

LibTopoART
v0.91

Generated by Doxygen 1.8.16

Sat Dec 21 2019 13:05:03

1 Namespace Index	1
1.1 Packages	1
2 Hierarchical Index	2
2.1 Class Hierarchy	2
3 Class Index	4
3.1 Class List	4
4 Namespace Documentation	7
4.1 LibTopoART Namespace Reference	7
4.1.1 Enumeration Type Documentation	9
4.2 LibTopoART_samples Namespace Reference	9
5 Class Documentation	10
5.1 LibTopoART.CategoryInfo Struct Reference	10
5.1.1 Detailed Description	10
5.1.2 Constructor & Destructor Documentation	11
5.1.3 Member Data Documentation	11
5.2 LibTopoART.Episodic_TopoART Class Reference	12
5.2.1 Detailed Description	12
5.3 LibTopoART_samples.Episodic_TopoART_sample1 Class Reference	12
5.3.1 Detailed Description	12
5.4 LibTopoART_samples.Episodic_TopoART_sample2 Class Reference	12
5.4.1 Detailed Description	12
5.5 LibTopoART.F2_output Class Reference	12
5.5.1 Detailed Description	13
5.5.2 Constructor & Destructor Documentation	13
5.5.3 Member Data Documentation	13
5.6 LibTopoART.Fast_Episodic_TopoART Class Reference	14
5.6.1 Detailed Description	16
5.6.2 Constructor & Destructor Documentation	16
5.6.3 Member Function Documentation	16
5.6.4 Property Documentation	18
5.7 LibTopoART.Fast_TopoART Class Reference	18
5.7.1 Detailed Description	20
5.7.2 Constructor & Destructor Documentation	20
5.7.3 Member Function Documentation	21
5.8 LibTopoART.Fast_TopoART_AM Class Reference	21
5.8.1 Detailed Description	23
5.8.2 Constructor & Destructor Documentation	23
5.8.3 Member Function Documentation	23
5.8.4 Property Documentation	26
5.9 LibTopoART.Fast_TopoART_base Class Reference	26

5.9.1 Detailed Description	28
5.9.2 Member Function Documentation	28
5.9.3 Member Data Documentation	32
5.9.4 Property Documentation	32
5.10 LibTopoART.Fast_TopoART_C Class Reference	33
5.10.1 Detailed Description	35
5.10.2 Constructor & Destructor Documentation	35
5.10.3 Member Function Documentation	36
5.10.4 Member Data Documentation	38
5.10.5 Property Documentation	38
5.11 LibTopoART.Fast_TopoART_R Class Reference	38
5.11.1 Detailed Description	40
5.11.2 Constructor & Destructor Documentation	40
5.11.3 Member Function Documentation	41
5.11.4 Property Documentation	43
5.12 LibTopoART.Hypersphere_TopoART Class Reference	43
5.12.1 Detailed Description	44
5.12.2 Constructor & Destructor Documentation	45
5.12.3 Property Documentation	46
5.13 LibTopoART.Hypersphere_TopoART_C Class Reference	46
5.13.1 Detailed Description	48
5.13.2 Constructor & Destructor Documentation	48
5.13.3 Member Function Documentation	49
5.13.4 Member Data Documentation	51
5.13.5 Property Documentation	51
5.14 LibTopoART.IAdaptationStateCheck Interface Reference	51
5.14.1 Detailed Description	52
5.14.2 Member Function Documentation	52
5.15 LibTopoART.IAssociative_recall Interface Reference	52
5.15.1 Detailed Description	53
5.15.2 Member Function Documentation	53
5.16 LibTopoART.ICategoryAccess Interface Reference	55
5.16.1 Detailed Description	55
5.16.2 Member Function Documentation	55
5.17 LibTopoART.IEpisodic_recall Interface Reference	56
5.17.1 Detailed Description	56
5.17.2 Member Function Documentation	56
5.18 LibTopoART.InvalidClassIDException Class Reference	58
5.18.1 Detailed Description	58
5.19 LibTopoART.InvalidFileException Class Reference	58
5.19.1 Detailed Description	59
5.20 LibTopoART.InvalidModuleIndexException Class Reference	59

5.20.1 Detailed Description	60
5.21 LibTopoART.InvalidNodeNumberException Class Reference	60
5.21.1 Detailed Description	61
5.22 LibTopoART.InvalidSizeException Class Reference	61
5.22.1 Detailed Description	62
5.23 LibTopoART.InvalidStateException Class Reference	62
5.23.1 Detailed Description	63
5.24 LibTopoART.ITopoART Interface Reference	63
5.24.1 Detailed Description	64
5.24.2 Member Function Documentation	64
5.24.3 Property Documentation	66
5.25 LibTopoART.ITopoART_AM Interface Reference	67
5.25.1 Detailed Description	68
5.25.2 Member Function Documentation	68
5.25.3 Property Documentation	69
5.26 LibTopoART.ITopoART_C Interface Reference	69
5.26.1 Detailed Description	70
5.26.2 Member Function Documentation	70
5.27 LibTopoART.ITopoART_R Interface Reference	71
5.27.1 Detailed Description	72
5.27.2 Member Function Documentation	72
5.27.3 Property Documentation	74
5.28 LibTopoART.LibTopoART_info Struct Reference	74
5.28.1 Detailed Description	74
5.28.2 Member Data Documentation	74
5.29 LibTopoART.TopoART Class Reference	75
5.29.1 Detailed Description	77
5.29.2 Constructor & Destructor Documentation	78
5.29.3 Member Function Documentation	78
5.29.4 Member Data Documentation	82
5.29.5 Property Documentation	82
5.30 LibTopoART.TopoART_AM Class Reference	83
5.30.1 Detailed Description	83
5.31 LibTopoART.samples.TopoART_AM_sample1 Class Reference	83
5.31.1 Detailed Description	83
5.32 LibTopoART.TopoART_C Class Reference	83
5.32.1 Detailed Description	85
5.32.2 Constructor & Destructor Documentation	85
5.32.3 Member Function Documentation	86
5.32.4 Member Data Documentation	87
5.32.5 Property Documentation	88
5.33 LibTopoART.TopoART_C_Prediction Struct Reference	88

5.33.1 Detailed Description	88
5.33.2 Constructor & Destructor Documentation	88
5.33.3 Member Data Documentation	89
5.34 LibTopoART_samples.TopoART_C_sample1 Class Reference	89
5.34.1 Detailed Description	89
5.35 LibTopoART_samples.TopoART_C_sample2 Class Reference	89
5.35.1 Detailed Description	89
5.36 LibTopoART.TopoART_R Class Reference	90
5.36.1 Detailed Description	91
5.36.2 Constructor & Destructor Documentation	91
5.36.3 Member Function Documentation	92
5.36.4 Property Documentation	94
5.37 LibTopoART.TopoART_R_Prediction Struct Reference	94
5.37.1 Detailed Description	95
5.37.2 Constructor & Destructor Documentation	95
5.37.3 Member Function Documentation	95
5.37.4 Member Data Documentation	95
5.38 LibTopoART_samples.TopoART_R_sample1 Class Reference	96
5.38.1 Detailed Description	96
5.39 LibTopoART_samples.TopoART_R_sample2 Class Reference	96
5.39.1 Detailed Description	96
5.40 LibTopoART_samples.TopoART_R_sample3 Class Reference	96
5.40.1 Detailed Description	96
5.41 LibTopoART_samples.TopoART_sample1 Class Reference	97
5.41.1 Detailed Description	97
5.42 LibTopoART_samples.TopoART_sample2 Class Reference	97
5.42.1 Detailed Description	97
5.43 LibTopoART_samples.TopoART_sample3 Class Reference	97
5.43.1 Detailed Description	97
Index	99

1 Namespace Index

1.1 Packages

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

LibTopoART	7
LibTopoART_samples	9

2 Hierarchical Index

2.1 Class Hierarchy

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

LibTopoART.CategoryInfo	10
LibTopoART.Episodic_TopoART	12
LibTopoART_samples.Episodic_TopoART_sample1	12
LibTopoART_samples.Episodic_TopoART_sample2	12
Exception	
LibTopoART.InvalidClassIDException	58
LibTopoART.InvalidFileException	58
LibTopoART.InvalidModuleIndexException	59
LibTopoART.InvalidNodeNumberException	60
LibTopoART.InvalidSizeException	61
LibTopoART.InvalidStateException	62
LibTopoART.F2_output	12
LibTopoART.IAdaptationStateCheck	51
LibTopoART.ITopoART	63
LibTopoART.Fast_TopoART_base	26
LibTopoART.Fast_Episodic_TopoART	14
LibTopoART.Fast_TopoART	18
LibTopoART.Fast_TopoART_AM	21
LibTopoART.Fast_TopoART_C	33
LibTopoART.Fast_TopoART_R	38
LibTopoART.ITopoART_AM	67
LibTopoART.Fast_TopoART_AM	21
LibTopoART.ITopoART_C	69
LibTopoART.Fast_TopoART_C	33
LibTopoART.Hypersphere_TopoART_C	46
LibTopoART.TopoART_C	83
LibTopoART.ITopoART_R	71
LibTopoART.Fast_TopoART_R	38

LibTopoART.TopoART_R	90
LibTopoART.TopoART	75
LibTopoART.Hypersphere_TopoART	43
LibTopoART.Hypersphere_TopoART_C	46
LibTopoART.TopoART_C	83
LibTopoART.TopoART_R	90
LibTopoART.IAssociative_recall	52
LibTopoART.ITopoART_AM	67
LibTopoART.ICategoryAccess	55
LibTopoART.Fast_TopoART_base	26
LibTopoART.TopoART	75
IDisposable	
LibTopoART.Fast_TopoART_base	26
LibTopoART.TopoART	75
LibTopoART.IEpisodic_recall	56
LibTopoART.Fast_Episodic_TopoART	14
LibTopoART.LibTopoART_info	74
LibTopoART.TopoART_AM	83
LibTopoART_samples.TopoART_AM_sample1	83
LibTopoART.TopoART_C_Prediction	88
LibTopoART_samples.TopoART_C_sample1	89
LibTopoART_samples.TopoART_C_sample2	89
LibTopoART.TopoART_R_Prediction	94
LibTopoART_samples.TopoART_R_sample1	96
LibTopoART_samples.TopoART_R_sample2	96
LibTopoART_samples.TopoART_R_sample3	96
LibTopoART_samples.TopoART_sample1	97
LibTopoART_samples.TopoART_sample2	97
LibTopoART_samples.TopoART_sample3	97

3 Class Index

3.1 Class List

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

LibTopoART.CategoryInfo	
Struct CategoryInfo summarises information about a node's category	10
LibTopoART.Episodic_TopoART	
Class Episodic_TopoART is obsolete. Please use class Fast_Episodic_TopoART instead!	12
LibTopoART_samples.Episodic_TopoART_sample1	
Episodic clustering sample using synthetic two-dimensional data. [C#]	12
LibTopoART_samples.Episodic_TopoART_sample2	
Episodic clustering sample using real-world video data. [C#]	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_Episodic_TopoART	
Class Fast_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."	14
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."	18
LibTopoART.Fast_TopoART_AM	
Class Fast_TopoART_AM provides an implementation of the TopoART-AM neural network as proposed in "Marko Tscherepanow, Marco Kortkamp and Marc Kammer (2011). A Hierarchical ART Network for the Stable Incremental Learning of Topological Structures and Associations from Noisy Data. Neural Networks 24(8): 906-916. Elsevier."	21
LibTopoART.Fast_TopoART_base	
Base class providing functionality common to several TopoART networks	26
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."	33
LibTopoART.Fast_TopoART_R	
Class Fast_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."	38

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."

43

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."

46

LibTopoART.IAdaptationStateCheck

Interface enabling checks whether certain adaptations of a network occurred

51

LibTopoART.IAssociative_recall

Interface summarising the associative recall functionality

52

LibTopoART.ICategoryAccess

Interface providing access to the learnt categories, e.g for drawing

55

LibTopoART.IEpisodic_recall

Interface summarising the episodic recall functionality

56

LibTopoART.InvalidClassIDException

Exception signalling an invalid class ID

58

LibTopoART.InvalidFileException

Exception signalling an invalid file

58

LibTopoART.InvalidModuleIndexException

Exception signalling an invalid module index

59

LibTopoART.InvalidNodeNumberException

Exception signalling an invalid node number

60

LibTopoART.InvalidSizeException

Exception signalling an invalid size

61

LibTopoART.InvalidStateException

Exception signalling an invalid state of the neural network

62

LibTopoART.ITopoART

Interface summarising the basic **TopoART** functionality

63

LibTopoART.ITopoART_AM

Interface summarising the **TopoART-AM** functionality

67

LibTopoART.ITopoART_C

Interface summarising the **TopoART-C** functionality

69

LibTopoART.ITopoART_R

Interface summarising the **TopoART-R** functionality

71

LibTopoART.LibTopoART_info	
Struct LibTopoART_info provides some metainformation regarding the respective implementation of LibTopoART	74
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."	75
LibTopoART.TopoART_AM	
Class TopoART_AM is obsolete. Please use class Fast_TopoART_AM instead!	83
LibTopoART_samples.TopoART_AM_sample1	
Sample using TopoART-AM with synthetic two-dimensional data. [C#]	83
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."	83
LibTopoART.TopoART_C_Prediction	
Struct TopoART_C_Prediction contains a prediction made by a TopoART-C network	88
LibTopoART_samples.TopoART_C_sample1	
Simple classification sample. [C#]	89
LibTopoART_samples.TopoART_C_sample2	
Classification sample using more complex synthetic two-dimensional data with associated class IDs. [C#]	89
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."	90
LibTopoART.TopoART_R_Prediction	
Struct TopoART_R_Prediction contains a prediction made by a TopoART-R network	94
LibTopoART_samples.TopoART_R_sample1	
Regression sample using TopoART-R. (simplified version) [C#]	96
LibTopoART_samples.TopoART_R_sample2	
Regression sample using TopoART-R. (advanced version) [C#]	96
LibTopoART_samples.TopoART_R_sample3	
Pixel-wise regression analysis of an image using TopoART-R. [F#]	96
LibTopoART_samples.TopoART_sample1	
Simple clustering sample. [C#]	97
LibTopoART_samples.TopoART_sample2	
Clustering sample using more complex synthetic two-dimensional data. [C#]	97
LibTopoART_samples.TopoART_sample3	
Clustering sample using very noisy synthetic two-dimensional data. [VB]	97

4 Namespace Documentation

4.1 LibTopoART Namespace Reference

Classes

- struct [CategoryInfo](#)
 Struct [CategoryInfo](#) summarises information about a node's category.
- class [Episodic_TopoART](#)
 Class [Episodic_TopoART](#) is obsolete. Please use class [Fast_Episodic_TopoART](#) instead!
- 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_Episodic_TopoART](#)
 Class [Fast_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 [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_AM](#)
 Class [Fast_TopoART_AM](#) provides an implementation of the TopoART-AM neural network as proposed in "Marko Tscherepanow, Marco Kortkamp and Marc Kammer (2011). A Hierarchical ART Network for the Stable Incremental Learning of Topological Structures and Associations from Noisy Data. *Neural Networks* 24(8): 906-916. Elsevier."
- 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 [Fast_TopoART_R](#)
 Class [Fast_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 [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 [IAssociative_recall](#)

- Interface summarising the associative recall functionality.*

 - interface [ICategoryAccess](#)

Interface providing access to the learnt categories, e.g for drawing.
 - 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 [InvalidModuleIndexException](#)

Exception signalling an invalid module index.
 - class [InvalidNodeNumberException](#)

Exception signalling an invalid node number.
 - class [InvalidSizeException](#)

Exception signalling an invalid size.
 - class [InvalidStateException](#)

Exception signalling an invalid state of the neural network.
 - interface [ITopoART](#)

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

Interface summarising the [TopoART-AM](#) 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 metainformation 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_AM](#)

Class [TopoART_AM](#) is obsolete. Please use class [Fast_TopoART_AM](#) instead!
 - 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_](#)
= 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. [C#]
- class [Episodic_TopoART_sample2](#)
Episodic clustering sample using real-world video data. [C#]
- class [TopoART_AM_sample1](#)
Sample using TopoART-AM with synthetic two-dimensional data. [C#]
- class [TopoART_C_sample1](#)

- Simple classification sample. [C#]*

 - class [TopoART_C_sample2](#)

Classification sample using more complex synthetic two-dimensional data with associated class IDs. [C#]

 - class [TopoART_R_sample1](#)

Regression sample using TopoART-R. (simplified version) [C#]

 - class [TopoART_R_sample2](#)

Regression sample using TopoART-R. (advanced version) [C#]

 - class [TopoART_R_sample3](#)

Pixel-wise regression analysis of an image using TopoART-R. [F#]

 - class [TopoART_sample1](#)

Simple clustering sample. [C#]

 - class [TopoART_sample2](#)

Clustering sample using more complex synthetic two-dimensional data. [C#]

 - class [TopoART_sample3](#)

Clustering sample using very noisy synthetic two-dimensional data. [VB]

5 Class Documentation

5.1 LibTopoART.CategoryInfo Struct Reference

Struct [CategoryInfo](#) summarises information about a node's category.

Public Member Functions

- [CategoryInfo](#) (decimal[] [spatial_weights](#), decimal[] [temporal_weights](#), long [cluster_ID](#), long [class_ID](#))
This constructor sets the instance variables [spatial_weights](#), [temporal_weights](#), [cluster_ID](#), and [class_ID](#) of struct [CategoryInfo](#).

Public Attributes

- readonly decimal[] [spatial_weights](#)
Instance variable [spatial_weights](#) represents the spatial weights of the considered node.
- readonly decimal[] [temporal_weights](#)
Instance variable [temporal_weights](#) represents the temporal weights of the considered node if it support temporal learning.
- readonly long [cluster_ID](#)
Instance variable [cluster_ID](#) represents the cluster ID of the considered node.
- readonly long [class_ID](#)
Instance variable [class_ID](#) represents the class ID of the considered node. If class IDs are not supported by the respective node, this value is set to [LibTopoART_info.UNDEFINED](#).

5.1.1 Detailed Description

Struct [CategoryInfo](#) summarises information about a node's category.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 CategoryInfo() `LibTopoART.CategoryInfo.CategoryInfo (decimal[] spatial_weights, decimal[] temporal_weights, long cluster_ID, long class_ID)`

This constructor sets the instance variables `spatial_weights`, `temporal_weights`, `cluster_ID`, and `class_ID` of struct `CategoryInfo`.

Parameters

<i>spatial_weights</i>	The spatial weights to be set.
<i>temporal_weights</i>	The temporal weights to be set.
<i>cluster_ID</i>	The cluster ID to be set.
<i>class_ID</i>	The class ID to be set.

5.1.3 Member Data Documentation

5.1.3.1 class_ID `readonly long LibTopoART.CategoryInfo.class_ID`

Instance variable `class_ID` represents the class ID of the considered node. If class IDs are not supported by the respective node, this value is set to `LibTopoART_info.UNDEFINED`.

5.1.3.2 cluster_ID `readonly long LibTopoART.CategoryInfo.cluster_ID`

Instance variable `cluster_ID` represents the cluster ID of the considered node.

5.1.3.3 spatial_weights `readonly decimal [] LibTopoART.CategoryInfo.spatial_weights`

Instance variable `spatial_weights` represents the spatial weights of the considered node.

5.1.3.4 temporal_weights `readonly decimal [] LibTopoART.CategoryInfo.temporal_weights`

Instance variable `temporal_weights` represents the temporal weights of the considered node if it support temporal learning.

5.2 LibTopoART.Episodic_TopoART Class Reference

Class [Episodic_TopoART](#) is obsolete. Please use class [Fast_Episodic_TopoART](#) instead!

5.2.1 Detailed Description

Class [Episodic_TopoART](#) is obsolete. Please use class [Fast_Episodic_TopoART](#) instead!

5.3 LibTopoART_samples.Episodic_TopoART_sample1 Class Reference

Episodic clustering sample using synthetic two-dimensional data. [C#]

5.3.1 Detailed Description

Episodic clustering sample using synthetic two-dimensional data. [C#]

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.

5.4 LibTopoART_samples.Episodic_TopoART_sample2 Class Reference

Episodic clustering sample using real-world video data. [C#]

5.4.1 Detailed Description

Episodic clustering sample using real-world video data. [C#]

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 length 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.

5.5 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.

5.5.1 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`.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 [F2_output](#)() `LibTopoART.F2_output.F2_output ()`

This constructor sets all instance variables of class [F2_output](#) to [LibTopoART_info.UNDEFINED](#).

5.5.3 Member Data Documentation

5.5.3.1 [bm_cluster_ID](#) `long LibTopoART.F2_output.bm_cluster_ID`

Instance variable [bm_cluster_ID](#) represents the cluster ID of the best-matching node.

5.5.3.2 **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).

5.5.3.3 **bm_node_ID** long LibTopoART.F2_output.bm_node_ID

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

5.5.3.4 **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.

5.5.3.5 **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).

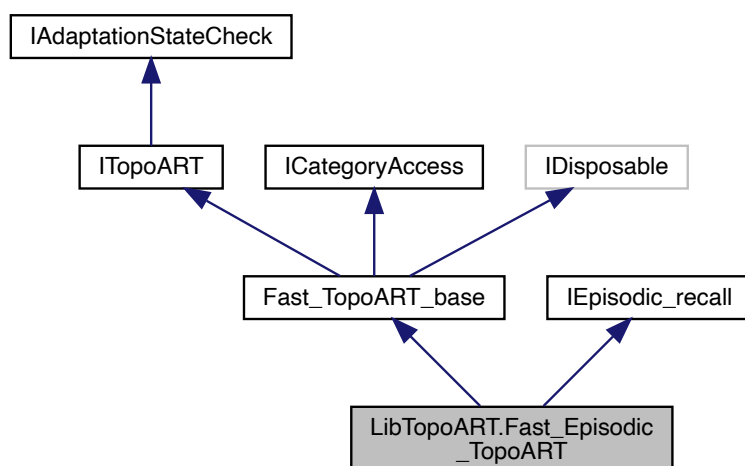
5.5.3.6 **bm_permanent_node_ID** long LibTopoART.F2_output.bm_permanent_node_ID

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

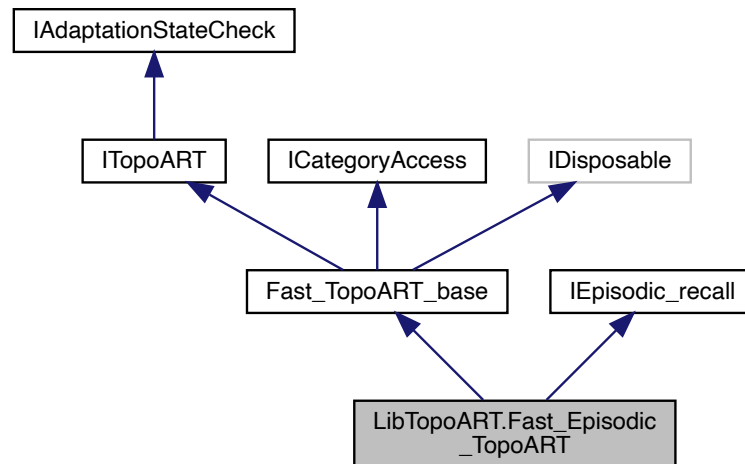
5.6 LibTopoART.Fast_Episodic_TopoART Class Reference

Class `Fast_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."

Inheritance diagram for `LibTopoART.Fast_Episodic_TopoART`:



Collaboration diagram for LibTopoART.Fast_Episodic_TopoART:



Public Member Functions

- [Fast_Episodic_TopoART](#) (long input_length, long module_number, decimal rho_a, long t_max)
This constructor initialises an Episodic [TopoART](#) network.
- [Fast_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

5.6.1 Detailed Description

Class `Fast_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."

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Fast_Episodic_TopoART() [1/2] `LibTopoART.Fast_Episodic_TopoART.Fast_Episodic_TopoART (long input_length, long module_number, decimal rho_a, long t_max)`

This constructor initialises an Episodic `TopoART` network.

Parameters

<i>input_length</i>	The length of input vectors to be learnt.
<i>module_number</i>	The number of Episodic <code>TopoART</code> modules.
<i>rho_a</i>	The vigilance parameter of the first Episodic <code>TopoART</code> module (ETA a).
<i>t_max</i>	The parameter limiting the considered time frame.

5.6.2.2 Fast_Episodic_TopoART() [2/2] `LibTopoART.Fast_Episodic_TopoART.Fast_Episodic_TopoART (string path)`

This constructor loads a saved Episodic `TopoART` network.

Parameters

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

Exceptions

<code>InvalidFileException</code>	Throws when the given file cannot be loaded.
-----------------------------------	--

5.6.3 Member Function Documentation

5.6.3.1 BeginRecall() `long LibTopoART.Fast_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](#).

5.6.3.2 EndRecall() `void LibTopoART.Fast_Episodic_TopoART.EndRecall ()`

This method stops the recall process and frees temporary resources.

Implements [LibTopoART.IEpisodic_recall](#).

5.6.3.3 InterEpisodeRecallStep() `bool LibTopoART.Fast_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](#).

5.6.3.4 IntraEpisodeRecallStep() `bool LibTopoART.Fast_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](#).

5.6.3.5 Learn() `override void LibTopoART.Fast_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](#).

5.6.4 Property Documentation

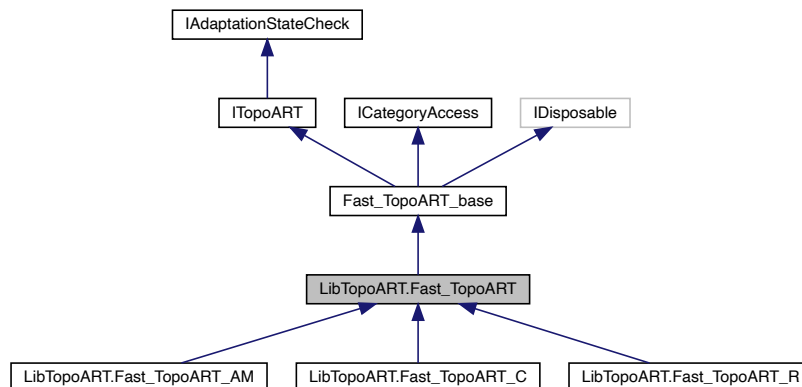
5.6.4.1 FileFormatVersion `new decimal LibTopoART.Fast_Episodic_TopoART.FileFormatVersion [get]`

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

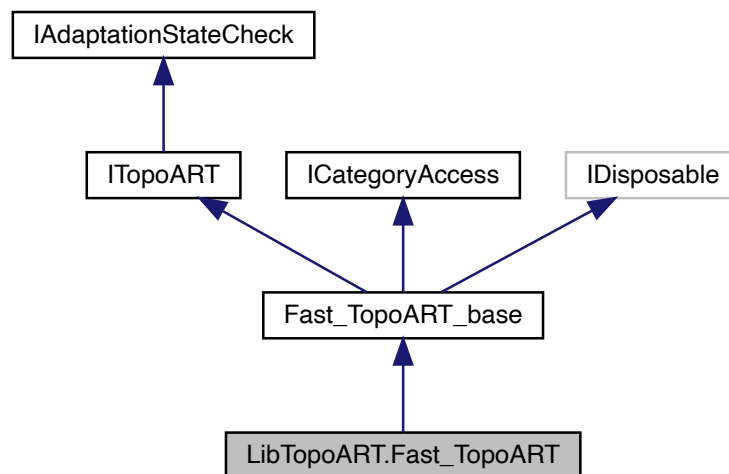
5.7 LibTopoART.Fast_TopoART Class Reference

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."

Inheritance diagram for LibTopoART.Fast_TopoART:



Collaboration diagram for LibTopoART.Fast_TopoART:



Public Member Functions

- **Fast_TopoART** (long input_length, 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

5.7.1 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 $[0,1]$.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 `Fast_TopoART()` [1/2] `LibTopoART.Fast_TopoART.Fast_TopoART (`
`long input_length,`
`long module_number,`
`decimal rho_a)`

This constructor initialises a `TopoART` network.

Parameters

<i>input_length</i>	The length 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).

5.7.2.2 `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.
-------------	---

Exceptions

<code>InvalidFileException</code>	Throws when the given file cannot be loaded.
-----------------------------------	--

5.7.3 Member Function Documentation

5.7.3.1 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.
--------------	--------------------------------

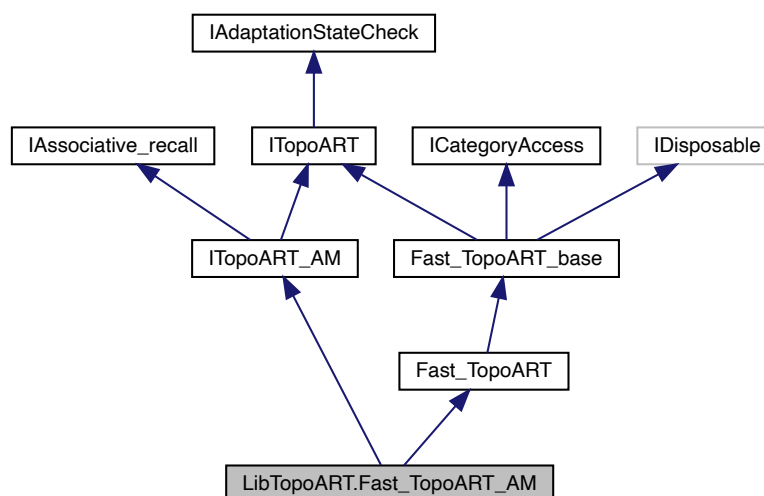
Implements [LibTopoART.Fast_TopoART_base](#).

Reimplemented in [LibTopoART.Fast_TopoART_R](#), and [LibTopoART.Fast_TopoART_C](#).

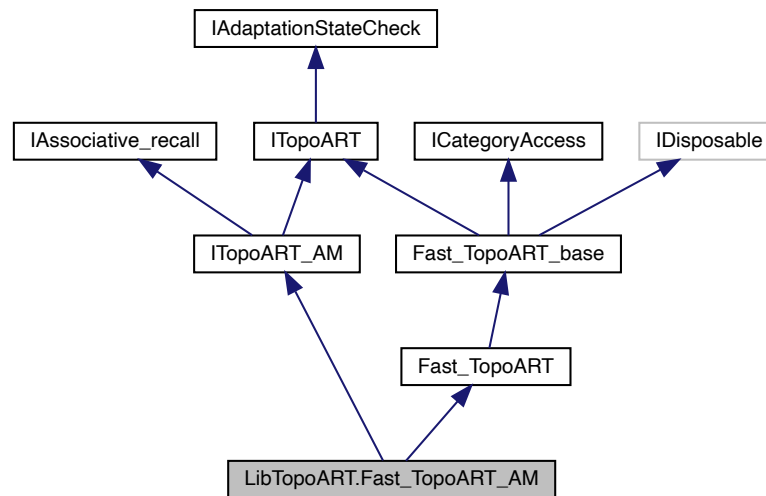
5.8 LibTopoART.Fast_TopoART_AM Class Reference

Class [Fast_TopoART_AM](#) provides an implementation of the TopoART-AM neural network as proposed in "Marko Tscherepanow, Marco Kortkamp and Marc Kammer (2011). A Hierarchical ART Network for the Stable Incremental Learning of Topological Structures and Associations from Noisy Data. Neural Networks 24(8): 906-916. Elsevier."

Inheritance diagram for LibTopoART.Fast_TopoART_AM:



Collaboration diagram for LibTopoART.Fast_TopoART_AM:



Public Member Functions

- [Fast_TopoART_AM](#) (long key_1_length, long key_2_length, long module_number, decimal rho_a)
This constructor initialises a TopoART-AM network.
- [Fast_TopoART_AM](#) (string path)
This constructor loads a saved TopoART-AM network.
- [F2_output\[\] GetBMOOutput](#) (decimal[] key_1_vec, decimal[] key_2_vec)
This method finds the closest category for a given pair of keys.
- void [Learn](#) (decimal[] key_1_vec, decimal[] key_2_vec)
This method performs a single training step.
- long [BeginRecallKey1](#) (decimal[] key_2_vec, long module_index=FINAL_MODULE)
This method starts the recall process for the first key vector.
- long [BeginRecallKey2](#) (decimal[] key_1_vec, long module_index=FINAL_MODULE)
This method starts the recall process for the second key vector.
- bool [RecallStep](#) (out decimal[] recall_result, out decimal F3_activation)
This method performs a single associative 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 TopoART_AM.
- long [Key_1_len](#) [get]
Property Key_1_len returns the length of the first key vector.
- long [Key_2_len](#) [get]
Property Key_2_len returns the length of the second key vector.

Additional Inherited Members

5.8.1 Detailed Description

Class [Fast_TopoART_AM](#) provides an implementation of the TopoART-AM neural network as proposed in "Marko Tscherepanow, Marco Kortkamp and Marc Kammer (2011). A Hierarchical ART Network for the Stable Incremental Learning of Topological Structures and Associations from Noisy Data. Neural Networks 24(8): 906-916. Elsevier."

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

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Fast_TopoART_AM() [1/2] `LibTopoART.Fast_TopoART_AM.Fast_TopoART_AM (long key_1_length, long key_2_length, long module_number, decimal rho_a)`

This constructor initialises a TopoART-AM network.

Parameters

<i>key_1_length</i>	The length of the first key vector to be learnt.
<i>key_2_length</i>	The length of the second key vector to be learnt.
<i>module_number</i>	The number of TopoART-AM modules.
<i>rho_a</i>	The vigilance parameter of the first TopoART-AM module (TopoART-AM a).

5.8.2.2 Fast_TopoART_AM() [2/2] `LibTopoART.Fast_TopoART_AM.Fast_TopoART_AM (string path)`

This constructor loads a saved TopoART-AM network.

Parameters

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

Exceptions

InvalidFileException	Throws when the given file cannot be loaded.
--------------------------------------	--

5.8.3 Member Function Documentation

5.8.3.1 BeginRecallKey1() `long LibTopoART.Fast_TopoART_AM.BeginRecallKey1 (`
`decimal[] key_2_vec,`
`long module_index = FINAL_MODULE)`

This method starts the recall process for the first key vector.

Parameters

<i>key_2_vec</i>	The stimulus (second key vector) which is used to trigger recall.
<i>module_index</i>	Index of the TopoART-AM module to be used for recall. (FINAL_MODULE denotes the module with the highest index.)

Returns

The number of F3 nodes created.

Exceptions

InvalidModuleIndexException	Throws when <i>module_index</i> is invalid.
---	---

Implements [LibTopoART.IAssociative_recall](#).

5.8.3.2 BeginRecallKey2() `long LibTopoART.Fast_TopoART_AM.BeginRecallKey2 (`
`decimal[] key_1_vec,`
`long module_index = FINAL_MODULE)`

This method starts the recall process for the second key vector.

Parameters

<i>key_1_vec</i>	The stimulus (first key vector) which is used to trigger recall.
<i>module_index</i>	Index of the TopoART-AM module to be used for recall. (FINAL_MODULE denotes the module with the highest index.)

Returns

The number of F3 nodes created.

Exceptions

InvalidModuleIndexException	Throws when <i>module_index</i> is invalid.
---	---

Implements [LibTopoART.IAssociative_recall](#).

5.8.3.3 EndRecall() `void LibTopoART.Fast_TopoART_AM.EndRecall ()`

This method stops the recall process and frees temporary resources.

Implements [LibTopoART.IAssociative_recall](#).

5.8.3.4 GetBMOutput() `F2_output [] LibTopoART.Fast_TopoART_AM.GetBMOutput (decimal[] key_1_vec, decimal[] key_2_vec)`

This method finds the closest category for a given pair of keys.

Parameters

<i>key_1_vec</i>	The first key vector.
<i>key_2_vec</i>	The second key vector corresponding to <i>key_1_vec</i> .

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-AM module.

Implements [LibTopoART.ITopoART_AM](#).

5.8.3.5 Learn() `void LibTopoART.Fast_TopoART_AM.Learn (decimal[] key_1_vec, decimal[] key_2_vec)`

This method performs a single training step.

Parameters

<i>key_1_vec</i>	The first key vector to be learnt.
<i>key_2_vec</i>	The second key vector corresponding to <i>key_1_vec</i> .

Implements [LibTopoART.ITopoART_AM](#).

5.8.3.6 RecallStep() `bool LibTopoART.Fast_TopoART_AM.RecallStep (out decimal[] recall_result, out decimal F3_activation)`

This method performs a single associative recall step.

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.IAssociative_recall](#).

5.8.4 Property Documentation

5.8.4.1 FileFormatVersion `new decimal LibTopoART.Fast_TopoART_AM.FileFormatVersion [get]`

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

5.8.4.2 Key_1_len `long LibTopoART.Fast_TopoART_AM.Key_1_len [get]`

Property `Key_1_len` returns the length of the first key vector.

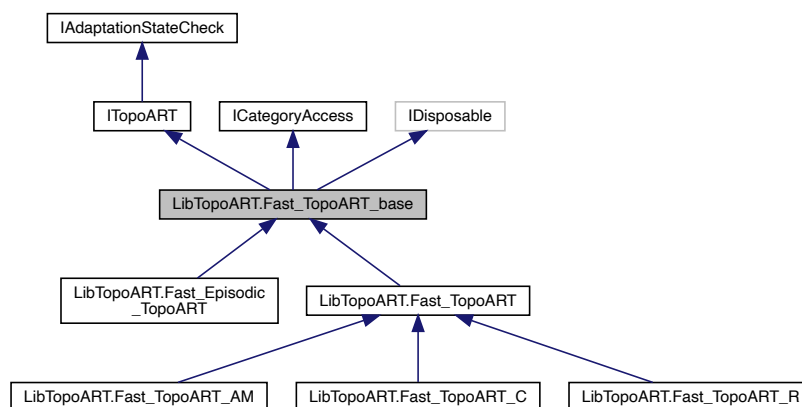
5.8.4.3 Key_2_len `long LibTopoART.Fast_TopoART_AM.Key_2_len [get]`

Property `Key_2_len` returns the length of the second key vector.

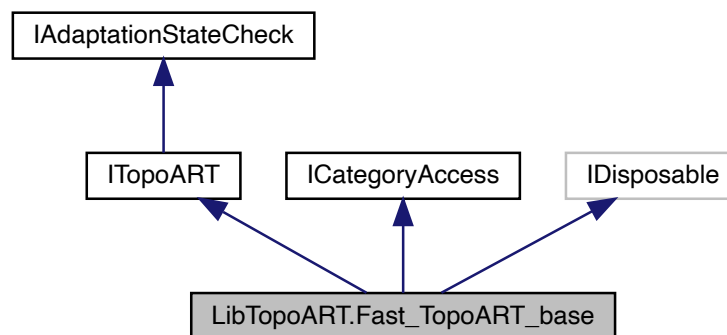
5.9 LibTopoART.Fast_TopoART_base Class Reference

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

Inheritance diagram for `LibTopoART.Fast_TopoART_base`:



Collaboration diagram for LibTopoART.Fast_TopoART_base:



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.
- List< [CategoryInfo](#) > [GetCategories](#) (long module_index=[FINAL_MODULE](#))
This method collects information on the categories of a specified module.

Static Public Attributes

- const long [FINAL_MODULE](#) = LibTopoART_info.FINAL_MODULE
Instance variable [FINAL_MODULE](#) gives the value used for indicating that the [TopoART](#) module with the highest index is to be used.

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 [InputLen](#) [get]
Property InputLen returns the length of the input vector.
- 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]
- long[] [Phis](#) [get, set]
- 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](#).

5.9.1 Detailed Description

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

5.9.2 Member Function Documentation

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

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

5.9.2.2 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.

5.9.2.3 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.
------------------	---

5.9.2.4 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.

Exceptions

InvalidStateException	Throws when the network is in an invalid state.
---------------------------------------	---

Implements [LibTopoART.IAdaptationStateCheck](#).

5.9.2.5 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`.

5.9.2.6 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`.

5.9.2.7 GetCategories() `List<CategoryInfo> LibTopoART.Fast_TopoART_base.GetCategories (long module_index = FINAL_MODULE)`

This method collects information on the categories of a specified module.

Parameters

<i>module_index</i>	The index of the module the categories of which are to be analysed.
---------------------	---

Returns

A list containing information about the respective categories.

Exceptions

<i>InvalidModuleIndexException</i>	Throws when <i>module_index</i> is invalid.
<i>InvalidNodeNumberException</i>	Throws when the number of nodes of a module is greater than <code>int.MaxValue</code> .

Implements [LibTopoART.ICategoryAccess](#).

5.9.2.8 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_R](#), [LibTopoART.Fast_TopoART_C](#), [LibTopoART.Fast_Episodic_TopoART](#), and [LibTopoART.Fast_TopoART](#).

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

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

Implements [LibTopoART.IAdaptationStateCheck](#).

5.9.2.10 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](#).

5.9.2.11 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.

5.9.2.12 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](#).

5.9.3 Member Data Documentation

5.9.3.1 FINAL_MODULE `const long LibTopoART.Fast_TopoART_base.FINAL_MODULE = LibTopoART_↵`
`info.FINAL_MODULE [static]`

Instance variable `FINAL_MODULE` gives the value used for indicating that the [TopoART](#) module with the highest index is to be used.

5.9.4 Property Documentation

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

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

5.9.4.2 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.

5.9.4.3 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.

5.9.4.4 Phi `long LibTopoART.Fast_TopoART_base.Phi [get], [set]`

Property `Phi` represents the parameter phi required for the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules.

5.9.4.5 Phis `long [] LibTopoART.Fast_TopoART_base.Phis [get], [set]`

Property `Phis` constitutes an extension of property `Phi` that enables individual values of phi for each module. By this, the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules can be controlled in a task dependent manner.

Exceptions

InvalidSizeException	Throws when the array length does not fit the module number.
--------------------------------------	--

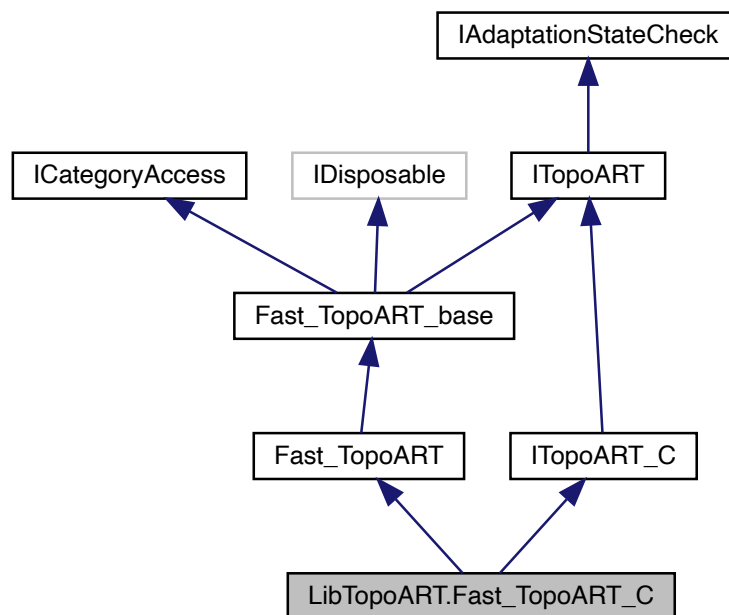
5.9.4.6 TopoARTFileFormatVersion `decimal LibTopoART.Fast_TopoART_base.TopoARTFileFormatVersion [get]`

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

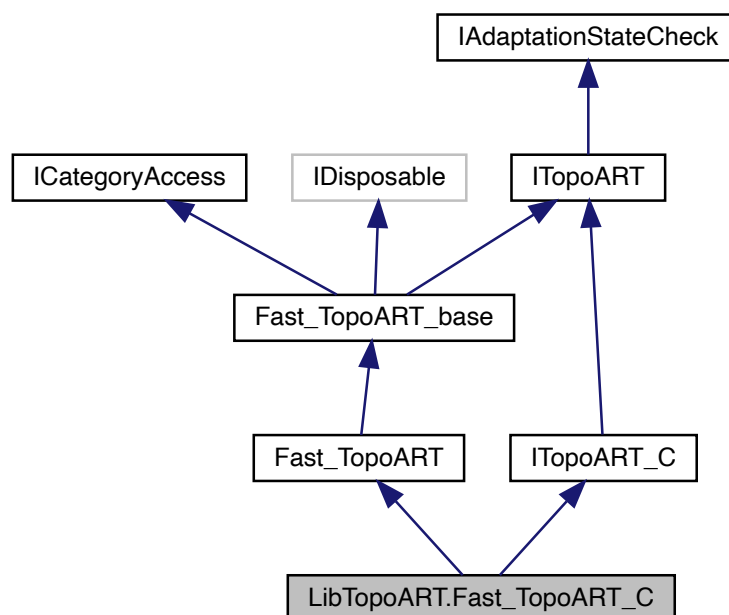
5.10 LibTopoART.Fast_TopoART_C Class Reference

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."

Inheritance diagram for LibTopoART.Fast_TopoART_C:



Collaboration diagram for LibTopoART.Fast_TopoART_C:



Public Member Functions

- [Fast_TopoART_C](#) (long input_length, 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.

Static 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

5.10.1 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.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 Fast_TopoART_C() [1/2] LibTopoART.Fast_TopoART_C.Fast_TopoART_C (
long input_length,
long module_number,
decimal rho_a)

This constructor initialises a TopoART-C network.

Parameters

<i>input_length</i>	The length 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).

5.10.2.2 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.
-------------	--------------------------------------

Exceptions

<i>InvalidFileException</i>	Throws when the given file cannot be loaded.
---	--

5.10.3 Member Function Documentation

5.10.3.1 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](#).

5.10.3.2 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)

Exceptions

<i>InvalidClassIDException</i>	Throws when <i>class_ID</i> is less than 0.
--	---

Implements [LibTopoART.ITopoART_C](#).

5.10.3.3 Predict() [1/2] [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](#).

5.10.3.4 Predict() [2/2] 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](#).

5.10.4 Member Data Documentation

5.10.4.1 UNDEFINED_CLASS_ID `const long LibTopoART.Fast_TopoART_C.UNDEFINED_CLASS_ID = -2`
[static]

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.

5.10.5 Property Documentation

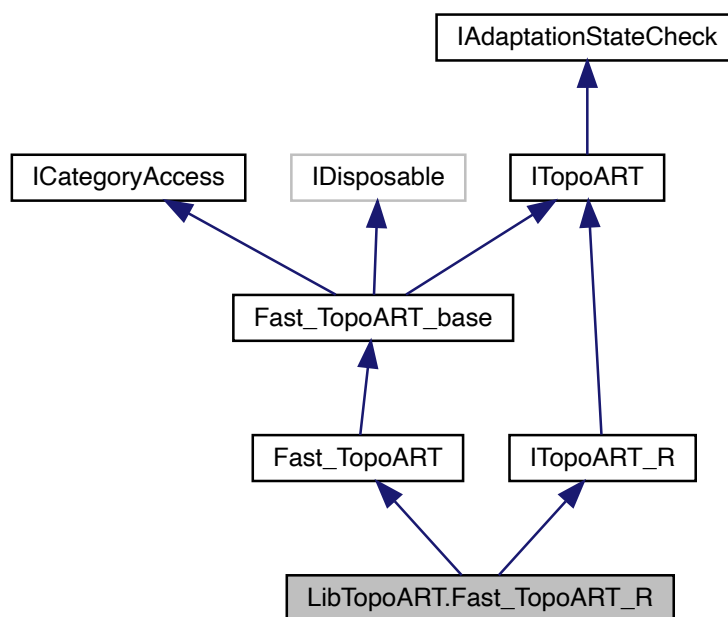
5.10.5.1 FileFormatVersion `new decimal LibTopoART.Fast_TopoART_C.FileFormatVersion` [get]

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

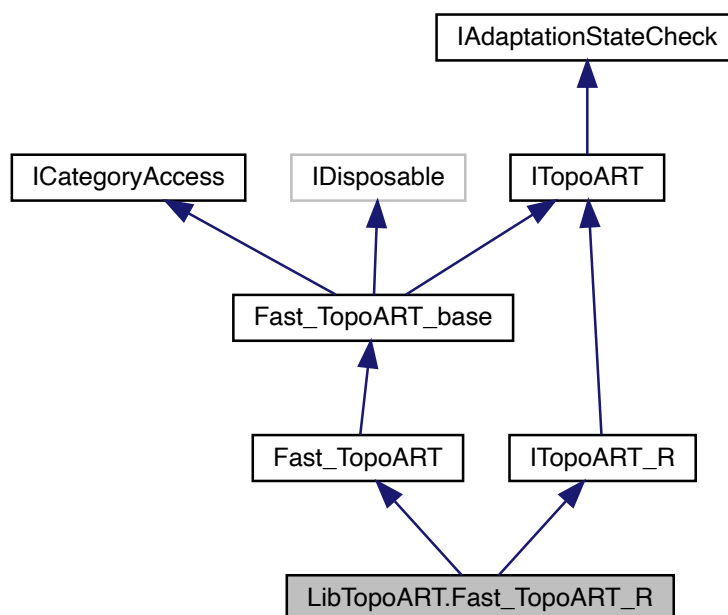
5.11 LibTopoART.Fast_TopoART_R Class Reference

Class [Fast_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."

Inheritance diagram for LibTopoART.Fast_TopoART_R:



Collaboration diagram for LibTopoART.Fast_TopoART_R:



Public Member Functions

- [Fast_TopoART_R](#) (long i_length, long d_length, long module_number, decimal rho_a)
This constructor initialises a TopoART-R network.
- [Fast_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

- long [D_len](#) [get]
Property D_len returns the length of the output vector (dependent variables).
- new decimal [FileFormatVersion](#) [get]
Property FileFormatVersion returns the version of the file format used by class [Fast_TopoART_R](#).
- long [I_len](#) [get]
Property I_len returns the length of the input vector (independent variables).

Additional Inherited Members

5.11.1 Detailed Description

Class [Fast_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."

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_R](#).

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

5.11.2 Constructor & Destructor Documentation

5.11.2.1 [Fast_TopoART_R\(\)](#) [1/2] `LibTopoART.Fast_TopoART_R.Fast_TopoART_R (`
`long i_length,`
`long d_length,`
`long module_number,`
`decimal rho_a)`

This constructor initialises a TopoART-R network.

Parameters

<i>i_length</i>	The length of the input vector (independent variables) to be learnt.
<i>d_length</i>	The length 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).

5.11.2.2 Fast_TopoART_R() [2/2] LibTopoART.Fast_TopoART_R.Fast_TopoART_R (string path)

This constructor loads a saved TopoART-R network.

Parameters

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

Exceptions

InvalidFileException	Throws when the given file cannot be loaded.
--------------------------------------	--

5.11.3 Member Function Documentation

5.11.3.1 Learn() [1/2] void LibTopoART.Fast_TopoART_R.Learn (decimal[] i_vec, decimal[] d_vec)

This method performs a single training step.

Parameters

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

Implements [LibTopoART.ITopoART_R](#).

5.11.3.2 Learn() [2/2] override void LibTopoART.Fast_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.Fast_TopoART](#).

5.11.3.3 Predict() [1/2] [TopoART_R_Prediction](#) LibTopoART.Fast_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](#).

5.11.3.4 Predict() [2/2] decimal[] LibTopoART.Fast_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](#).

5.11.4 Property Documentation

5.11.4.1 D_len long LibTopoART.Fast_TopoART_R.D_len [get]

Property `D_len` returns the length of the output vector (dependent variables).

5.11.4.2 FileFormatVersion new decimal LibTopoART.Fast_TopoART_R.FileFormatVersion [get]

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

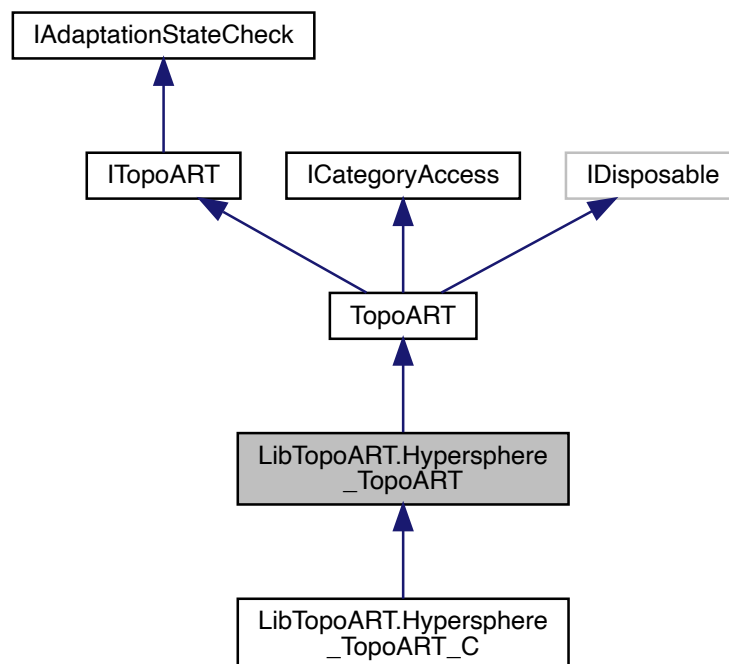
5.11.4.3 I_len long LibTopoART.Fast_TopoART_R.I_len [get]

Property `I_len` returns the length of the input vector (independent variables).

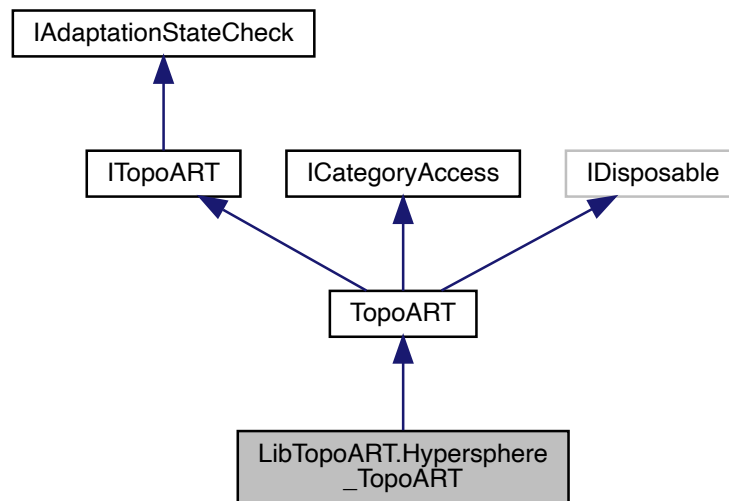
5.12 LibTopoART.Hypersphere_TopoART Class Reference

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."

Inheritance diagram for LibTopoART.Hypersphere_TopoART:



Collaboration diagram for LibTopoART.Hypersphere_TopoART:



Public Member Functions

- [Hypersphere_TopoART](#) (long input_length, long module_number, decimal rho_a)
This constructor initialises a Hypersphere [TopoART](#) network and sets the radial extend parameter to $\text{Math}.\sqrt{\text{input_length}}/2$.
- [Hypersphere_TopoART](#) (long input_length, 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

5.12.1 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 .

5.12.2 Constructor & Destructor Documentation

5.12.2.1 Hypersphere_TopoART() [1/3] `LibTopoART.Hypersphere_TopoART.Hypersphere_TopoART (long input_length, long module_number, decimal rho_a)`

This constructor initialises a Hypersphere [TopoART](#) network and sets the radial extend parameter to `Math.Sqrt(input_length)/2`.

Parameters

<i>input_length</i>	The length 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).

5.12.2.2 Hypersphere_TopoART() [2/3] `LibTopoART.Hypersphere_TopoART.Hypersphere_TopoART (long input_length, long module_number, decimal rho_a, decimal R)`

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

Parameters

<i>input_length</i>	The length 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.

5.12.2.3 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.
-------------	--

Exceptions

InvalidFileException	Throws when the given file cannot be loaded.
--------------------------------------	--

5.12.3 Property Documentation

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

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

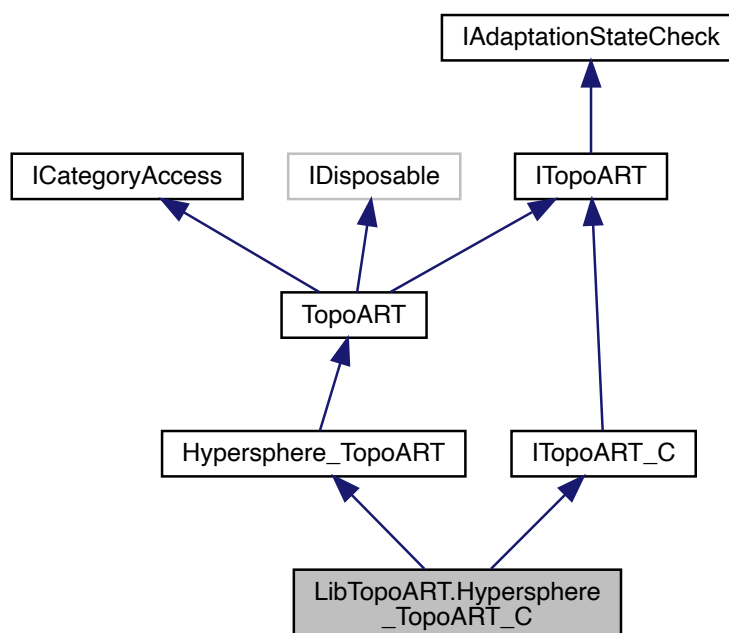
5.12.3.2 HypersphereTopoARTFileFormatVersion `decimal LibTopoART.Hypersphere_TopoART.Hypersphere↔
TopoARTFileFormatVersion [get]`

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

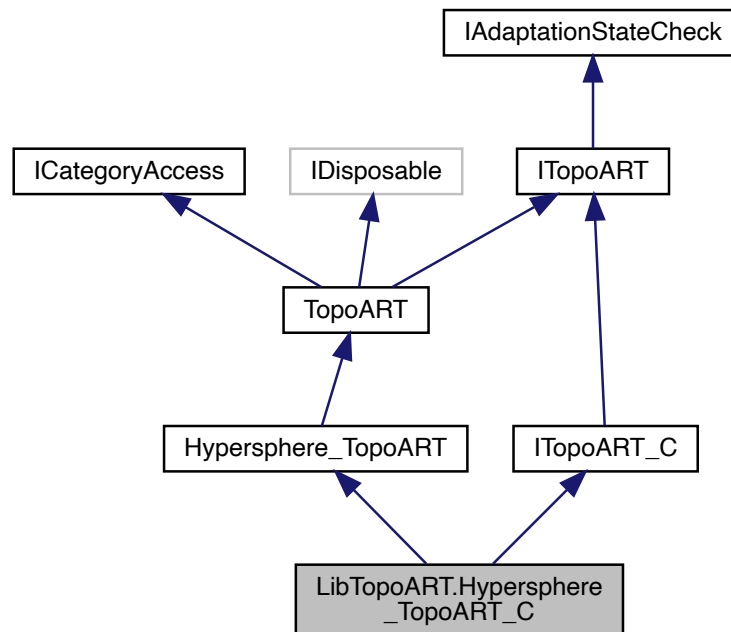
5.13 LibTopoART.Hypersphere_TopoART_C Class Reference

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."

Inheritance diagram for `LibTopoART.Hypersphere_TopoART_C`:



Collaboration diagram for LibTopoART.Hypersphere_TopoART_C:



Public Member Functions

- [Hypersphere_TopoART_C](#) (long input_length, long module_number, decimal rho_a)
This constructor initialises a Hypersphere TopoART-C network and sets the radial extend parameter to $\text{Math}.\sqrt{\text{input_length}}/2$.
- [Hypersphere_TopoART_C](#) (long input_length, 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.

Static 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

5.13.1 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.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 [Hypersphere_TopoART_C\(\)](#) [1/3] `LibTopoART.Hypersphere_TopoART_C.Hypersphere_TopoART_C (`
`long input_length,`
`long module_number,`
`decimal rho_a)`

This constructor initialises a Hypersphere TopoART-C network and sets the radial extend parameter to `Math.Sqrt(input_length)/2`.

Parameters

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

5.13.2.2 [Hypersphere_TopoART_C\(\)](#) [2/3] `LibTopoART.Hypersphere_TopoART_C.Hypersphere_TopoART_C (`
`long input_length,`
`long module_number,`
`decimal rho_a,`
`decimal R)`

This constructor initialises a Hypersphere TopoART-C network.

Parameters

<i>input_length</i>	The length 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.

5.13.2.3 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.
-------------	--

Exceptions

<i>InvalidFileException</i>	Throws when the given file cannot be loaded.
---	--

5.13.3 Member Function Documentation

5.13.3.1 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](#).

5.13.3.2 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)

Exceptions

<i>InvalidClassIDException</i>	Throws when <i>class_ID</i> is less than 0.
--	---

Implements [LibTopoART.ITopoART_C](#).

5.13.3.3 Predict() [1/2] [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](#).

5.13.3.4 Predict() [2/2] 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](#).

5.13.4 Member Data Documentation

```
5.13.4.1 UNDEFINED_CLASS_ID const long LibTopoART.Hypersphere_TopoART_C.UNDEFINED_CLASS_ID
= -2 [static]
```

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.

5.13.5 Property Documentation

5.13.5.1 FileFormatVersion

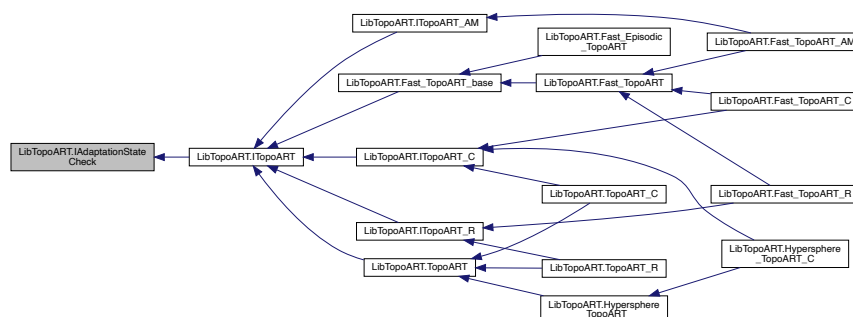
new decimal LibTopoART.Hypersphere_TopoART_C.FileFormatVersion [get]

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

5.14 LibTopoART.IAdaptationStateCheck Interface Reference

Interface enabling checks whether certain adaptations of a network occurred.

Inheritance diagram for LibTopoART.IAdaptationStateCheck:



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.

5.14.1 Detailed Description

Interface enabling checks whether certain adaptations of a network occurred.

5.14.2 Member Function Documentation

5.14.2.1 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](#).

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

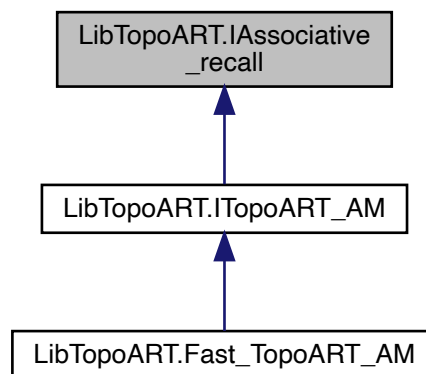
This method resets the adaptation state to [AdaptationState.NO_ADAPTATION](#).

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

5.15 LibTopoART.IAssociative_recall Interface Reference

Interface summarising the associative recall functionality.

Inheritance diagram for LibTopoART.IAssociative_recall:



Public Member Functions

- long [BeginRecallKey1](#) (decimal[] key_2_vec, long module_index=[Fast_TopoART_AM.FINAL_MODULE](#))
This method starts the recall process for the first key vector.
- long [BeginRecallKey2](#) (decimal[] key_1_vec, long module_index=[Fast_TopoART_AM.FINAL_MODULE](#))
This method starts the recall process for the second key vector.
- bool [RecallStep](#) (out decimal[] recall_result, out decimal F3_activation)
This method performs a single associative recall step.
- void [EndRecall](#) ()
This method stops the recall process and frees temporary resources.

5.15.1 Detailed Description

Interface summarising the associative recall functionality.

5.15.2 Member Function Documentation

5.15.2.1 BeginRecallKey1() long LibTopoART.IAssociative_recall.BeginRecallKey1 (decimal[] key_2_vec, long module_index = [Fast_TopoART_AM.FINAL_MODULE](#))

This method starts the recall process for the first key vector.

Parameters

<i>key_2_vec</i>	The stimulus (second key vector) which is used to trigger recall.
<i>module_index</i>	Index of the TopoART-AM module to be used for recall. (Fast_TopoART_AM.FINAL_MODULE denotes the module with the highest index.)

Returns

The number of F3 nodes created.

Implemented in [LibTopoART.Fast_TopoART_AM](#).

```
5.15.2.2 BeginRecallKey2() long LibTopoART.IAssociative_recall.BeginRecallKey2 (
    decimal[] key_1_vec,
    long module_index = Fast_TopoART_AM.FINAL_MODULE )
```

This method starts the recall process for the second key vector.

Parameters

<i>key_1_vec</i>	The stimulus (first key vector) which is used to trigger recall.
<i>module_index</i>	Index of the TopoART-AM module to be used for recall. (<i>Fast_TopoART_AM.FINAL_MODULE</i> denotes the module with the highest index.)

Returns

The number of F3 nodes created.

Implemented in [LibTopoART.Fast_TopoART_AM](#).

```
5.15.2.3 EndRecall() void LibTopoART.IAssociative_recall.EndRecall ( )
```

This method stops the recall process and frees temporary resources.

Implemented in [LibTopoART.Fast_TopoART_AM](#).

```
5.15.2.4 RecallStep() bool LibTopoART.IAssociative_recall.RecallStep (
    out decimal[] recall_result,
    out decimal F3_activation )
```

This method performs a single associative recall step.

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

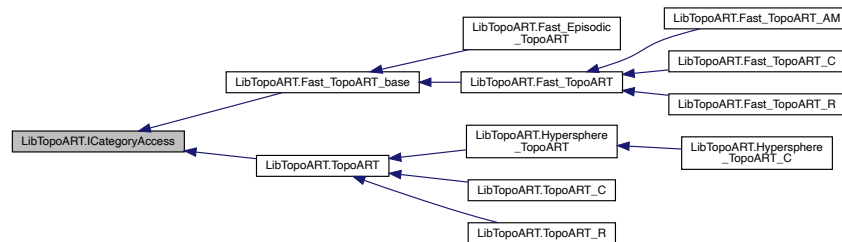
A boolean result indicating whether the recall step was successfully completed, or not.

Implemented in [LibTopoART.Fast_TopoART_AM](#).

5.16 LibTopoART.ICategoryAccess Interface Reference

Interface providing access to the learnt categories, e.g for drawing.

Inheritance diagram for LibTopoART.ICategoryAccess:



Public Member Functions

- List< [CategoryInfo](#) > [GetCategories](#) (long module_index=[LibTopoART_info.FINAL_MODULE](#))
This method collects information on the categories of a specified module.

5.16.1 Detailed Description

Interface providing access to the learnt categories, e.g for drawing.

5.16.2 Member Function Documentation

5.16.2.1 GetCategories() List<[CategoryInfo](#)> LibTopoART.ICategoryAccess.GetCategories (long module_index = [LibTopoART_info.FINAL_MODULE](#))

This method collects information on the categories of a specified module.

Parameters

<i>module_index</i>	The index of the module information on the categories of which is to be returned.
---------------------	---

Returns

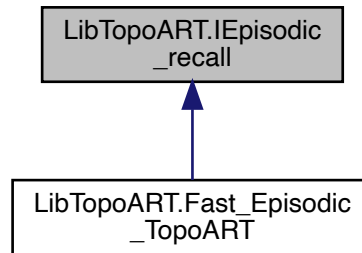
A list containing information about the respective categories.

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

5.17 LibTopoART.IEpisodic_recall Interface Reference

Interface summarising the episodic recall functionality.

Inheritance diagram for LibTopoART.IEpisodic_recall:



Public Member Functions

- long [BeginRecall](#) (decimal[] stimulus)
This method starts the recall process.
- bool [InterEpisodeRecallStep](#) (out decimal[] recall_result, out decimal F3_activation)
This method performs a single inter-episode recall step and sets the starting point for intra-episode recall.
- bool [IntraEpisodeRecallStep](#) (out decimal[] recall_result)
This method performs a single intra-episode recall step.
- void [EndRecall](#) ()
This method stops the recall process and frees temporary resources.

5.17.1 Detailed Description

Interface summarising the episodic recall functionality.

5.17.2 Member Function Documentation

5.17.2.1 [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.Fast_Episodic_TopoART](#).

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

This method stops the recall process and frees temporary resources.

Implemented in [LibTopoART.Fast_Episodic_TopoART](#).

5.17.2.3 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.Fast_Episodic_TopoART](#).

5.17.2.4 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

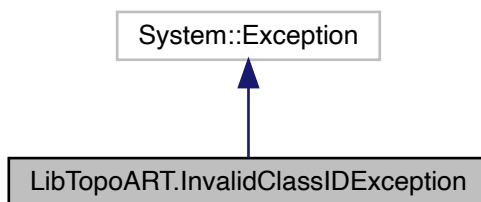
A boolean result indicating whether the recall step was successfully completed or not.

Implemented in [LibTopoART.Fast_Episodic_TopoART](#).

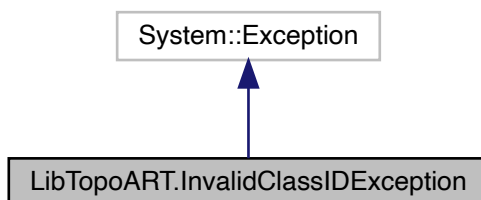
5.18 LibTopoART.InvalidClassIDException Class Reference

Exception signalling an invalid class ID.

Inheritance diagram for LibTopoART.InvalidClassIDException:



Collaboration diagram for LibTopoART.InvalidClassIDException:



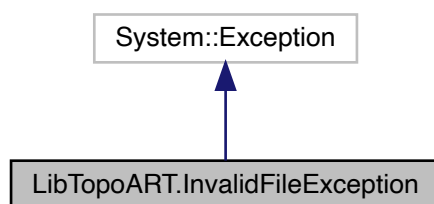
5.18.1 Detailed Description

Exception signalling an invalid class ID.

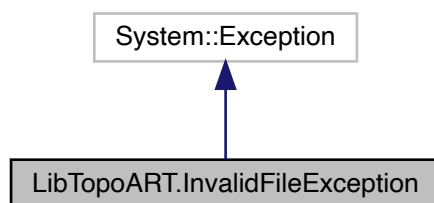
5.19 LibTopoART.InvalidFileException Class Reference

Exception signalling an invalid file.

Inheritance diagram for LibTopoART.InvalidFileException:



Collaboration diagram for LibTopoART.InvalidFileException:



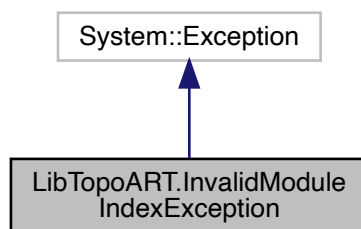
5.19.1 Detailed Description

Exception signalling an invalid file.

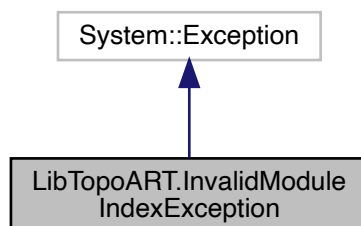
5.20 LibTopoART.InvalidModuleIndexException Class Reference

Exception signalling an invalid module index.

Inheritance diagram for LibTopoART.InvalidModuleIndexException:



Collaboration diagram for LibTopoART.InvalidModuleIndexException:



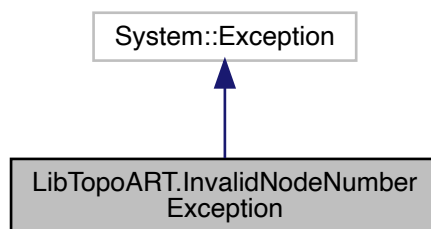
5.20.1 Detailed Description

Exception signalling an invalid module index.

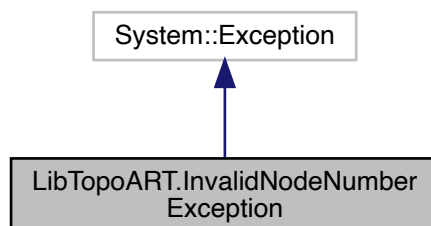
5.21 LibTopoART.InvalidNodeNumberException Class Reference

Exception signalling an invalid node number.

Inheritance diagram for LibTopoART.InvalidNodeNumberException:



Collaboration diagram for LibTopoART.InvalidNodeNumberException:



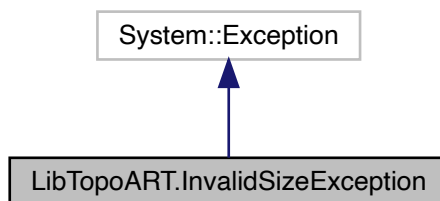
5.21.1 Detailed Description

Exception signalling an invalid node number.

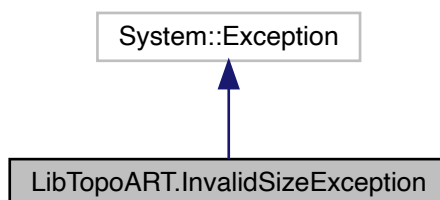
5.22 LibTopoART.InvalidSizeException Class Reference

Exception signalling an invalid size.

Inheritance diagram for LibTopoART.InvalidSizeException:



Collaboration diagram for LibTopoART.InvalidSizeException:



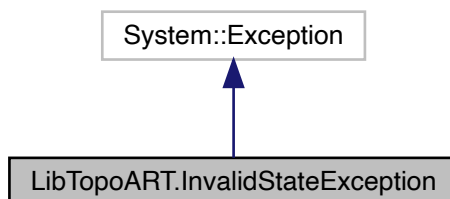
5.22.1 Detailed Description

Exception signalling an invalid size.

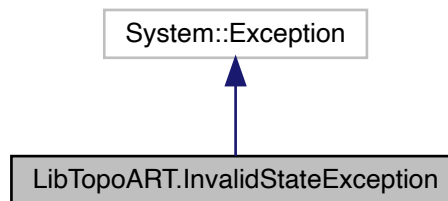
5.23 LibTopoART.InvalidStateException Class Reference

Exception signalling an invalid state of the neural network.

Inheritance diagram for LibTopoART.InvalidStateException:



Collaboration diagram for LibTopoART.InvalidStateException:



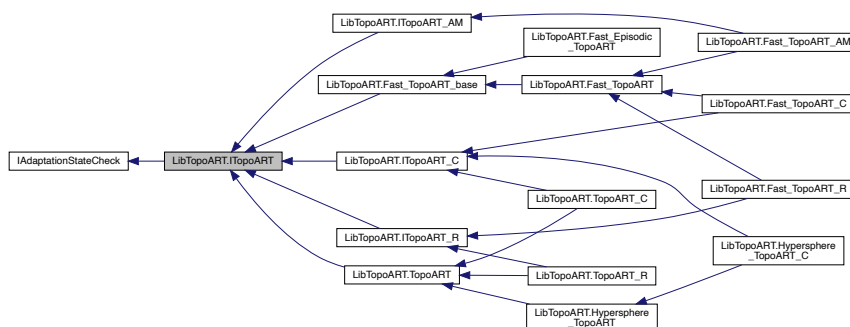
5.23.1 Detailed Description

Exception signalling an invalid state of the neural network.

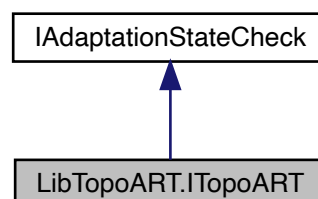
5.24 LibTopoART.ITopoART Interface Reference

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

Inheritance diagram for LibTopoART.ITopoART:



Collaboration diagram for LibTopoART.ITopoART:



Public Member Functions

- void [ComputeClusterIDs](#) ()
This method computes the cluster IDs for all neurons.
- [F2_output](#)[] [GetBMOutput](#) (decimal[] input)
This method finds the closest category for a given test input.
- [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]
- long[] [Phis](#) [get, set]
- decimal [Alpha](#) [get, set]
Property Alpha represents the choice parameter alpha.

5.24.1 Detailed Description

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

5.24.2 Member Function Documentation

5.24.2.1 [ComputeClusterIDs](#)() void [LibTopoART.ITopoART.ComputeClusterIDs](#) ()

This method computes the cluster IDs for all neurons.

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

5.24.2.2 [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`.

5.24.2.3 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`.

5.24.2.4 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_R`, `LibTopoART.Fast_TopoART_C`, `LibTopoART.Fast_Episodic_Top`, `LibTopoART.Fast_TopoART`, `LibTopoART.Hypersphere_Top`, `LibTopoART.TopoART_R`, `LibTopoART.TopoART_C`, and `LibTopoART.Fast_TopoART_base`.

5.24.2.5 Save() `void LibTopoART.ITopoART.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.
-------------	---

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

5.24.2.6 SaveText() `void LibTopoART.ITopoART.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.
-------------	---

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

5.24.3 Property Documentation

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

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

5.24.3.2 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.

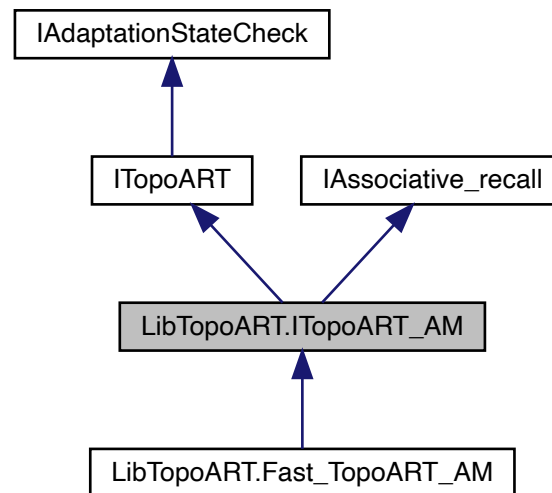
5.24.3.3 Phis `long [] LibTopoART.ITopoART.Phis [get], [set]`

Property `Phis` constitutes an extension of property `Phi` that enables individual values of `phi` for each module. By this, the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules can be controlled in a task dependent manner.

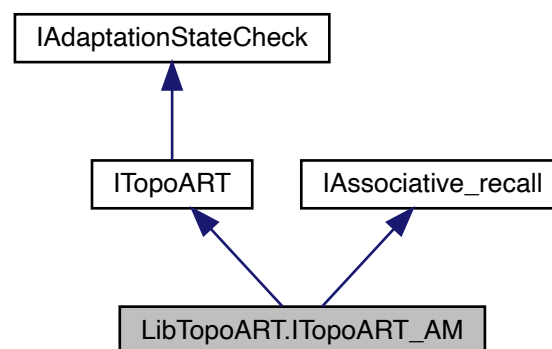
5.25 LibTopoART.ITopoART_AM Interface Reference

Interface summarising the TopoART-AM functionality.

Inheritance diagram for LibTopoART.ITopoART_AM:



Collaboration diagram for LibTopoART.ITopoART_AM:



Public Member Functions

- [F2_output\[\]](#) [GetBMOutput](#) (decimal[] key_1_vec, decimal[] key_2_vec)
This method finds the closest category for a given pair of keys.
- void [Learn](#) (decimal[] key_1_vec, decimal[] key_2_vec)
This method performs a single training step.

Properties

- long [Key_1_len](#) [get]
Property `K_1_len` returns the length of the first key vector.
- long [Key_2_len](#) [get]
Property `K_2_len` returns the length of the second key vector.

5.25.1 Detailed Description

Interface summarising the TopoART-AM functionality.

5.25.2 Member Function Documentation

5.25.2.1 GetBMOutput() [F2_output](#) [] `LibTopoART.ITopoART_AM.GetBMOutput (`
 `decimal[] key_1_vec,`
 `decimal[] key_2_vec)`

This method finds the closest category for a given pair of keys.

Parameters

<i>key_1_vec</i>	The first key vector.
<i>key_2_vec</i>	The second key vector corresponding to <i>key_1_vec</i> .

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-AM module.

Implemented in [LibTopoART.Fast_TopoART_AM](#).

5.25.2.2 Learn() `void LibTopoART.ITopoART_AM.Learn (`
 `decimal[] key_1_vec,`
 `decimal[] key_2_vec)`

This method performs a single training step.

Parameters

<i>key_1_vec</i>	The first key vector to be learnt.
<i>key_2_vec</i>	The second key vector corresponding to <i>key_1_vec</i> .

Implemented in [LibTopoART.Fast_TopoART_AM](#).

5.25.3 Property Documentation

5.25.3.1 Key_1_len `long LibTopoART.ITopoART_AM.Key_1_len [get]`

Property `K_1_len` returns the length of the first key vector.

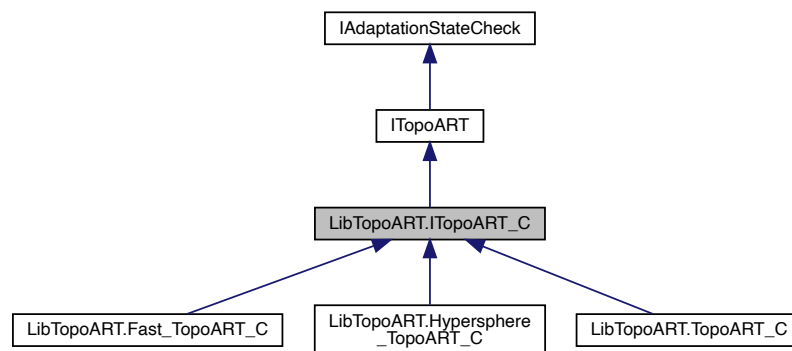
5.25.3.2 Key_2_len `long LibTopoART.ITopoART_AM.Key_2_len [get]`

Property `K_2_len` returns the length of the second key vector.

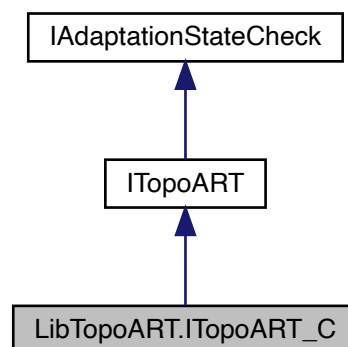
5.26 LibTopoART.ITopoART_C Interface Reference

Interface summarising the TopoART-C functionality.

Inheritance diagram for LibTopoART.ITopoART_C:



Collaboration diagram for LibTopoART.ITopoART_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

5.26.1 Detailed Description

Interface summarising the TopoART-C functionality.

5.26.2 Member Function Documentation

5.26.2.1 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](#).

5.26.2.2 Predict() [1/2] [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](#).

```
5.26.2.3 Predict() [2/2] 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

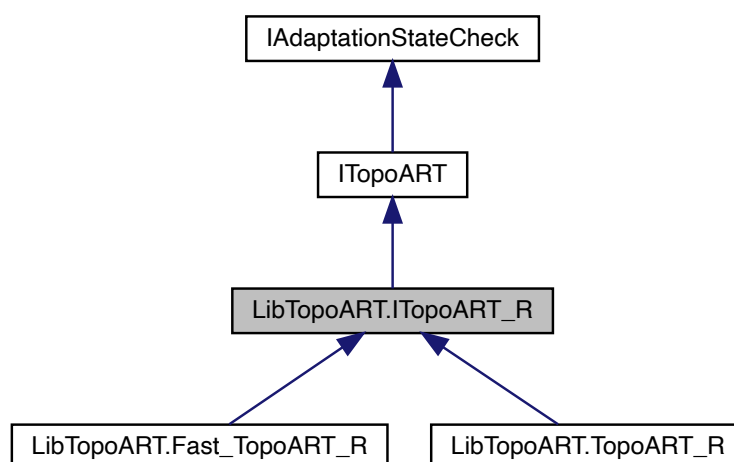
The predicted class ID.

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

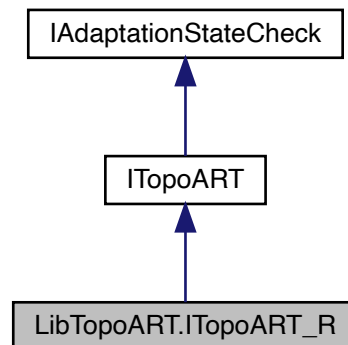
5.27 LibTopoART.ITopoART_R Interface Reference

Interface summarising the TopoART-R functionality.

Inheritance diagram for LibTopoART.ITopoART_R:



Collaboration diagram for LibTopoART.ITopoART_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`.

Properties

- long [D_len](#) [get]
Property `D_len` returns the length of the output vector (dependent variables).
- long [I_len](#) [get]
Property `I_len` returns the length of the input vector (independent variables).

5.27.1 Detailed Description

Interface summarising the TopoART-R functionality.

5.27.2 Member Function Documentation

5.27.2.1 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.Fast_TopoART_R](#), and [LibTopoART.TopoART_R](#).

5.27.2.2 Predict() [1/2] [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

<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, <i>nu</i> 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.Fast_TopoART_R](#), and [LibTopoART.TopoART_R](#).

5.27.2.3 Predict() [2/2] 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, <i>nu</i> 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.Fast_TopoART_R](#), and [LibTopoART.TopoART_R](#).

5.27.3 Property Documentation

5.27.3.1 D_len long LibTopoART.ITopoART_R.D_len [get]

Property `D_len` returns the length of the output vector (dependent variables).

5.27.3.2 I_len long LibTopoART.ITopoART_R.I_len [get]

Property `I_len` returns the length of the input vector (independent variables).

5.28 LibTopoART.LibTopoART_info Struct Reference

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

Static Public Attributes

- const decimal [version](#) = 0.91m
Instance variable `version` represents the version of [LibTopoART](#).
- static readonly string[] [networks](#)
Instance variable `networks` provides a string array containing the networks implemented in the current version of [LibTopoART](#) and the corresponding class names.
- const long [UNDEFINED](#) = -1
Instance variable `UNDEFINED` gives the value used for indicating undefined and uninitialised variables.
- const long [FINAL_MODULE](#) = [UNDEFINED](#)
Instance variable `FINAL_MODULE` gives the value used for indicating that the [TopoART](#) module with the highest index is to be used.

5.28.1 Detailed Description

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

5.28.2 Member Data Documentation

5.28.2.1 FINAL_MODULE `const long LibTopoART.LibTopoART_info.FINAL_MODULE = UNDEFINED [static]`

Instance variable `FINAL_MODULE` gives the value used for indicating that the [TopoART](#) module with the highest index is to be used.

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

Initial value:

```
= {
    "Episodic TopoART (class Fast_Episodic_TopoART)",
    "Hypersphere TopoART (class Hypersphere_TopoART)",
    "Hypersphere TopoART-C (class Hypersphere_TopoART_C)",
    "TopoART (class TopoART, class Fast_TopoART)",
    "TopoART-AM (class Fast_TopoART-AM)",
    "TopoART-C (class TopoART_C, class Fast_TopoART_C)",
    "TopoART-R (class TopoART_R, class Fast_TopoART_R)"
}
```

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

5.28.2.3 UNDEFINED `const long LibTopoART.LibTopoART_info.UNDEFINED = -1 [static]`

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

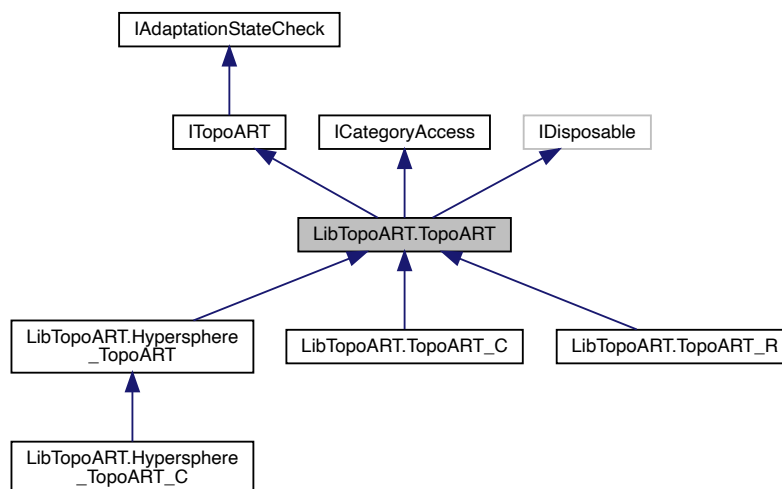
5.28.2.4 version `const decimal LibTopoART.LibTopoART_info.version = 0.91m [static]`

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

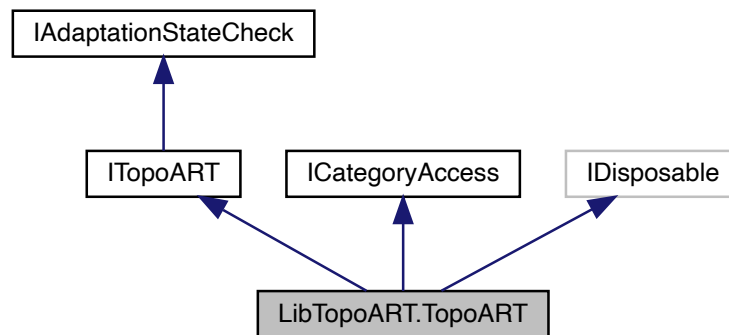
5.29 LibTopoART.TopoART Class Reference

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."

Inheritance diagram for LibTopoART.TopoART:



Collaboration diagram for LibTopoART.TopoART:



Public Member Functions

- `TopoART` (long input_length, 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.
- List< `CategoryInfo` > `GetCategories` (long module_index=FINAL_MODULE)
This method collects information on the categories of a specified module.

Static Public Attributes

- const long `FINAL_MODULE` = LibTopoART_info.FINAL_MODULE
Instance variable `FINAL_MODULE` gives the value used for indicating that the `TopoART` module with the highest index is to be used.

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 [InputLen](#) [get]
Property [InputLen](#) returns the length of the input vector.
- 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]
- long[] [Phis](#) [get, set]
- 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.
- 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](#).

5.29.1 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]`.

5.29.2 Constructor & Destructor Documentation

5.29.2.1 TopoART() [1/2] `LibTopoART.TopoART.TopoART (`
`long input_length,`
`long module_number,`
`decimal rho_a)`

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

Parameters

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

5.29.2.2 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.
-------------	--

Exceptions

InvalidFileException	Throws when the given file cannot be loaded.
--------------------------------------	--

5.29.3 Member Function Documentation

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

This method computes the cluster IDs for all neurons.

Implements [LibTopoART.ITopoART](#).

5.29.3.2 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.

5.29.3.3 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.
------------------	---

5.29.3.4 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.

Exceptions

InvalidStateException	Throws when the network is in an invalid state.
---------------------------------------	---

Implements [LibTopoART.IAdaptationStateCheck](#).

5.29.3.5 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](#).

```
5.29.3.6 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](#).

```
5.29.3.7 GetCategories() List<CategoryInfo> LibTopoART.TopoART.GetCategories (
    long module_index = FINAL\_MODULE )
```

This method collects information on the categories of a specified module.

Parameters

<i>module_index</i>	The index of the module the categories of which are to be analysed.
---------------------	---

Returns

A list containing information about the respective categories.

Exceptions

InvalidModuleIndexException	Throws when <i>module_index</i> is invalid.
InvalidNodeNumberException	Throws when the number of nodes of a module is greater than <code>int.MaxValue</code> .

Implements [LibTopoART.ICategoryAccess](#).

5.29.3.8 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](#).

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

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

Implements [LibTopoART.IAdaptationStateCheck](#).

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

This method saves the entire network as a binary file.

Parameters

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

Implements [LibTopoART.ITopoART](#).

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

This method saves the entire network as a text file.

Parameters

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

Implements [LibTopoART.ITopoART](#).

5.29.4 Member Data Documentation

5.29.4.1 FINAL_MODULE `const long LibTopoART.TopoART.FINAL_MODULE = LibTopoART_info.FINAL_MODULE [static]`

Instance variable `FINAL_MODULE` gives the value used for indicating that the [TopoART](#) module with the highest index is to be used.

5.29.5 Property Documentation

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

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

5.29.5.2 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.

5.29.5.3 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.

5.29.5.4 Phi `long LibTopoART.TopoART.Phi [get], [set]`

Property `Phi` represents the parameter `phi` required for the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules.

5.29.5.5 Phis `long [] LibTopoART.TopoART.Phis [get], [set]`

Property `Phis` constitutes an extension of property `Phi` that enables individual values of `phi` for each module. By this, the removal of nodes and edges as well as for the propagation of input to subsequent [TopoART](#) modules can be controlled in a task dependent manner.

Exceptions

InvalidSizeException	Throws when the array length does not fit the module number.
--------------------------------------	--

5.29.5.6 TopoARTFileFormatVersion decimal LibTopoART.TopoART.TopoARTFileFormatVersion [get]

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

5.30 LibTopoART.TopoART_AM Class Reference

Class [TopoART_AM](#) is obsolete. Please use class [Fast_TopoART_AM](#) instead!

5.30.1 Detailed Description

Class [TopoART_AM](#) is obsolete. Please use class [Fast_TopoART_AM](#) instead!

5.31 LibTopoART_samples.TopoART_AM_sample1 Class Reference

Sample using TopoART-AM with synthetic two-dimensional data. [C#]

5.31.1 Detailed Description

Sample using TopoART-AM with synthetic two-dimensional data. [C#]

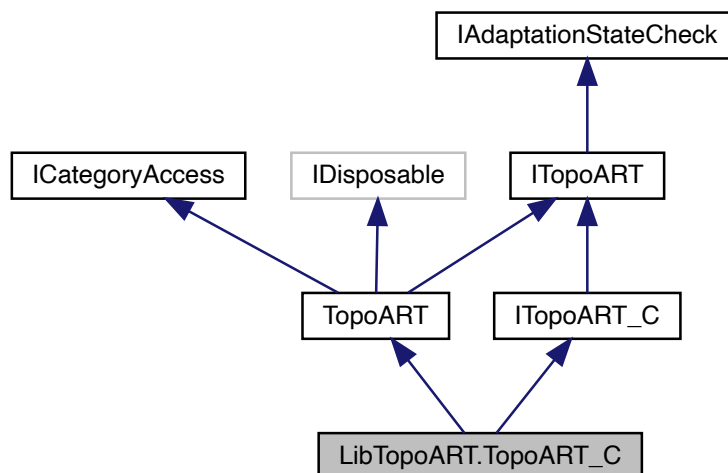
A TopoART-AM network is trained with the well-known Two Spirals dataset augmented with additional information. The resulting network maps two-dimensional points lying on each spiral (`key_1`) to their Euclidean distance from the origin and the corresponding spiral ID (`key_2`). Therefore, it can recall spiral points if a distance and a spiral ID are given, and vice versa.

The resulting network can be visualised using the R script `ShowTopoARTAMResults.R` provided in the sub-folder R.

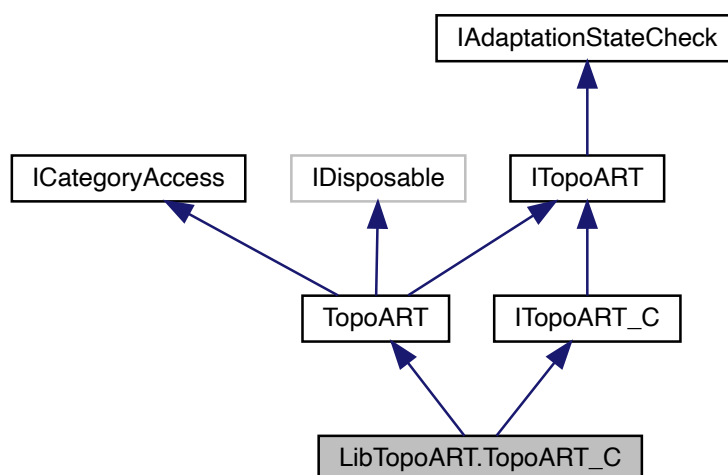
5.32 LibTopoART.TopoART_C Class Reference

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."

Inheritance diagram for LibTopoART.TopoART_C:



Collaboration diagram for LibTopoART.TopoART_C:



Public Member Functions

- [TopoART_C](#) (long input_length, 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.

Static 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

5.32.1 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.

5.32.2 Constructor & Destructor Documentation

5.32.2.1 TopoART_C() [1/2] `LibTopoART.TopoART_C.TopoART_C (`
 long *input_length*,
 long *module_number*,
 decimal *rho_a*)

This constructor initialises a TopoART-C network.

Parameters

<i>input_length</i>	The length 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).

5.32.2.2 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.
-------------	--------------------------------------

Exceptions

<i>InvalidFileException</i>	Throws when the given file cannot be loaded.
---	--

5.32.3 Member Function Documentation

5.32.3.1 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](#).

5.32.3.2 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)

Exceptions

<i>InvalidClassIDException</i>	Throws when <i>class_ID</i> is less than 0.
--	---

Implements [LibTopoART.ITopoART_C](#).

5.32.3.3 Predict() [1/2] [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](#).

5.32.3.4 Predict() [2/2] 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](#).

5.32.4 Member Data Documentation

5.32.4.1 UNDEFINED_CLASS_ID `const long LibTopoART.TopoART_C.UNDEFINED_CLASS_ID = -2 [static]`

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.

5.32.5 Property Documentation

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

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

5.33 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

- readonly long `class_ID`

Instance variable `class_ID` gives the predicted class ID.

- readonly decimal `confidence`

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

5.33.1 Detailed Description

Struct `TopoART_C_Prediction` contains a prediction made by a TopoART-C network.

5.33.2 Constructor & Destructor Documentation

5.33.2.1 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.

5.33.3 Member Data Documentation

5.33.3.1 class_ID readonly long LibTopoART.TopoART_C_Prediction.class_ID

Instance variable `class_ID` gives the predicted class ID.

5.33.3.2 confidence readonly decimal LibTopoART.TopoART_C_Prediction.confidence

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

5.34 LibTopoART_samples.TopoART_C_sample1 Class Reference

Simple classification sample. [C#]

5.34.1 Detailed Description

Simple classification sample. [C#]

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

5.35 LibTopoART_samples.TopoART_C_sample2 Class Reference

Classification sample using more complex synthetic two-dimensional data with associated class IDs. [C#]

5.35.1 Detailed Description

Classification sample using more complex synthetic two-dimensional data with associated class IDs. [C#]

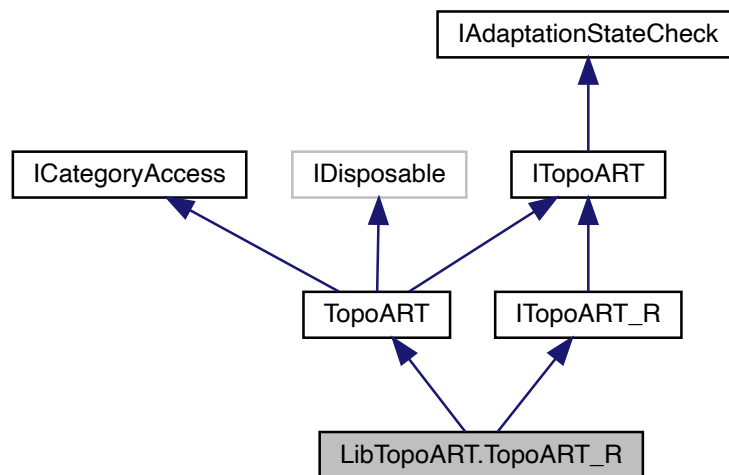
Train TopoART-C with a two-dimensional dataset similar to the one used in "Marko Tscherepanow and Sören Riechers (2012). An Incremental On-line Classifier for Imbalanced, Incomplete, and Noisy Data. In Proceedings of the European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental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

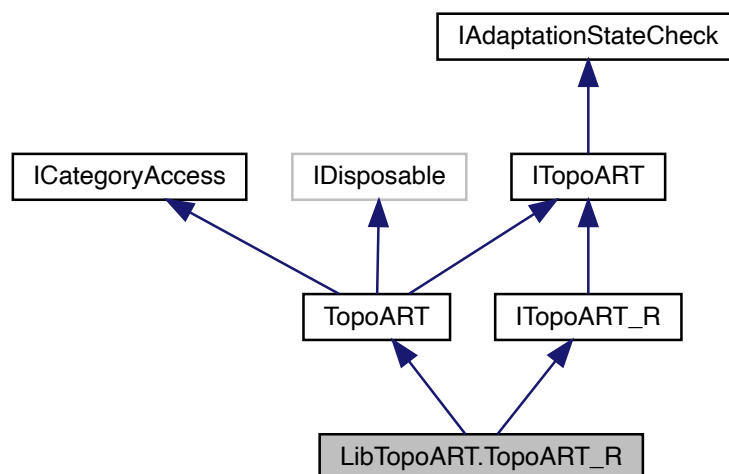
5.36 LibTopoART.TopoART_R Class Reference

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."

Inheritance diagram for LibTopoART.TopoART_R:



Collaboration diagram for LibTopoART.TopoART_R:



Public Member Functions

- [TopoART_R](#) (long i_length, long d_length, 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

- long [D_len](#) [get]
Property D_len returns the length of the output vector (dependent variables).
- new decimal [FileFormatVersion](#) [get]
Property FileFormatVersion returns the version of the file format used by class TopoART_R.
- long [I_len](#) [get]
Property I_len returns the length of the input vector (independent variables).

Additional Inherited Members

5.36.1 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].

5.36.2 Constructor & Destructor Documentation

5.36.2.1 TopoART_R() [1/2] `LibTopoART.TopoART_R.TopoART_R (`
 long i_length,
 long d_length,
 long module_number,
 decimal rho_a)

This constructor initialises a TopoART-R network.

Parameters

<i>i_length</i>	The length of the input vector (independent variables) to be learnt.
<i>d_length</i>	The length 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).

5.36.2.2 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.
-------------	--------------------------------------

Exceptions

<i>InvalidFileException</i>	Throws when the given file cannot be loaded.
---	--

5.36.3 Member Function Documentation

5.36.3.1 Learn() [1/2] `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](#).

5.36.3.2 Learn() [2/2] `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](#).

5.36.3.3 Predict() [1/2] [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](#).

5.36.3.4 Predict() [2/2] 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](#).

5.36.4 Property Documentation

5.36.4.1 D_len long LibTopoART.TopoART_R.D_len [get]

Property `D_len` returns the length of the output vector (dependent variables).

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

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

5.36.4.3 I_len long LibTopoART.TopoART_R.I_len [get]

Property `I_len` returns the length of the input vector (independent variables).

5.37 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

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

Static Public Attributes

- 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.

5.37.1 Detailed Description

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

5.37.2 Constructor & Destructor Documentation

5.37.2.1 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

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

5.37.3 Member Function Documentation

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

This method prints the predictions on the console.

5.37.4 Member Data Documentation

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

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

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

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

5.37.4.3 NO_PREDICTION `const decimal LibTopoART.TopoART_R_Prediction.NO_PREDICTION = LibTopoART_info.UNDEFI`
[static]

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

5.38 LibTopoART_samples.TopoART_R_sample1 Class Reference

Regression sample using TopoART-R. (simplified version) [C#]

5.38.1 Detailed Description

Regression sample using TopoART-R. (simplified version) [C#]

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`.

5.39 LibTopoART_samples.TopoART_R_sample2 Class Reference

Regression sample using TopoART-R. (advanced version) [C#]

5.39.1 Detailed Description

Regression sample using TopoART-R. (advanced version) [C#]

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`.

5.40 LibTopoART_samples.TopoART_R_sample3 Class Reference

Pixel-wise regression analysis of an image using TopoART-R. [F#]

5.40.1 Detailed Description

Pixel-wise regression analysis of an image using TopoART-R. [F#]

This sample trains a TopoART-R network with a colour image depicting the Mandelbrot set. The pixel coordinates are used as independent variables and the corresponding colour values as dependent variables. The accuracy of the regression function can be controlled by means of the vigilance parameter `rho_a`.

The training image `Mandelbrot_reference.png` and the predicted image `TopoART-R_Mandelbrot←_regression.png` are written into the subfolder `results/regression`.

5.41 LibTopoART_samples.TopoART_sample1 Class Reference

Simple clustering sample. [C#]

5.41.1 Detailed Description

Simple clustering sample. [C#]

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.

5.42 LibTopoART_samples.TopoART_sample2 Class Reference

Clustering sample using more complex synthetic two-dimensional data. [C#]

5.42.1 Detailed Description

Clustering sample using more complex synthetic two-dimensional data. [C#]

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.

5.43 LibTopoART_samples.TopoART_sample3 Class Reference

Clustering sample using very noisy synthetic two-dimensional data. [VB]

5.43.1 Detailed Description

Clustering sample using very noisy synthetic two-dimensional data. [VB]

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." The dataset applied here comprises 1,000,000 samples equally allotted to six clusters (each containing one sixth of the samples). Additionally, 1,000,000 uniformly distributed random samples are added. Finally, all samples are randomly shuffled.

Due to the randomness involved, the results differ between different runs of this program. However, they are qualitatively comparable: The first module creates a coarse clustering of the data while the second module refines it to the six clusters of the undisturbed portion of the dataset.

These results show the abilities of TopoART to cope with a high amount of noise data and produce stable results independent of the sample order.

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, [9](#)
- ADAPTED_NONPERMANENT_WEIGHT
 - LibTopoART, [9](#)
- ADAPTED_PERMANENT_WEIGHT
 - LibTopoART, [9](#)
- ADDED_EDGE_CANDIDATE
 - LibTopoART, [9](#)
- ADDED_NODE_CANDIDATE
 - LibTopoART, [9](#)
- ADDED_PERMANENT_EDGE
 - LibTopoART, [9](#)
- ADDED_PERMANENT_NODE
 - LibTopoART, [9](#)
- ANY_NONPERMANENT_ADAPTATION_MASK
 - LibTopoART, [9](#)
- ANY_PERMANENT_ADAPTATION_MASK
 - LibTopoART, [9](#)
- BeginRecall
 - LibTopoART.Fast_Episodic_TopoART, [16](#)
 - LibTopoART.IEpisodic_recall, [56](#)
- BeginRecallKey1
 - LibTopoART.Fast_TopoART_AM, [23](#)
 - LibTopoART.IAssociative_recall, [53](#)
- BeginRecallKey2
 - LibTopoART.Fast_TopoART_AM, [24](#)
 - LibTopoART.IAssociative_recall, [54](#)
- bm_cluster_ID
 - LibTopoART.F2_output, [13](#)
- bm_node_activation
 - LibTopoART.F2_output, [13](#)
- bm_node_ID
 - LibTopoART.F2_output, [14](#)
- bm_permanent_cluster_ID
 - LibTopoART.F2_output, [14](#)
- bm_permanent_node_activation
 - LibTopoART.F2_output, [14](#)
- bm_permanent_node_ID
 - LibTopoART.F2_output, [14](#)
- CategoryInfo
 - LibTopoART.CategoryInfo, [11](#)
- class_ID
 - LibTopoART.CategoryInfo, [11](#)
 - LibTopoART.TopoART_C_Prediction, [89](#)
- cluster_ID
 - LibTopoART.CategoryInfo, [11](#)
- ComputeClusterIDs
 - LibTopoART.Fast_TopoART_base, [28](#)
 - LibTopoART.ITopoART, [64](#)
 - LibTopoART.TopoART, [78](#)
- confidence
 - LibTopoART.TopoART_C_Prediction, [89](#)
- D_len
 - LibTopoART.Fast_TopoART_R, [43](#)
 - LibTopoART.ITopoART_R, [74](#)
 - LibTopoART.TopoART_R, [94](#)
- d_vec_prediction
 - LibTopoART.TopoART_R_Prediction, [95](#)
- Dispose
 - LibTopoART.Fast_TopoART_base, [29](#)
 - LibTopoART.TopoART, [78, 79](#)
- EndRecall
 - LibTopoART.Fast_Episodic_TopoART, [17](#)
 - LibTopoART.Fast_TopoART_AM, [24](#)
 - LibTopoART.IAssociative_recall, [54](#)
 - LibTopoART.IEpisodic_recall, [57](#)
- F2_output
 - LibTopoART.F2_output, [13](#)
- Fast_Episodic_TopoART
 - LibTopoART.Fast_Episodic_TopoART, [16](#)
- Fast_TopoART
 - LibTopoART.Fast_TopoART, [20](#)
- Fast_TopoART_AM
 - LibTopoART.Fast_TopoART_AM, [23](#)
- Fast_TopoART_C
 - LibTopoART.Fast_TopoART_C, [35, 36](#)
- Fast_TopoART_R
 - LibTopoART.Fast_TopoART_R, [40, 41](#)
- FileFormatVersion
 - LibTopoART.Fast_Episodic_TopoART, [18](#)
 - LibTopoART.Fast_TopoART_AM, [26](#)
 - LibTopoART.Fast_TopoART_base, [32](#)
 - LibTopoART.Fast_TopoART_C, [38](#)
 - LibTopoART.Fast_TopoART_R, [43](#)
 - LibTopoART.Hypersphere_TopoART, [46](#)
 - LibTopoART.Hypersphere_TopoART_C, [51](#)
 - LibTopoART.TopoART, [82](#)
 - LibTopoART.TopoART_C, [88](#)
 - LibTopoART.TopoART_R, [94](#)
- FINAL_MODULE
 - LibTopoART.Fast_TopoART_base, [32](#)
 - LibTopoART.LibTopoART_info, [74](#)
 - LibTopoART.TopoART, [82](#)
- FloatBaseType
 - LibTopoART.Fast_TopoART_base, [32](#)
 - LibTopoART.TopoART, [82](#)
- GetAdaptationState
 - LibTopoART.Fast_TopoART_base, [29](#)
 - LibTopoART.IAdaptationStateCheck, [52](#)
 - LibTopoART.TopoART, [79](#)
- GetBMOutput
 - LibTopoART.Fast_TopoART_AM, [25](#)
 - LibTopoART.Fast_TopoART_base, [29, 30](#)
 - LibTopoART.ITopoART, [64, 65](#)
 - LibTopoART.ITopoART_AM, [68](#)
 - LibTopoART.TopoART, [79, 80](#)

- GetCategories
 - LibTopoART.Fast_TopoART_base, 30
 - LibTopoART.ICategoryAccess, 55
 - LibTopoART.TopoART, 80
- Hypersphere_TopoART
 - LibTopoART.Hypersphere_TopoART, 45
- Hypersphere_TopoART_C
 - LibTopoART.Hypersphere_TopoART_C, 48, 49
- HypersphereTopoARTFileFormatVersion
 - LibTopoART.Hypersphere_TopoART, 46
- I_len
 - LibTopoART.Fast_TopoART_R, 43
 - LibTopoART.ITopoART_R, 74
 - LibTopoART.TopoART_R, 94
- i_vec_prediction
 - LibTopoART.TopoART_R_Prediction, 95
- IntegerBaseType
 - LibTopoART.Fast_TopoART_base, 32
 - LibTopoART.TopoART, 82
- InterEpisodeRecallStep
 - LibTopoART.Fast_Episodic_TopoART, 17
 - LibTopoART.IEpisodic_recall, 57
- IntraEpisodeRecallStep
 - LibTopoART.Fast_Episodic_TopoART, 17
 - LibTopoART.IEpisodic_recall, 57
- Key_1_len
 - LibTopoART.Fast_TopoART_AM, 26
 - LibTopoART.ITopoART_AM, 69
- Key_2_len
 - LibTopoART.Fast_TopoART_AM, 26
 - LibTopoART.ITopoART_AM, 69
- Learn
 - LibTopoART.Fast_Episodic_TopoART, 18
 - LibTopoART.Fast_TopoART, 21
 - LibTopoART.Fast_TopoART_AM, 25
 - LibTopoART.Fast_TopoART_base, 31
 - LibTopoART.Fast_TopoART_C, 36
 - LibTopoART.Fast_TopoART_R, 41
 - LibTopoART.Hypersphere_TopoART_C, 49
 - LibTopoART.ITopoART, 65
 - LibTopoART.ITopoART_AM, 68
 - LibTopoART.ITopoART_C, 70
 - LibTopoART.ITopoART_R, 72
 - LibTopoART.TopoART, 81
 - LibTopoART.TopoART_C, 86
 - LibTopoART.TopoART_R, 92
- LibTopoART, 7
 - AdaptationState, 9
 - ADAPTED_NONPERMANENT_WEIGHT, 9
 - ADAPTED_PERMANENT_WEIGHT, 9
 - ADDED_EDGE_CANDIDATE, 9
 - ADDED_NODE_CANDIDATE, 9
 - ADDED_PERMANENT_EDGE, 9
 - ADDED_PERMANENT_NODE, 9
 - ANY_NONPERMANENT_ADAPTATION_MASK, 9
 - ANY_PERMANENT_ADAPTATION_MASK, 9
 - NO_ADAPTATION, 9
 - REMOVED_EDGE_CANDIDATE, 9
 - REMOVED_NODE_CANDIDATE, 9
- LibTopoART.CategoryInfo, 10
 - CategoryInfo, 11
 - class_ID, 11
 - cluster_ID, 11
 - spatial_weights, 11
 - temporal_weights, 11
- LibTopoART.Episodic_TopoART, 12
- LibTopoART.F2_output, 12
 - bm_cluster_ID, 13
 - bm_node_activation, 13
 - bm_node_ID, 14
 - bm_permanent_cluster_ID, 14
 - bm_permanent_node_activation, 14
 - bm_permanent_node_ID, 14
 - F2_output, 13
- LibTopoART.Fast_Episodic_TopoART, 14
 - BeginRecall, 16
 - EndRecall, 17
 - Fast_Episodic_TopoART, 16
 - FileFormatVersion, 18
 - InterEpisodeRecallStep, 17
 - IntraEpisodeRecallStep, 17
 - Learn, 18
- LibTopoART.Fast_TopoART, 18
 - Fast_TopoART, 20
 - Learn, 21
- LibTopoART.Fast_TopoART_AM, 21
 - BeginRecallKey1, 23
 - BeginRecallKey2, 24
 - EndRecall, 24
 - Fast_TopoART_AM, 23
 - FileFormatVersion, 26
 - GetBMOutput, 25
 - Key_1_len, 26
 - Key_2_len, 26
 - Learn, 25
 - RecallStep, 25
- LibTopoART.Fast_TopoART_base, 26
 - ComputeClusterIDs, 28
 - Dispose, 29
 - FileFormatVersion, 32
 - FINAL_MODULE, 32
 - FloatBaseType, 32
 - GetAdaptationState, 29
 - GetBMOutput, 29, 30
 - GetCategories, 30
 - IntegerBaseType, 32
 - Learn, 31
 - Phi, 33
 - Phis, 33
 - ResetAdaptationState, 31
 - Save, 31
 - SaveText, 32
 - TopoARTFileFormatVersion, 33

- LibTopoART.Fast_TopoART_C, 33
 - Fast_TopoART_C, 35, 36
 - FileFormatVersion, 38
 - Learn, 36
 - Predict, 37
 - UNDEFINED_CLASS_ID, 38
- LibTopoART.Fast_TopoART_R, 38
 - D_len, 43
 - Fast_TopoART_R, 40, 41
 - FileFormatVersion, 43
 - I_len, 43
 - Learn, 41
 - Predict, 42
- LibTopoART.Hypersphere_TopoART, 43
 - FileFormatVersion, 46
 - Hypersphere_TopoART, 45
 - HypersphereTopoARTFileFormatVersion, 46
- LibTopoART.Hypersphere_TopoART_C, 46
 - FileFormatVersion, 51
 - Hypersphere_TopoART_C, 48, 49
 - Learn, 49
 - Predict, 50
 - UNDEFINED_CLASS_ID, 51
- LibTopoART.IAdaptationStateCheck, 51
 - GetAdaptationState, 52
 - ResetAdaptationState, 52
- LibTopoART.IAssociative_recall, 52
 - BeginRecallKey1, 53
 - BeginRecallKey2, 54
 - EndRecall, 54
 - RecallStep, 54
- LibTopoART.ICategoryAccess, 55
 - GetCategories, 55
- LibTopoART.IEpisodic_recall, 56
 - BeginRecall, 56
 - EndRecall, 57
 - InterEpisodeRecallStep, 57
 - IntraEpisodeRecallStep, 57
- LibTopoART.InvalidClassIDException, 58
- LibTopoART.InvalidFileException, 58
- LibTopoART.InvalidModuleIndexException, 59
- LibTopoART.InvalidNodeNumberException, 60
- LibTopoART.InvalidSizeException, 61
- LibTopoART.InvalidStateException, 62
- LibTopoART.ITopoART, 63
 - ComputeClusterIDs, 64
 - GetBMOutput, 64, 65
 - Learn, 65
 - ModuleNum, 66
 - Phi, 66
 - Phis, 66
 - Save, 65
 - SaveText, 66
- LibTopoART.ITopoART_AM, 67
 - GetBMOutput, 68
 - Key_1_len, 69
 - Key_2_len, 69
 - Learn, 68
- LibTopoART.ITopoART_C, 69
 - Learn, 70
 - Predict, 70, 71
- LibTopoART.ITopoART_R, 71
 - D_len, 74
 - I_len, 74
 - Learn, 72
 - Predict, 73
- LibTopoART.LibTopoART_info, 74
 - FINAL_MODULE, 74
 - networks, 75
 - UNDEFINED, 75
 - version, 75
- LibTopoART.TopoART, 75
 - ComputeClusterIDs, 78
 - Dispose, 78, 79
 - FileFormatVersion, 82
 - FINAL_MODULE, 82
 - FloatBaseType, 82
 - GetAdaptationState, 79
 - GetBMOutput, 79, 80
 - GetCategories, 80
 - IntegerBaseType, 82
 - Learn, 81
 - Phi, 82
 - Phis, 82
 - ResetAdaptationState, 81
 - Save, 81
 - SaveText, 81
 - TopoART, 78
 - TopoARTFileFormatVersion, 83
- LibTopoART.TopoART_AM, 83
- LibTopoART.TopoART_C, 83
 - FileFormatVersion, 88
 - Learn, 86
 - Predict, 87
 - TopoART_C, 85, 86
 - UNDEFINED_CLASS_ID, 87
- LibTopoART.TopoART_C_Prediction, 88
 - class_ID, 89
 - confidence, 89
 - TopoART_C_Prediction, 88
- LibTopoART.TopoART_R, 90
 - D_len, 94
 - FileFormatVersion, 94
 - I_len, 94
 - Learn, 92
 - Predict, 93
 - TopoART_R, 91, 92
- LibTopoART.TopoART_R_Prediction, 94
 - d_vec_prediction, 95
 - i_vec_prediction, 95
 - NO_PREDICTION, 95
 - PrintPredictions, 95
 - TopoART_R_Prediction, 95
- LibTopoART_samples, 9
- LibTopoART_samples.Episodic_TopoART_sample1, 12
- LibTopoART_samples.Episodic_TopoART_sample2, 12

- LibTopoART_samples.TopoART_AM_sample1, [83](#)
- LibTopoART_samples.TopoART_C_sample1, [89](#)
- LibTopoART_samples.TopoART_C_sample2, [89](#)
- LibTopoART_samples.TopoART_R_sample1, [96](#)
- LibTopoART_samples.TopoART_R_sample2, [96](#)
- LibTopoART_samples.TopoART_R_sample3, [96](#)
- LibTopoART_samples.TopoART_sample1, [97](#)
- LibTopoART_samples.TopoART_sample2, [97](#)
- LibTopoART_samples.TopoART_sample3, [97](#)
- ModuleNum
 - LibTopoART.ITopoART, [66](#)
- networks
 - LibTopoART.LibTopoART_info, [75](#)
- NO_ADAPTATION
 - LibTopoART, [9](#)
- NO_PREDICTION
 - LibTopoART.TopoART_R_Prediction, [95](#)
- Phi
 - LibTopoART.Fast_TopopART_base, [33](#)
 - LibTopoART.ITopoART, [66](#)
 - LibTopoART.TopoART, [82](#)
- Phis
 - LibTopoART.Fast_TopopART_base, [33](#)
 - LibTopoART.ITopoART, [66](#)
 - LibTopoART.TopoART, [82](#)
- Predict
 - LibTopoART.Fast_TopopART_C, [37](#)
 - LibTopoART.Fast_TopopART_R, [42](#)
 - LibTopoART.Hypersphere_TopopART_C, [50](#)
 - LibTopoART.ITopoART_C, [70](#), [71](#)
 - LibTopoART.ITopoART_R, [73](#)
 - LibTopoART.TopoART_C, [87](#)
 - LibTopoART.TopoART_R, [93](#)
- PrintPredictions
 - LibTopoART.TopoART_R_Prediction, [95](#)
- RecallStep
 - LibTopoART.Fast_TopopART_AM, [25](#)
 - LibTopoART.IAssociative_recall, [54](#)
- REMOVED_EDGE_CANDIDATE
 - LibTopoART, [9](#)
- REMOVED_NODE_CANDIDATE
 - LibTopoART, [9](#)
- ResetAdaptationState
 - LibTopoART.Fast_TopopART_base, [31](#)
 - LibTopoART.IAdaptationStateCheck, [52](#)
 - LibTopoART.TopoART, [81](#)
- Save
 - LibTopoART.Fast_TopopART_base, [31](#)
 - LibTopoART.ITopoART, [65](#)
 - LibTopoART.TopoART, [81](#)
- SaveText
 - LibTopoART.Fast_TopopART_base, [32](#)
 - LibTopoART.ITopoART, [66](#)
 - LibTopoART.TopoART, [81](#)
- spatial_weights
 - LibTopoART.CategoryInfo, [11](#)
- temporal_weights
 - LibTopoART.CategoryInfo, [11](#)
- TopoART
 - LibTopoART.TopoART, [78](#)
- TopoART_C
 - LibTopoART.TopoART_C, [85](#), [86](#)
- TopoART_C_Prediction
 - LibTopoART.TopoART_C_Prediction, [88](#)
- TopoART_R
 - LibTopoART.TopoART_R, [91](#), [92](#)
- TopoART_R_Prediction
 - LibTopoART.TopoART_R_Prediction, [95](#)
- TopoARTFileFormatVersion
 - LibTopoART.Fast_TopopART_base, [33](#)
 - LibTopoART.TopoART, [83](#)
- UNDEFINED
 - LibTopoART.LibTopoART_info, [75](#)
- UNDEFINED_CLASS_ID
 - LibTopoART.Fast_TopopART_C, [38](#)
 - LibTopoART.Hypersphere_TopopART_C, [51](#)
 - LibTopoART.TopoART_C, [87](#)
- version
 - LibTopoART.LibTopoART_info, [75](#)