LibTopoART

v0.50

Generated by Doxygen 1.8.5

Fri Nov 1 2013 14:37:35

Contents

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

LibTopoART.Episodic_TopoART

Class `Episodic_TopoART` provides an implementation of the Episodic TopoART neural network as proposed in "Marko Tscherepanow, Sina Kühnel, and Sören Riechers, 'Episodic Clustering of Data Streams Using a Topology-Learning Neural Network', European Conference on Artificial Intelligence (ECAI), Workshop on Active and Incremental Learning (AIL), pp. 24-29, 2012." 9

LibTopoART_samples.Episodic_TopoART_sample1

Sample using artificial two-dimensional data 11

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. 11

LibTopoART.Fast_TopoART

Class `Fast_TopoART` provides an implementation of the TopoART neural network as proposed in "Tscherepanow, Marko (2010). TopoART: A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer." 12

LibTopoART.Fast_TopoART_Base

Base class providing functionality common to several TopoART networks. 14

LibTopoART.ITopoART

Interface summarising the basic TopoART functionality. 16

LibTopoART.ITopoART_R

Interface summarising the basic TopoART-R functionality. 19

LibTopoART.LibTopoART_info

Struct `LibTopoART_info` provides some metainformation regarding the respective implementation of LibTopoART. 21

LibTopoART.TopoART

Class `TopoART` provides an implementation of the TopoART neural network as proposed in "Tscherepanow, Marko (2010). TopoART: A topology learning hierarchical ART network. In Proceedings of the International Conference on Artificial Neural Networks, LNCS 6354 (pp. 157–167). Berlin, Germany: Springer." 22

LibTopoART.TopoART_R

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

LibTopoART.TopoART_R_Prediction

Class `F2_output` contains a prediction made by a TopoART-R network. 27

Chapter 4

Namespace Documentation

4.1 Package LibTopoART

Classes

- class Episodic_TopoART

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

- class Fast_TopoART

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

- class Fast_TopoART_Base

  Base class providing functionality common to several TopoART networks.

- struct LibTopoART_info

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

- class F2_output

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

- class TopoART_R_Prediction

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

- interface ITopoART

  Interface summarising the basic TopoART functionality.

- class TopoART

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

- interface ITopoART_R

  Interface summarising the basic TopoART-R functionality.

- class TopoART_R

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

4.2 Package LibTopoART_samples

Classes

- class Episodic_TopoART_sample1

  Sample using artificial two-dimensional data.
- class TopoART_R_sample1

  Function approximation using TopoART-R.
- class TopoART_sample1

  Simple TopoART sample.
- class TopoART_sample2

  Sample using artificial two-dimensional data.

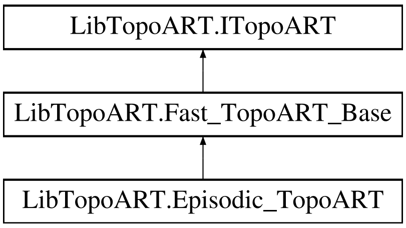
Chapter 5

Class Documentation

5.1 LibTopoART.Episodic_TopoART Class Reference

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

Inheritance diagram for LibTopoART.Episodic_TopoART:



Public Member Functions

- Episodic_TopoART (long input_dimension, long module_number, decimal rho_a_value, long t_max_value)

  This constructor initialises an Episodic TopoART network.
- Episodic_TopoART (string path)

  This constructor loads a saved Episodic TopoART network.
- override void Learn (decimal[] input)

  This method performs a single training step.

Static Public Attributes

- new static readonly decimal file_format_version =0.01m

  Instance variable `file_format_version` represents the version of the file format used by class `Episodic_TopoART`.

Properties

- long T_max `[get]`

  Property `T_max` represents the considered time frame.

5.1.1 Detailed Description

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

5.1.2 Constructor & Destructor Documentation

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

This constructor initialises an Episodic TopoART network.

Parameters

input_dimension	The dimension of input vectors to be learnt.
module_number	The number of Episodic TopoART modules.
rho_a_value	The vigilance parameter of the first Episodic TopoART module (ETA a).
t_max_value	The parameter limiting the considered time frame.

5.1.2.2 LibTopoART.Episodic_TopoART.Episodic_TopoART (string *path*)

This constructor loads a saved Episodic TopoART network.

Parameters

path	The path of a binary Episodic TopoART file.

5.1.3 Member Function Documentation

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

This method performs a single training step.

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

Parameters

input	The input vector to be learnt.

Implements LibTopoART.Fast_TopoART_Base.

5.1.4 Member Data Documentation

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

Instance variable `file_format_version` represents the version of the file format used by class `Episodic_TopoART`.

5.2 LibTopoART_samples.Episodic_TopoART_sample1 Class Reference

Sample using artificial two-dimensional data.

5.2.1 Detailed Description

Sample using artificial two-dimensional data.

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

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

5.3 LibTopoART.F2_output Class Reference

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

Public Member Functions

- F2_output ()

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

Public Attributes

- decimal bm_node_activation

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

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

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

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

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

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

5.3.1 Detailed Description

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

5.3.2 Constructor & Destructor Documentation

5.3.2.1 LibTopoART.F2_output.F2_output ( )

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

5.3.3 Member Data Documentation

5.3.3.1 long LibTopoART.F2_output.bm_cluster_ID

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

5.3.3.2 decimal LibTopoART.F2_output.bm_node_activation

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

5.3.3.3 long LibTopoART.F2_output.bm_node_ID

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

5.3.3.4 long LibTopoART.F2_output.bm_permanent_cluster_ID

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

5.3.3.5 decimal LibTopoART.F2_output.bm_permanent_node_activation

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

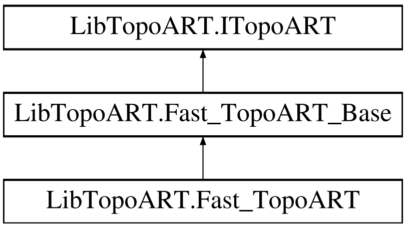
5.3.3.6 long LibTopoART.F2_output.bm_permanent_node_ID

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

5.4 LibTopoART.Fast_TopoART Class Reference

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

Inheritance diagram for LibTopoART.Fast_TopoART:



Public Member Functions

- Fast_TopoART (long input_dimension, long module_number, decimal rho_a_value)

  This constructor initialises a TopoART network.
- Fast_TopoART (string path)

  This constructor loads a saved TopoART network.
- override void Learn (decimal[] input)

  This method performs a single training step.

Additional Inherited Members

5.4.1 Detailed Description

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

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

Class `Fast_TopoART` requires all input to lie in the interval [0,1].

5.4.2 Constructor & Destructor Documentation

5.4.2.1 LibTopoART.Fast_TopoART.Fast_TopoART (long *input_dimension,* long *module_number,* decimal *rho_a_value*)

This constructor initialises a TopoART network.

Parameters

input_dimension	The dimension of input vectors to be learnt.
module_number	The number of TopoART modules.
rho_a_value	The vigilance parameter of the first TopoART module (TA a).

5.4.2.2 LibTopoART.Fast_TopoART.Fast_TopoART (string *path*)

This constructor loads a saved TopoART network.

Parameters

path	The path of a binary TopoART file.

5.4.3 Member Function Documentation

5.4.3.1 override void LibTopoART.Fast_TopoART.Learn (decimal[] *input*) `[virtual]`

This method performs a single training step.

Parameters

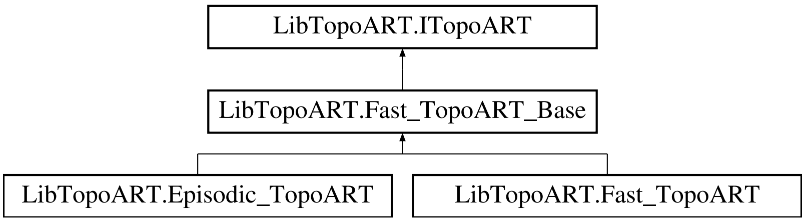
input	The input vector to be learnt.

Implements LibTopoART.Fast_TopoART_Base.

5.5 LibTopoART.Fast_TopoART_Base Class Reference

Base class providing functionality common to several TopoART networks.

Inheritance diagram for LibTopoART.Fast_TopoART_Base:



Public Member Functions

- abstract void Learn (decimal[] input)

  This method performs a single training step.
- void ComputeClusterIDs ()

  This method computes the cluster IDs for all neurons.
- F2_output[] GetBMOutput (decimal[] input)

  This method finds the closest category for a given test input.
- void SaveText (string path)

  This method saves the entire network as a text file.
- void Save (string path)

  This method saves the entire network as a binary file.

Static Public Attributes

- static readonly decimal file_format_version =0.09m

  Instance variable `file_format_version` *represents the version of the file format used by class* `Fast_TopoART`*.*
- static readonly string integer_base_type ="long"

  Instance variable `integer_base_type` *provides a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.*
- static readonly string float_base_type ="long"

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

Properties

- long ModuleNum `[get]`

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

  Property `LearningSteps` *represents the total number of performed learning steps.*
- decimal Rho_a `[get]`

  Property `Rho_a` *represents the vigilance parameter of the first TopoART module (TA a).*
- decimal Beta_sbm `[get, set]`

  Property `Beta_sbm` *represents the learning rate of the second best-matching nodes.*
- long Tau `[get, set]`

  Property `Tau` *represents the parameter tau required for the removal of nodes and edges.*

- long Phi `[get, set]`

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

  Property `Alpha` *represents the choice parameter alpha.*
- long[] NodeNum `[get]`

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

  Property `ClusterNum` *represents the number of TopoART clusters found by each module.*

5.5.1 Detailed Description

Base class providing functionality common to several TopoART networks.

5.5.2 Member Function Documentation

5.5.2.1 void LibTopoART.Fast_TopoART_Base.ComputeClusterIDs ( )

This method computes the cluster IDs for all neurons.

Implements LibTopoART.ITopoART.

5.5.2.2 F2_output [ ] LibTopoART.Fast_TopoART_Base.GetBMOutput (decimal[ ] *input*)

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

Parameters

input	The input vector.

Returns

> An array of type `F2_output`. Each entry contains the ID of the best-matching node and the corresponding cluster ID for one TopoART module.

Implements LibTopoART.ITopoART.

5.5.2.3 abstract void LibTopoART.Fast_TopoART_Base.Learn (decimal[ ] *input*) `[pure virtual]`

This method performs a single training step.

Parameters

input	The input vector to be learnt.

Implements LibTopoART.ITopoART.

Implemented in LibTopoART.Episodic_TopoART, and LibTopoART.Fast_TopoART.

5.5.2.4 void LibTopoART.Fast_TopoART_Base.Save (string *path*)

This method saves the entire network as a binary file.

Parameters

path	A `string` representing the save file.

Implements LibTopoART.ITopoART.

5.5.2.5 void LibTopoART.Fast_TopoART_Base.SaveText (string *path*)

This method saves the entire network as a text file.

Parameters

path	A `string` representing the save file.

Implements LibTopoART.ITopoART.

5.5.3 Member Data Documentation

5.5.3.1 readonly decimal LibTopoART.Fast_TopoART_Base.file_format_version =0.09m `[static]`

Instance variable `file_format_version` represents the version of the file format used by class `Fast_TopoART`.

5.5.3.2 readonly string LibTopoART.Fast_TopoART_Base.float_base_type ="long" `[static]`

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

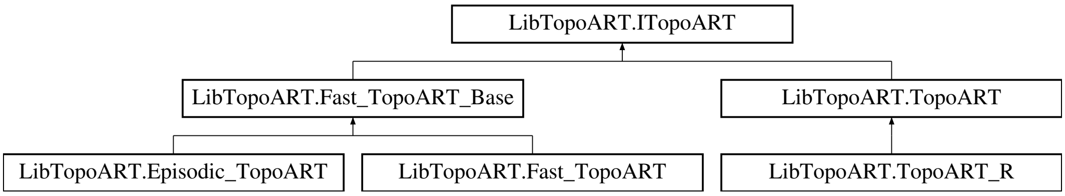
5.5.3.3 readonly string LibTopoART.Fast_TopoART_Base.integer_base_type ="long" `[static]`

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

5.6 LibTopoART.ITopoART Interface Reference

Interface summarising the basic TopoART functionality.

Inheritance diagram for LibTopoART.ITopoART:



Public Member Functions

- void ComputeClusterIDs ()

  This method computes the cluster IDs for all neurons.
- F2_output[] GetBMOutput (decimal[] input)

  This method finds the closest category for a given test input.
- void Learn (decimal[] input)

This method performs a single training step.

- void SaveText (string path)

  This method saves the entire network as a text file.
- void Save (string path)

  This method saves the entire network as a binary file.

Properties

- long[] NodeNum `[get]`

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

  Property `ClusterNum` *represents the number of TopoART clusters found by each module.*
- long ModuleNum `[get]`

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

  Property `LearningSteps` *represents the total number of performed learning steps.*
- decimal Beta_sbm `[get, set]`

  Property `Beta_sbm` *represents the learning rate of the second best-matching nodes.*
- decimal Rho_a `[get]`

  Property `Rho_a` *represents the vigilance parameter of the first TopoART module (TA a).*
- long Tau `[get, set]`

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

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

  Property `Alpha` *represents the choice parameter alpha.*

5.6.1 Detailed Description

Interface summarising the basic TopoART functionality.

5.6.2 Member Function Documentation

5.6.2.1 void LibTopoART.ITopoART.ComputeClusterIDs ( )

This method computes the cluster IDs for all neurons.

Implemented in LibTopoART.TopoART, and LibTopoART.Fast_TopoART_Base.

5.6.2.2 F2_output [ ] LibTopoART.ITopoART.GetBMOutput (decimal[ ] *input*)

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

Parameters

input	The input vector.

Returns

An array of type `F2_output`. Each entry contains the ID of the best-matching node and the corresponding cluster ID for one TopoART module.

Implemented in LibTopoART.TopoART, and LibTopoART.Fast_TopoART_Base.

5.6.2.3 void LibTopoART.ITopoART.Learn (decimal[ ] *input*)

This method performs a single training step.

Parameters

input	The input vector to be learnt.

Implemented in LibTopoART.TopoART, LibTopoART.Episodic_TopoART, LibTopoART.Fast_TopoART, and LibTopoART.Fast_TopoART_Base.

5.6.2.4 void LibTopoART.ITopoART.Save (string *path*)

This method saves the entire network as a binary file.

Parameters

path	A `string` representing the save file.

Implemented in LibTopoART.TopoART, and LibTopoART.Fast_TopoART_Base.

5.6.2.5 void LibTopoART.ITopoART.SaveText (string *path*)

This method saves the entire network as a text file.

Parameters

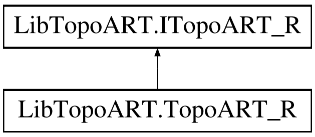
path	A `string` representing the save file.

Implemented in LibTopoART.TopoART, and LibTopoART.Fast_TopoART_Base.

5.7 LibTopoART.ITopoART_R Interface Reference

Interface summarising the basic TopoART-R functionality.

Inheritance diagram for LibTopoART.ITopoART_R:



Public Member Functions

- void Learn (decimal[] i_vec, decimal[] d_vec)

  This method performs a single training step.
- TopoART_R_Prediction Predict (decimal[] i_vec, bool[] m_i_vec)

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

5.7.1 Detailed Description

Interface summarising the basic TopoART-R functionality.

5.7.2 Member Function Documentation

5.7.2.1 void LibTopoART.ITopoART_R.Learn (decimal[ ] *i_vec*, decimal[ ] *d_vec*)

This method performs a single training step.

Parameters

i_vec	The input vector (independent variables) to be learnt.
d_vec	The output vector (dependent variables) corresponding to `i_vec`.

Implemented in LibTopoART.TopoART_R.

5.7.2.2 TopoART_R_Prediction LibTopoART.ITopoART_R.Predict (decimal[ ] *i_vec*, bool[ ] *m_i_vec*)

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

Parameters

i_vec	The input vector (independent variables).
m_i_vec	The mask vector corresponding to `i_vec`.

Returns

An object of type `TopoART_R_Prediction` containing the predicted values for the unknown independent variables and all dependent variables.

Implemented in LibTopoART.TopoART_R.

5.8 LibTopoART.LibTopoART_info Struct Reference

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

Static Public Attributes

- static readonly decimal version =0.50m

  Instance variable `version` represents the version of LibTopoART.
- static readonly string[] networks

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

5.8.1 Detailed Description

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

5.8.2 Member Data Documentation

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

Initial value:

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

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

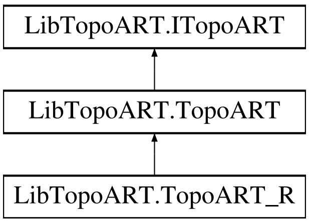
5.8.2.2 readonly decimal LibTopoART.LibTopoART_info.version =0.50m `[static]`

Instance variable `version` represents the version of LibTopoART.

5.9 LibTopoART.TopoART Class Reference

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

Inheritance diagram for LibTopoART.TopoART:



Public Member Functions

- TopoART (long input_dimension, long module_number, decimal rho_a_value)

  This constructor initialises a TopoART network.
- TopoART (string path)

  This constructor loads a saved TopoART network.
- void ComputeClusterIDs ()

  This method computes the cluster IDs for all neurons.
- F2_output[] GetBMOutput (decimal[] input)

  This method finds the closest category for a given test input.
- void Learn (decimal[] input)

  This method performs a single training step.
- void SaveText (string path)

  This method saves the entire network as a text file.
- void Save (string path)

  This method saves the entire network as a binary file.

Static Public Attributes

- static readonly decimal file_format_version =0.09m

  Instance variable `file_format_version` represents the version of the file format used by class `TopoART`.
- static readonly string integer_base_type ="long"

  Instance variable `integer_base_type` provides a string containing the data type used for representing integer variables (IDs, parameters, counters, etc.) internally.
- static readonly string float_base_type ="decimal"

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

Protected Attributes

- readonly long x_F0_dim

  Instance variable `x_F0_dim` *represents the input dimension.*
- readonly decimal[] x_F0

  Instance variable `x_F0` *represents the current input vector.*

Properties

- long[] NodeNum `[get]`

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

  Property `ClusterNum` *represents the number of TopoART clusters found by each module.*
- long ModuleNum `[get]`

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

  Property `LearningSteps` *represents the total number of performed learning steps.*
- decimal Rho_a `[get]`

  Property `Rho_a` *represents the vigilance parameter of the first TopoART module (TA a).*
- decimal Beta_sbm `[get, set]`

  Property `Beta_sbm` *represents the learning rate of the second best-matching nodes.*
- long Tau `[get, set]`

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

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

  Property `Alpha` *represents the choice parameter alpha.*

5.9.1 Detailed Description

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

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

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

5.9.2 Constructor & Destructor Documentation

5.9.2.1 LibTopoART.TopoART.TopoART (long *input_dimension*, long *module_number*, decimal *rho_a_value*)

This constructor initialises a TopoART network.

Parameters

input_dimension	The dimension of input vectors to be learnt.
module_number	The number of TopoART modules.

rho_a_value	The vigilance parameter of the first TopoART module (TA a).

5.9.2.2 LibTopoART.TopoART.TopoART (string *path*)

This constructor loads a saved TopoART network.

Parameters

path	The path of a binary TopoART file.

5.9.3 Member Function Documentation

5.9.3.1 void LibTopoART.TopoART.ComputeClusterIDs ( )

This method computes the cluster IDs for all neurons.

Implements LibTopoART.ITopoART.

5.9.3.2 F2_output [ ] LibTopoART.TopoART.GetBMOutput (decimal[ ] *input*)

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

Parameters

input	The input vector.

Returns

> An array of type `F2_output`. Each entry contains the ID of the best-matching node and the corresponding cluster ID for one TopoART module.

Implements LibTopoART.ITopoART.

5.9.3.3 void LibTopoART.TopoART.Learn (decimal[ ] *input*)

This method performs a single training step.

Parameters

input	The input vector to be learnt.

Implements LibTopoART.ITopoART.

5.9.3.4 void LibTopoART.TopoART.Save (string *path*)

This method saves the entire network as a binary file.

Parameters

path	A `string` representing the save file.

Implements LibTopoART.ITopoART.

5.9.3.5 void LibTopoART.TopoART.SaveText (string *path*)

This method saves the entire network as a text file.

Parameters

path	A `string` representing the save file.

Implements LibTopoART.ITopoART.

5.9.4 Member Data Documentation

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

Instance variable `file_format_version` represents the version of the file format used by class `TopoART`.

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

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

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

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

5.9.4.4 readonly decimal [] LibTopoART.TopoART.x_F0 `[protected]`

Instance variable `x_F0` represents the current input vector.

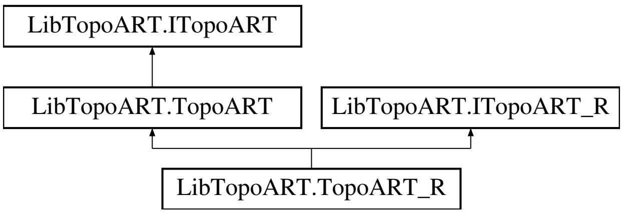
5.9.4.5 readonly long LibTopoART.TopoART.x_F0_dim `[protected]`

Instance variable `x_F0_dim` represents the input dimension.

5.10 LibTopoART.TopoART_R Class Reference

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

Inheritance diagram for LibTopoART.TopoART_R:



Public Member Functions

- TopoART_R (long i_dimension, long d_dimension, long module_number, decimal rho_a_value)

    This constructor initialises a TopoART-R network.

- TopoART_R (string path)

  This constructor loads a saved TopoART-R network.
- void Learn (decimal[] i_vec, decimal[] d_vec)

  This method performs a single training step.
- TopoART_R_Prediction Predict (decimal[] i_vec, bool[] m_i_vec)

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

Static Public Attributes

- new static readonly decimal file_format_version =0.01m

  Instance variable `file_format_version` *represents the version of the file format used by class* `TopoART_R`.

Additional Inherited Members

5.10.1 Detailed Description

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

Class `TopoART_R` requires all input and output to lie in the interval [0,1].

5.10.2 Constructor & Destructor Documentation

5.10.2.1 LibTopoART.TopoART_R.TopoART_R (long *i_dimension,* long *d_dimension,* long *module_number,* decimal *rho_a_value*)

This constructor initialises a TopoART-R network.

Parameters

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

5.10.2.2 LibTopoART.TopoART_R.TopoART_R (string *path*)

This constructor loads a saved TopoART-R network.

Parameters

path	The path of a binary TopoART-R file.

5.10.3 Member Function Documentation

5.10.3.1 void LibTopoART.TopoART_R.Learn (decimal[] *i_vec,* decimal[] *d_vec*)

This method performs a single training step.

Parameters

i_vec	The input vector (independent variables) to be learnt.
d_vec	The output vector (dependent variables) corresponding to `i_vec`.

Implements LibTopoART.ITopoART_R.

5.10.3.2 TopoART_R_Prediction LibTopoART.TopoART_R.Predict (decimal[ ] *i_vec,* bool[ ] *m_i_vec*)

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

Parameters

i_vec	The input vector (independent variables).
m_i_vec	The mask vector corresponding to `i_vec`.

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.

5.10.4 Member Data Documentation

5.10.4.1 new static readonly decimal LibTopoART.TopoART_R.file_format_version =0.01m `[static]`

Instance variable `file_format_version` represents the version of the file format used by class `TopoART_R`.

5.11 LibTopoART.TopoART_R_Prediction Class Reference

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

Public Member Functions

- TopoART_R_Prediction (decimal[ ] i_vec_prediction, decimal[ ] d_vec_prediction)

  This constructor sets the instance variables `i_vec_prediction` and `d_vec_prediction` of class `TopoART_R_Prediction`.
- void PrintPredictions ()

  This method prints the predictions on the console.

Public Attributes

- readonly decimal[ ] i_vec_prediction

  Instance variable `i_vec_prediction` represents predictions for unknown independent variables.
- readonly decimal[ ] d_vec_prediction

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

Static Public Attributes

- static readonly decimal NO_PREDICTION =-1.0m

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

5.11.1 Detailed Description

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

5.11.2 Constructor & Destructor Documentation

5.11.2.1 LibTopoART.TopoART_R_Prediction.TopoART_R_Prediction (decimal[ ] *i_vec_prediction,* decimal[ ] *d_vec_prediction*)

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

5.11.3 Member Function Documentation

5.11.3.1 void LibTopoART.TopoART_R_Prediction.PrintPredictions ( )

This method prints the predictions on the console.

5.11.4 Member Data Documentation

5.11.4.1 readonly decimal [ ] LibTopoART.TopoART_R_Prediction.d_vec_prediction

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

5.11.4.2 readonly decimal [ ] LibTopoART.TopoART_R_Prediction.i_vec_prediction

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

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

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

5.12 LibTopoART_samples.TopoART_R_sample1 Class Reference

Function approximation using TopoART-R.

5.12.1 Detailed Description

Function approximation using TopoART-R.

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

5.13 LibTopoART_samples.TopoART_sample1 Class Reference

Simple TopoART sample.

5.13.1 Detailed Description

Simple TopoART sample.

First, a datased 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.

5.14 LibTopoART_samples.TopoART_sample2 Class Reference

Sample using artificial two-dimensional data.

5.14.1 Detailed Description

Sample using artificial two-dimensional data.

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

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

Index