided for such input samples during training.

5.9.5 Property Documentation**5.9.5.1 FileFormatVersion**

```
new decimal LibTopoART.Hypersphere_TopoART_C.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [Hypersphere_TopoART_C](#).

5.10 LibTopoART.IAdaptationStateCheck Interface Reference

Interface enabling checks whether certain adaptations of a network occurred.

Inherited by [LibTopoART.ITopoART](#).

Public Member Functions

- void [ResetAdaptationState](#) ()
This method resets the adaptation state to [AdaptationState.NO_ADAPTATION](#).
- [AdaptationState GetAdaptationState](#) (decimal epsilon=0.001m)
This method returns the current adaptation state.

5.10.1 Detailed Description

Interface enabling checks whether certain adaptations of a network occurred.

5.10.2 Member Function Documentation

5.10.2.1 GetAdaptationState()

```
AdaptationState LibTopoART.IAdaptationStateCheck.GetAdaptationState (
    decimal epsilon = 0.001m )
```

This method returns the current adaptation state.

Parameters

<i>epsilon</i>	The threshold for weight adaptations to be considered.
----------------	--

Returns

An enumeration describing the adaptation state.

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast_TopoART_base](#).

5.10.2.2 ResetAdaptationState()

```
void LibTopoART.IAdaptationStateCheck.ResetAdaptationState ( )
```

This method resets the adaptation state to [AdaptationState.NO_ADAPTATION](#).

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast_TopoART_base](#).

5.11 LibTopoART.IEpisodic_recall Interface Reference

Interface summarising the episodic recall functionality.

Inherited by [LibTopoART.Episodic_TopoART](#).

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.

5.11.1 Detailed Description

Interface summarising the episodic recall functionality.

5.11.2 Member Function Documentation

5.11.2.1 BeginRecall()

```
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](#).

5.11.2.2 EndRecall()

```
void LibTopoART.IEpisodic_recall.EndRecall ( )
```

This method stops the recall process and frees temporary resources.

Implemented in [LibTopoART.Episodic_TopoART](#).

5.11.2.3 InterEpisodeRecallStep()

```
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](#).

5.11.2.4 IntraEpisodeRecallStep()

```
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](#).

5.12 LibTopoART.InvalidClassIDException Class Reference

Exception signalling an invalid class ID.

Inherits Exception.

5.12.1 Detailed Description

Exception signalling an invalid class ID.

5.13 LibTopoART.InvalidFileException Class Reference

Exception signalling an invalid file.

Inherits Exception.

5.13.1 Detailed Description

Exception signalling an invalid file.

5.14 LibTopoART.InvalidStateException Class Reference

Exception signalling an invalid state of the neural network.

Inherits Exception.

5.14.1 Detailed Description

Exception signalling an invalid state of the neural network.

5.15 LibTopoART.ITopoART Interface Reference

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

Inherits [LibTopoART.IAdaptationStateCheck](#).

Inherited by [LibTopoART.Fast_TopopART_base](#), [LibTopoART.ITopoART_C](#), [LibTopoART.ITopoART_R](#), and [LibTopoART.TopoART](#).

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.
- [F2_output \[\] GetBMOutput](#) (decimal[] input, bool[] mask_vector)
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.

5.15.1 Detailed Description

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

5.15.2 Member Function Documentation

5.15.2.1 ComputeClusterIDs()

```
void LibTopoART.ITopoART.ComputeClusterIDs ( )
```

This method computes the cluster IDs for all neurons.

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast_TopoART_base](#).

5.15.2.2 GetBMOutput() [1/2]

```
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 x(t).
--------------	------------------------

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](#).

5.15.2.3 GetBMOutput() [2/2]

```
F2_output [ ] LibTopoART.ITopoART.GetBMOutput (
    decimal [ ] input,
    bool [ ] mask_vector )
```

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

Parameters

<i>input</i>	The input vector x(t).
<i>mask_vector</i>	A mask vector excluding individual dimensions of x(t) from the computation. (Setting an element of the mask vector to <code>true</code> , excludes the corresponding elements of x(t).)

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](#).

5.15.2.4 Learn()

```
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.Fast_TopoART_C](#), [LibTopoART.Episodic_TopoART](#), [LibTopoART.Hypersphere_TopoART_C](#), [LibTopoART.TopoART_R](#), [LibTopoART.Fast_TopoART](#), [LibTopoART.TopoART_C](#), and [LibTopoART.Fast_TopoART_base](#).

5.15.2.5 Save()

```
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 path of the file to save.
-------------	--

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast_TopoART_base](#).

5.15.2.6 SaveText()

```
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 path of the file to save.
-------------	--

Implemented in [LibTopoART.TopoART](#), and [LibTopoART.Fast_TopoART_base](#).

5.15.3 Property Documentation**5.15.3.1 ModuleNum**

```
long LibTopoART.ITopoART.ModuleNum [get]
```

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

5.15.3.2 Phi

```
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.

5.16 LibTopoART.ITopoART_C Interface Reference

Interface summarising the TopoART-C functionality.

Inherits [LibTopoART.ITopoART](#).

Inherited by [LibTopoART.Fast_TopoART_C](#), [LibTopoART.Hypersphere_TopoART_C](#), and [LibTopoART.TopoART_C](#).

Public Member Functions

- void [Learn](#) (decimal[] input, long class_ID)
This method performs a single training step.
- long [Predict](#) (decimal[] input, long nu)
This method predicts the class ID.
- [TopoART_C_Prediction Predict](#) (decimal[] input, bool[] mask_vector, long nu)
This method predicts the class ID.

Additional Inherited Members

5.16.1 Detailed Description

Interface summarising the TopoART-C functionality.

5.16.2 Member Function Documentation

5.16.2.1 Learn()

```
void LibTopoART.ITopoART_C.Learn (
    decimal [] input,
    long class_ID )
```

This method performs a single training step.

Parameters

<i>input</i>	The input vector to be learnt.
<i>class_ID</i>	The class ID corresponding to <i>input</i> .

Implemented in [LibTopoART.Fast_TopoART_C](#), [LibTopoART.Hypersphere_TopoART_C](#), and [LibTopoART.TopoART_C](#).

5.16.2.2 Predict() [1/2]

```
long LibTopoART.ITopoART_C.Predict (
    decimal [] input,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

The predicted class ID.

Implemented in [LibTopoART.Fast_TopoART_C](#), [LibTopoART.Hypersphere_TopoART_C](#), and [LibTopoART.TopoART_C](#).

5.16.2.3 Predict() [2/2]

```
TopoART_C_Prediction LibTopoART.ITopoART_C.Predict (
    decimal [ ] input,
    bool [ ] mask_vector,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>mask_vector</i>	The mask vector corresponding to <i>input</i> .
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

An object of type [TopoART_C_Prediction](#) containing the predicted class ID and a corresponding confidence value.

Implemented in [LibTopoART.Fast_TopoART_C](#), [LibTopoART.Hypersphere_TopoART_C](#), and [LibTopoART.TopoART_C](#).

5.17 LibTopoART.ITopoART_R Interface Reference

Interface summarising the TopoART-R functionality.

Inherits [LibTopoART.ITopoART](#).

Inherited by [LibTopoART.TopoART_R](#).

Public Member Functions

- void [Learn](#) (decimal[] i_vec, decimal[] d_vec)
This method performs a single training step.
- decimal [] [Predict](#) (decimal[] i_vec, long nu=10)
This method predicts the dependent variables.
- [TopoART_R_Prediction Predict](#) (decimal[] i_vec, bool[] m_i_vec, long nu=10)
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.

Additional Inherited Members

5.17.1 Detailed Description

Interface summarising the TopoART-R functionality.

5.17.2 Member Function Documentation

5.17.2.1 Learn()

```
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](#).

5.17.2.2 Predict() [1/2]

```
decimal [ ] LibTopoART.ITopoART_R.Predict (
    decimal [ ] i_vec,
    long nu = 10 )
```

This method predicts the dependent variables.

Parameters

<i>i_vec</i>	The input vector (independent variables).
<i>nu</i>	The maximum cardinality of the neighbourhood set N. (In the original TopoART-R network, nu is fixed to 10. But task-specific adaptations might lead to an improved prediction accuracy. This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

The predicted values for all dependent variables.

Implemented in [LibTopoART.TopoART_R](#).

5.17.2.3 Predict() [2/2]

```
TopoART_R_Prediction LibTopoART.ITopoART_R.Predict (
    decimal [ ] i_vec,
    bool [ ] m_i_vec,
    long nu = 10 )
```

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

<code>i_vec</code>	The input vector (independent variables).
<code>m_i_vec</code>	The mask vector corresponding to <code>i_vec</code> .
<code>nu</code>	The maximum cardinality of the neighbourhood set N. (In the original TopoART-R network, nu is fixed to 10. But task-specific adaptations might lead to an improved prediction accuracy. This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

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`.

5.18 LibTopoART.LibTopoART_info Struct Reference

Struct `LibTopoART_info` provides some metainformation regarding the respective implementation of `LibTopoART`.

Public Attributes

- const decimal `version` = 0.78m
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.

5.18.1 Detailed Description

Struct `LibTopoART_info` provides some metainformation regarding the respective implementation of `LibTopoART`.

5.18.2 Member Data Documentation

5.18.2.1 networks

```
readonly string [] LibTopoART.LibTopoART_info.networks [static]
```

Initial value:

```
= {
    "Episodic TopoART (class Episodic_TopoART)",
    "Hypersphere TopoART (class Hypersphere_TopoART)",
    "Hypersphere TopoART-C (class Hypersphere_TopoART_C)",
    "TopoART (class TopoART, class Fast_TopoART)",
    "TopoART-C (class TopoART_C, class Fast_TopoART_C)",
    "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.

5.18.2.2 UNDEFINED

```
const long LibTopoART.LibTopoART_info.UNDEFINED = -1
```

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

5.18.2.3 version

```
const decimal LibTopoART.LibTopoART_info.version = 0.78m
```

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

5.19 LibTopoART.TopoART Class Reference

Class [TopoART](#) provides an implementation of the [TopoART](#) neural network as proposed in "Marko Tscherepanow (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."

Inherits [LibTopoART.ITopoART](#), and `IDisposable`.

Inherited by [LibTopoART.Hypersphere_TopoART](#), [LibTopoART.TopoART_C](#), and [LibTopoART.TopoART_R](#).

Public Member Functions

- [TopoART](#) (long input_dimension, long module_number, decimal rho_a)
This constructor initialises a [TopoART](#) network.
- [TopoART](#) (string path)
This constructor loads a saved [TopoART](#) network.
- void [Dispose](#) ()
Releases all resources used by the [LibTopoART.TopoART](#) object.
- 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.
- [F2_output](#) [] [GetBMOutput](#) (decimal[] input, bool[] mask_vector)
This method finds the closest category for a given test input.
- virtual 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.
- void [ResetAdaptationState](#) ()
This method resets the adaptation state to [AdaptationState.NO_ADAPTATION](#).
- [AdaptationState](#) [GetAdaptationState](#) (decimal epsilon=0.001m)
This method returns the current adaptation state.

Protected Member Functions

- virtual void [Dispose](#) (bool disposing)
Release resources used by the [LibTopoART.TopoART](#) object.

Properties

- decimal [Alpha](#) [get, set]
Property [Alpha](#) represents the choice parameter alpha.
- decimal [Beta_sbm](#) [get, set]
Property [Beta_sbm](#) represents the learning rate of the second best-matching nodes.
- long [] [ClusterNum](#) [get]
Property [ClusterNum](#) represents the number of [TopoART](#) clusters found by each module.
- long [LearningSteps](#) [get, protected set]
Property [LearningSteps](#) represents the total number of performed learning steps.
- long [ModuleNum](#) [get]
Property [ModuleNum](#) represents the number of [TopoART](#) modules used. (The original [TopoART](#) uses two modules.)
- long [] [NodeNum](#) [get]
Property [NodeNum](#) represents the number of [TopoART](#) nodes used by each module.
- 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 [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 τ required for the removal of nodes and edges.

- string `IntegerBaseType` [get]

Property `IntegerBaseType` returns a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.

- decimal `FileFormatVersion` [get]

Property `FileFormatVersion` returns the version of the file format used by class `TopoART`.

- string `FloatBaseType` [get]

Property `FloatBaseType` returns a string containing the data type used for representing floating point variables (input, weights, etc.) internally.

- decimal `TopoARTFileFormatVersion` [get]

Property `TopoARTFileFormatVersion` returns the version of the file format used by class `TopoART`.

5.19.1 Detailed Description

Class `TopoART` provides an implementation of the `TopoART` neural network as proposed in "Marko Tscherepanow (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 $[0,1]$.

5.19.2 Constructor & Destructor Documentation

5.19.2.1 `TopoART()` [1/2]

```
LibTopoART.TopoART.TopoART (
    long input_dimension,
    long module_number,
    decimal rho_a )
```

This constructor initialises a `TopoART` network.

Parameters

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

5.19.2.2 `TopoART()` [2/2]

```
LibTopoART.TopoART.TopoART (
    string path )
```

This constructor loads a saved `TopoART` network.

Parameters

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

5.19.3 Member Function Documentation

5.19.3.1 ComputeClusterIDs()

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

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

5.19.3.2 Dispose() [1/2]

```
void LibTopoART.TopoART.Dispose ( )
```

Releases all resources used by the [LibTopoART.TopoART](#) object.

Call [Dispose\(\)](#) when you are finished using the [LibTopoART.TopoART](#). The [Dispose\(\)](#) method leaves the [LibTopoART.TopoART](#) in an unusable state. After calling [Dispose\(\)](#), you must release all references to the [LibTopoART.TopoART](#) so the garbage collector can reclaim the memory that the [LibTopoART.TopoART](#) was occupying.

5.19.3.3 Dispose() [2/2]

```
virtual void LibTopoART.TopoART.Dispose (
    bool disposing ) [protected], [virtual]
```

Release resources used by the [LibTopoART.TopoART](#) object.

Parameters

<i>disposing</i>	If set to <code>true</code> all managed resources are released.
------------------	---

5.19.3.4 GetAdaptationState()

```
AdaptationState LibTopoART.TopoART.GetAdaptationState (
    decimal epsilon = 0.001m )
```

This method returns the current adaptation state.

Parameters

<i>epsilon</i>	The threshold for weight adaptations to be considered.
----------------	--

Returns

An enumeration describing the adaptation state.

Implements [LibTopoART.IAdaptationStateCheck](#).

5.19.3.5 GetBMOutput() [1/2]

```
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 x(t).
--------------	------------------------

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](#).

5.19.3.6 GetBMOutput() [2/2]

```
F2_output [ ] LibTopoART.TopoART.GetBMOutput (
    decimal [ ] input,
    bool [ ] mask_vector )
```

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

Parameters

<i>input</i>	The input vector x(t).
<i>mask_vector</i>	A mask vector excluding individual dimensions of x(t) from the computation. (Setting an element of the mask vector to <code>true</code> , excludes the corresponding elements of x(t).)

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](#).

5.19.3.7 Learn()

```
virtual void LibTopoART.TopoART.Learn (
    decimal [] input ) [virtual]
```

This method performs a single training step.

Parameters

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

Implements [LibTopoART.ITopoART](#).

Reimplemented in [LibTopoART.Hypersphere_TopopART_C](#), [LibTopoART.TopoART_R](#), and [LibTopoART.TopoART_C](#).

5.19.3.8 ResetAdaptationState()

```
void LibTopoART.TopoART.ResetAdaptationState ( )
```

This method resets the adaptation state to [AdaptationState.NO_ADAPTATION](#).

Implements [LibTopoART.IAdaptationStateCheck](#).

5.19.3.9 Save()

```
void LibTopoART.TopoART.Save (
    string path )
```

This method saves the entire network as a binary file.

Parameters

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

Implements [LibTopoART.ITopoART](#).

5.19.3.10 SaveText()

```
void LibTopoART.TopoART.SaveText (
    string path )
```

This method saves the entire network as a text file.

Parameters

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

Implements [LibTopoART.ITopoART](#).

5.19.4 Property Documentation**5.19.4.1 FileFormatVersion**

```
decimal LibTopoART.TopoART.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [TopoART](#).

5.19.4.2 FloatBaseType

```
string LibTopoART.TopoART.FloatBaseType [get]
```

Property `FloatBaseType` returns a string containing the data type used for representing floating point variables (input, weights, etc.) internally.

5.19.4.3 IntegerBaseType

```
string LibTopoART.TopoART.IntegerBaseType [get]
```

Property `IntegerBaseType` returns a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.

5.19.4.4 TopoARTFileFormatVersion

```
decimal LibTopoART.TopoART.TopoARTFileFormatVersion [get]
```

Property `TopoARTFileFormatVersion` returns the version of the file format used by class [TopoART](#).

5.20 LibTopoART.TopoART_C Class Reference

Class [TopoART_C](#) provides an implementation of the TopoART-C neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

Inherits [LibTopoART.TopoART](#), and [LibTopoART.ITopoART_C](#).

Public Member Functions

- [TopoART_C](#) (long input_dimension, long module_number, decimal rho_a)
This constructor initialises a TopoART-C network.
- [TopoART_C](#) (string path)
This constructor loads a saved TopoART-C network.
- override void [Learn](#) (decimal[] input)
This method performs a single training step and sets the class ID corresponding to input to `UNDEFINED_CLASS_ID`.
- void [Learn](#) (decimal[] input, long class_ID)
This method performs a single training step.
- long [Predict](#) (decimal[] input, long nu)
This method predicts the class ID.
- [TopoART_C_Prediction Predict](#) (decimal[] input, bool[] mask_vector, long nu)
This method predicts the class ID.

Public Attributes

- const long [UNDEFINED_CLASS_ID](#) = -2
Instance variable `UNDEFINED_CLASS_ID` gives the value used for indicating that an input sample was predict to belong to the undefined class; i.e, no class ID was provided for such input samples during training.

Properties

- new decimal [FileFormatVersion](#) [get]
Property `FileFormatVersion` returns the version of the file format used by class `TopoART_C`.

Additional Inherited Members

5.20.1 Detailed Description

Class [TopoART_C](#) provides an implementation of the TopoART-C neural network as proposed in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France."

Class [TopoART_C](#) requires all input except the class IDs to lie in the interval [0,1]. The class IDs are signed integer values.

5.20.2 Constructor & Destructor Documentation

5.20.2.1 TopoART_C() [1/2]

```
LibTopoART.TopoART_C.TopoART_C (
    long input_dimension,
    long module_number,
    decimal rho_a )
```

This constructor initialises a TopoART-C network.

Parameters

<i>input_dimension</i>	The dimension of input vectors to be learnt.
<i>module_number</i>	The number of TopoART-C modules.
<i>rho_a</i>	The vigilance parameter of the first TopoART-C module (TopoART-C a).

5.20.2.2 TopoART_C() [2/2]

```
LibTopoART.TopoART_C.TopoART_C (
    string path )
```

This constructor loads a saved TopoART-C network.

Parameters

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

5.20.3 Member Function Documentation**5.20.3.1 Learn()** [1/2]

```
override void LibTopoART.TopoART_C.Learn (
    decimal [] input ) [virtual]
```

This method performs a single training step and sets the class ID corresponding to *input* to `UNDEFINED_CLASS_ID`.

Parameters

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

Reimplemented from [LibTopoART.TopoART](#).

5.20.3.2 Learn() [2/2]

```
void LibTopoART.TopoART_C.Learn (
    decimal [] input,
    long class_ID )
```

This method performs a single training step.

Parameters

<i>input</i>	The input vector to be learnt.
<i>class_ID</i>	The class ID corresponding to <i>input</i> . (must be equal to or larger than 0)

Implements [LibTopoART.ITopoART_C](#).

5.20.3.3 Predict() [1/2]

```
long LibTopoART.TopoART_C.Predict (
    decimal [ ] input,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

The predicted class ID.

Implements [LibTopoART.ITopoART_C](#).

5.20.3.4 Predict() [2/2]

```
TopoART_C_Prediction LibTopoART.TopoART_C.Predict (
    decimal [ ] input,
    bool [ ] mask_vector,
    long nu )
```

This method predicts the class ID.

Parameters

<i>input</i>	The input vector the class ID of which is to be predicted.
<i>mask_vector</i>	The mask vector corresponding to <i>input</i> .
<i>nu</i>	The maximum cardinality of the set of enclosing categories E and the neighbourhood set N. (This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

An object of type [TopoART_C_Prediction](#) containing the predicted class ID and a corresponding confidence value.

Implements [LibTopoART.ITopoART_C](#).

5.20.4 Member Data Documentation**5.20.4.1 UNDEFINED_CLASS_ID**

```
const long LibTopoART.TopoART_C.UNDEFINED_CLASS_ID = -2
```

Instance variable `UNDEFINED_CLASS_ID` gives the value used for indicating that an input sample was predicted to belong to the undefined class; i.e, no class ID was provided for such input samples during training.

5.20.5 Property Documentation**5.20.5.1 FileFormatVersion**

```
new decimal LibTopoART.TopoART_C.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [TopoART_C](#).

5.21 LibTopoART.TopoART_C_Prediction Struct Reference

Struct [TopoART_C_Prediction](#) contains a prediction made by a TopoART-C network.

Public Member Functions

- [TopoART_C_Prediction](#) (long `class_ID`, decimal `confidence`)
This constructor sets the instance variables `class_ID` and `confidence` of struct [TopoART_C_Prediction](#).

Public Attributes

- long `class_ID`
Instance variable `class_ID` gives the predicted class ID.
- decimal `confidence`
Instance variable `confidence` provides a confidence for the predicted class ID.

5.21.1 Detailed Description

Struct [TopoART_C_Prediction](#) contains a prediction made by a TopoART-C network.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 TopoART_C_Prediction()

```
LibTopoART.TopoART_C_Prediction.TopoART_C_Prediction (
    long class_ID,
    decimal confidence )
```

This constructor sets the instance variables `class_ID` and `confidence` of struct [TopoART_C_Prediction](#).

Parameters

<i>class_ID</i>	The class ID to be set.
<i>confidence</i>	The value of the confidence to be set.

5.21.3 Member Data Documentation**5.21.3.1 class_ID**

```
long LibTopoART.TopoART_C_Prediction.class_ID
```

Instance variable `class_ID` gives the predicted class ID.

5.21.3.2 confidence

```
decimal LibTopoART.TopoART_C_Prediction.confidence
```

Instance variable `confidence` provides a confidence for the predicted class ID.

5.22 LibTopoART_samples.TopoART_C_sample1 Class Reference

Simple classification sample.

5.22.1 Detailed Description

Simple classification sample.

This sample demonstrates training and several possibilities for prediction at the example of a simple classification task.

5.23 LibTopoART_samples.TopoART_C_sample2 Class Reference

Classification sample using more complex synthetic two-dimensional data with associated class IDs.

5.23.1 Detailed Description

Classification sample using more complex synthetic two-dimensional data with associated class IDs.

Train TopoART-C with a two-dimensional dataset similar to the one used in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France." This dataset comprises six clusters (each containing 15,000 samples) as well as 10,000 noise samples which were mixed randomly. The samples are divided into two classes. Each sample belonging to one of the six clusters is assigned a class ID depending on its position. In contrast, noise samples receive a random class ID.

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

5.24 LibTopoART.TopoART_R Class Reference

Class [TopoART_R](#) provides an implementation of the TopoART-R neural network as proposed in "Marko Tscherepanow (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."

Inherits [LibTopoART.TopoART](#), and [LibTopoART.ITopoART_R](#).

Public Member Functions

- [TopoART_R](#) (long i_dimension, long d_dimension, long module_number, decimal rho_a)
This constructor initialises a TopoART-R network.
- [TopoART_R](#) (string path)
This constructor loads a saved TopoART-R network.
- override void [Learn](#) (decimal[] input)
This method performs a single training step. The independent variables and the dependent variables are automatically separated.
- void [Learn](#) (decimal[] i_vec, decimal[] d_vec)
This method performs a single training step.
- decimal[] [Predict](#) (decimal[] i_vec, long nu=10)
This method predicts the dependent variables.
- [TopoART_R_Prediction Predict](#) (decimal[] i_vec, bool[] m_i_vec, long nu=10)
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.

Properties

- new decimal [FileFormatVersion](#) [get]
Property FileFormatVersion returns the version of the file format used by class [TopoART_R](#).

Additional Inherited Members

5.24.1 Detailed Description

Class [TopoART_R](#) provides an implementation of the TopoART-R neural network as proposed in "Marko Tscherepanow (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].

5.24.2 Constructor & Destructor Documentation

5.24.2.1 TopoART_R() [1/2]

```
LibTopoART.TopoART_R.TopoART_R (
    long i_dimension,
    long d_dimension,
    long module_number,
    decimal rho_a )
```

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</i>	The vigilance parameter of the first TopoART-R module (TopoART-R a).

5.24.2.2 TopoART_R() [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.
-------------	--------------------------------------

5.24.3 Member Function Documentation**5.24.3.1 Learn()** [1/2]

```
override void LibTopoART.TopoART_R.Learn (
    decimal [] input ) [virtual]
```

This method performs a single training step. The independent variables and the dependent variables are automatically separated.

Parameters

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

Reimplemented from [LibTopoART.TopoART](#).

5.24.3.2 Learn() [2/2]

```
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](#).

5.24.3.3 Predict() [1/2]

```
decimal [ ] LibTopoART.TopoART_R.Predict (
    decimal [ ] i_vec,
    long nu = 10 )
```

This method predicts the dependent variables.

Parameters

<i>i_vec</i>	The input vector (independent variables).
<i>nu</i>	The maximum cardinality of the neighbourhood set N. (In the original TopoART-R network, nu is fixed to 10. But task-specific adaptations might lead to an improved prediction accuracy. This parameter does not modify the network. It may be arbitrarily changed in each prediction step.)

Returns

The predicted values for all dependent variables.

Implements [LibTopoART.ITopoART_R](#).

5.24.3.4 Predict() [2/2]

```
TopoART_R_Prediction LibTopoART.TopoART_R.Predict (
    decimal [ ] i_vec,
    bool [ ] m_i_vec,
    long nu = 10 )
```

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> .
<i>nu</i>	The maximum cardinality of the neighbourhood set N. (In the original TopoART-R network, nu is fixed to 10. But task-specific adaptations might lead to an improved prediction accuracy. This parameter does not alter the network. It may be arbitrarily changed in each prediction step.)

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](#).

5.24.4 Property Documentation**5.24.4.1 FileFormatVersion**

```
new decimal LibTopoART.TopoART_R.FileFormatVersion [get]
```

Property `FileFormatVersion` returns the version of the file format used by class [TopoART_R](#).

5.25 LibTopoART.TopoART_R_Prediction Struct Reference

Struct [TopoART_R_Prediction](#) 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 struct [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.
- decimal[] [i_vec_prediction](#)
Instance variable [i_vec_prediction](#) represents predictions for unknown independent variables.
- decimal[] [d_vec_prediction](#)
Instance variable [d_vec_prediction](#) provides the predictions for the dependent variables.

5.25.1 Detailed Description

Struct [TopoART_R_Prediction](#) contains a prediction made by a TopoART-R network.

5.25.2 Constructor & Destructor Documentation**5.25.2.1 TopoART_R_Prediction()**

```
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 struct [TopoART_R_Prediction](#).

Parameters

<i>i_vec_prediction</i>	The prediction results for the independent variables to be set.
<i>d_vec_prediction</i>	The prediction results for the dependent variables to be set.

5.25.3 Member Function Documentation

5.25.3.1 PrintPredictions()

```
void LibTopoART.TopoART_R_Prediction.PrintPredictions ( )
```

This method prints the predictions on the console.

5.25.4 Member Data Documentation

5.25.4.1 d_vec_prediction

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

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

5.25.4.2 i_vec_prediction

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

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

5.25.4.3 NO_PREDICTION

```
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.

5.26 LibTopoART_samples.TopoART_R_sample1 Class Reference

Regression sample using TopoART-R. (simplified version)

5.26.1 Detailed Description

Regression sample using TopoART-R. (simplified version)

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`.

5.27 LibTopoART_samples.TopoART_R_sample2 Class Reference

Regression sample using TopoART-R. (advanced version)

5.27.1 Detailed Description

Regression sample using TopoART-R. (advanced version)

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`.

5.28 LibTopoART_samples.TopoART_sample1 Class Reference

Simple clustering sample.

5.28.1 Detailed Description

Simple clustering 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.

5.29 LibTopoART_samples.TopoART_sample2 Class Reference

Clustering sample using more complex synthetic two-dimensional data.

5.29.1 Detailed Description

Clustering sample using more complex synthetic two-dimensional data.

Train TopoART or Hypersphere TopoART with a two-dimensional dataset similar to the one used in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 18-23. Montpellier, France." 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

AdaptationState
 LibTopoART, 7

BeginRecall
 LibTopoART::Episodic_TopoART, 9
 LibTopoART::IEpisodic_recall, 33

bm_cluster_ID
 LibTopoART::F2_output, 13

bm_node_ID
 LibTopoART::F2_output, 13

bm_node_activation
 LibTopoART::F2_output, 13

bm_permanent_cluster_ID
 LibTopoART::F2_output, 14

bm_permanent_node_ID
 LibTopoART::F2_output, 14

bm_permanent_node_activation
 LibTopoART::F2_output, 14

class_ID
 LibTopoART::TopoART_C_Prediction, 56

ComputeClusterIDs
 LibTopoART::Fast_TopoART_base, 17
 LibTopoART::ITopoART, 36
 LibTopoART::TopoART, 47

confidence
 LibTopoART::TopoART_C_Prediction, 56

d_vec_prediction
 LibTopoART::TopoART_R_Prediction, 61

Dispose
 LibTopoART::Fast_TopoART_base, 17, 18
 LibTopoART::TopoART, 47

EndRecall
 LibTopoART::Episodic_TopoART, 10
 LibTopoART::IEpisodic_recall, 33

Episodic_TopoART
 LibTopoART::Episodic_TopoART, 9

F2_output
 LibTopoART::F2_output, 13

Fast_TopoART_C
 LibTopoART::Fast_TopoART_C, 22, 23

Fast_TopoART
 LibTopoART::Fast_TopoART, 15

FileFormatVersion
 LibTopoART::Episodic_TopoART, 11
 LibTopoART::Fast_TopoART_base, 21
 LibTopoART::Fast_TopoART_C, 25
 LibTopoART::Hypersphere_TopoART_C, 31
 LibTopoART::Hypersphere_TopoART, 27
 LibTopoART::TopoART_C, 54
 LibTopoART::TopoART_R, 60
 LibTopoART::TopoART, 50

FloatBaseType

LibTopoART::Fast_TopoART_base, 21
LibTopoART::TopoART, 50

GetAdaptationState
 LibTopoART::Fast_TopoART_base, 18
 LibTopoART::IAdaptationStateCheck, 32
 LibTopoART::TopoART, 47

GetBMOutput
 LibTopoART::Fast_TopoART_base, 18, 19
 LibTopoART::ITopoART, 36, 37
 LibTopoART::TopoART, 48

Hypersphere_TopoART_C
 LibTopoART::Hypersphere_TopoART_C, 28, 29

Hypersphere_TopoART
 LibTopoART::Hypersphere_TopoART, 26, 27

HypersphereTopoARTFileFormatVersion
 LibTopoART::Hypersphere_TopoART, 27

i_vec_prediction
 LibTopoART::TopoART_R_Prediction, 61

IntegerBaseType
 LibTopoART::Fast_TopoART_base, 21
 LibTopoART::TopoART, 50

InterEpisodeRecallStep
 LibTopoART::Episodic_TopoART, 10
 LibTopoART::IEpisodic_recall, 33

IntraEpisodeRecallStep
 LibTopoART::Episodic_TopoART, 10
 LibTopoART::IEpisodic_recall, 34

Learn
 LibTopoART::Episodic_TopoART, 11
 LibTopoART::Fast_TopoART_base, 19
 LibTopoART::Fast_TopoART_C, 23
 LibTopoART::Fast_TopoART, 15
 LibTopoART::Hypersphere_TopoART_C, 30
 LibTopoART::ITopoART_C, 40
 LibTopoART::ITopoART_R, 42
 LibTopoART::ITopoART, 37
 LibTopoART::TopoART_C, 52
 LibTopoART::TopoART_R, 58
 LibTopoART::TopoART, 49

LibTopoART.Episodic_TopoART, 8

LibTopoART.F2_output, 12

LibTopoART.Fast_TopoART_base, 16

LibTopoART.Fast_TopoART_C, 21

LibTopoART.Fast_TopoART, 14

LibTopoART.Hypersphere_TopoART_C, 27

LibTopoART.Hypersphere_TopoART, 25

LibTopoART.IAdaptationStateCheck, 32

LibTopoART.IEpisodic_recall, 33

LibTopoART.ITopoART_C, 39

LibTopoART.ITopoART_R, 41

LibTopoART.ITopoART, 35

LibTopoART.InvalidClassIDException, 34

- LibTopoART.InvalidFileException, 35
- LibTopoART.InvalidStateException, 35
- LibTopoART.LibTopoART_info, 43
- LibTopoART.TopoART_C_Prediction, 54
- LibTopoART.TopoART_R_Prediction, 60
- LibTopoART.TopoART_C, 50
- LibTopoART.TopoART_R, 57
- LibTopoART.TopoART, 44
- LibTopoART::Episodic_TopoART
 - BeginRecall, 9
 - EndRecall, 10
 - Episodic_TopoART, 9
 - FileFormatVersion, 11
 - InterEpisodeRecallStep, 10
 - IntraEpisodeRecallStep, 10
 - Learn, 11
- LibTopoART::F2_output
 - bm_cluster_ID, 13
 - bm_node_ID, 13
 - bm_node_activation, 13
 - bm_permanent_cluster_ID, 14
 - bm_permanent_node_ID, 14
 - bm_permanent_node_activation, 14
 - F2_output, 13
- LibTopoART::Fast_TopoART_base
 - ComputeClusterIDs, 17
 - Dispose, 17, 18
 - FileFormatVersion, 21
 - FloatBaseType, 21
 - GetAdaptationState, 18
 - GetBMOutput, 18, 19
 - IntegerBaseType, 21
 - Learn, 19
 - ResetAdaptationState, 19
 - Save, 20
 - SaveText, 20
 - TopoARTFileFormatVersion, 21
- LibTopoART::Fast_TopoART_C
 - Fast_TopoART_C, 22, 23
 - FileFormatVersion, 25
 - Learn, 23
 - Predict, 24
 - UNDEFINED_CLASS_ID, 25
- LibTopoART::Fast_TopoART
 - Fast_TopoART, 15
 - Learn, 15
- LibTopoART::Hypersphere_TopoART_C
 - FileFormatVersion, 31
 - Hypersphere_TopoART_C, 28, 29
 - Learn, 30
 - Predict, 30, 31
 - UNDEFINED_CLASS_ID, 31
- LibTopoART::Hypersphere_TopoART
 - FileFormatVersion, 27
 - Hypersphere_TopoART, 26, 27
 - HypersphereTopoARTFileFormatVersion, 27
- LibTopoART::IAdaptationStateCheck
 - GetAdaptationState, 32
 - ResetAdaptationState, 32
- LibTopoART::IEpisodic_recall
 - BeginRecall, 33
 - EndRecall, 33
 - InterEpisodeRecallStep, 33
 - IntraEpisodeRecallStep, 34
- LibTopoART::ITopoART_C
 - Learn, 40
 - Predict, 40, 41
- LibTopoART::ITopoART_R
 - Learn, 42
 - Predict, 42
- LibTopoART::ITopoART
 - ComputeClusterIDs, 36
 - GetBMOutput, 36, 37
 - Learn, 37
 - ModuleNum, 39
 - Phi, 39
 - Save, 37
 - SaveText, 39
- LibTopoART::LibTopoART_info
 - networks, 44
 - UNDEFINED, 44
 - version, 44
- LibTopoART::TopoART_C_Prediction
 - class_ID, 56
 - confidence, 56
 - TopoART_C_Prediction, 55
- LibTopoART::TopoART_R_Prediction
 - d_vec_prediction, 61
 - i_vec_prediction, 61
 - NO_PREDICTION, 61
 - PrintPredictions, 61
 - TopoART_R_Prediction, 60
- LibTopoART::TopoART_C
 - FileFormatVersion, 54
 - Learn, 52
 - Predict, 53
 - TopoART_C, 51, 52
 - UNDEFINED_CLASS_ID, 54
- LibTopoART::TopoART_R
 - FileFormatVersion, 60
 - Learn, 58
 - Predict, 59
 - TopoART_R, 57, 58
- LibTopoART::TopoART
 - ComputeClusterIDs, 47
 - Dispose, 47
 - FileFormatVersion, 50
 - FloatBaseType, 50
 - GetAdaptationState, 47
 - GetBMOutput, 48
 - IntegerBaseType, 50
 - Learn, 49
 - ResetAdaptationState, 49
 - Save, 49
 - SaveText, 49
 - TopoARTFileFormatVersion, 50

- TopoART, [46](#)
- LibTopoART_samples, [7](#)
- LibTopoART_samples.Episodic_TopopART_sample1, [11](#)
- LibTopoART_samples.Episodic_TopopART_sample2, [12](#)
- LibTopoART_samples.TopoART_C_sample1, [56](#)
- LibTopoART_samples.TopoART_C_sample2, [56](#)
- LibTopoART_samples.TopoART_R_sample1, [61](#)
- LibTopoART_samples.TopoART_R_sample2, [62](#)
- LibTopoART_samples.TopoART_sample1, [62](#)
- LibTopoART_samples.TopoART_sample2, [62](#)
- LibTopoART, [5](#)
 - AdaptationState, [7](#)
- ModuleNum
 - LibTopoART::ITopoART, [39](#)
- NO_PREDICTION
 - LibTopoART::TopoART_R_Prediction, [61](#)
- networks
 - LibTopoART::LibTopoART_info, [44](#)
- Phi
 - LibTopoART::ITopoART, [39](#)
- Predict
 - LibTopoART::Fast_TopopART_C, [24](#)
 - LibTopoART::Hypersphere_TopopART_C, [30](#), [31](#)
 - LibTopoART::ITopoART_C, [40](#), [41](#)
 - LibTopoART::ITopoART_R, [42](#)
 - LibTopoART::TopoART_C, [53](#)
 - LibTopoART::TopoART_R, [59](#)
- PrintPredictions
 - LibTopoART::TopoART_R_Prediction, [61](#)
- ResetAdaptationState
 - LibTopoART::Fast_TopopART_base, [19](#)
 - LibTopoART::IAdaptationStateCheck, [32](#)
 - LibTopoART::TopoART, [49](#)
- Save
 - LibTopoART::Fast_TopopART_base, [20](#)
 - LibTopoART::ITopoART, [37](#)
 - LibTopoART::TopoART, [49](#)
- SaveText
 - LibTopoART::Fast_TopopART_base, [20](#)
 - LibTopoART::ITopoART, [39](#)
 - LibTopoART::TopoART, [49](#)
- TopoART_C_Prediction
 - LibTopoART::TopoART_C_Prediction, [55](#)
- TopoART_R_Prediction
 - LibTopoART::TopoART_R_Prediction, [60](#)
- TopoART_C
 - LibTopoART::TopoART_C, [51](#), [52](#)
- TopoART_R
 - LibTopoART::TopoART_R, [57](#), [58](#)
- TopoARTFileFormatVersion
 - LibTopoART::Fast_TopopART_base, [21](#)
 - LibTopoART::TopoART, [50](#)
- TopoART
 - LibTopoART::TopoART, [46](#)
- UNDEFINED_CLASS_ID
 - LibTopoART::Fast_TopopART_C, [25](#)
 - LibTopoART::Hypersphere_TopopART_C, [31](#)
 - LibTopoART::TopoART_C, [54](#)
- UNDEFINED
 - LibTopoART::LibTopoART_info, [44](#)
- version
 - LibTopoART::LibTopoART_info, [44](#)