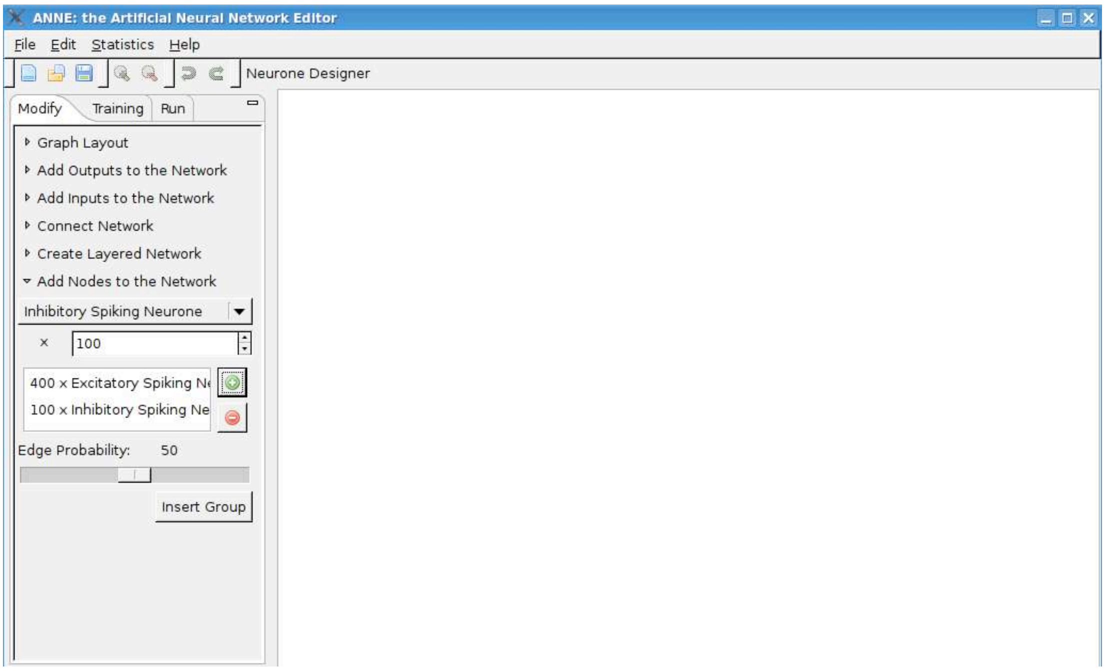
ANNE User Guide

Table of Contents



Inhibitory Spiking Neurone

× 100

400 x Excitatory Spiking N

100 x Inhibitory Spiking Ne

Edge Probability: 50

Figure 1: The Add Nodes Panel

1 User Guide

1.1 Creating a Group of Neurones

When ANNE is first opened, it contains an empty neural network. Therefore, the first thing to do is add some neurones. To do this, go to the Modify Panel in the sidebar and then select the “Add Nodes to the Network” option. Choose the type of neural network you would like to add. Next, choose the number of neurones you would like in the neural network. To add this group to the graph, select the “add group” button (the green plus sign) and then click “Insert Group” to add them to the graph. To see the resulting neurones, you can zoom into this neural network by clicking the neural network and then pressing the “Zoom In” button in the toolbar.

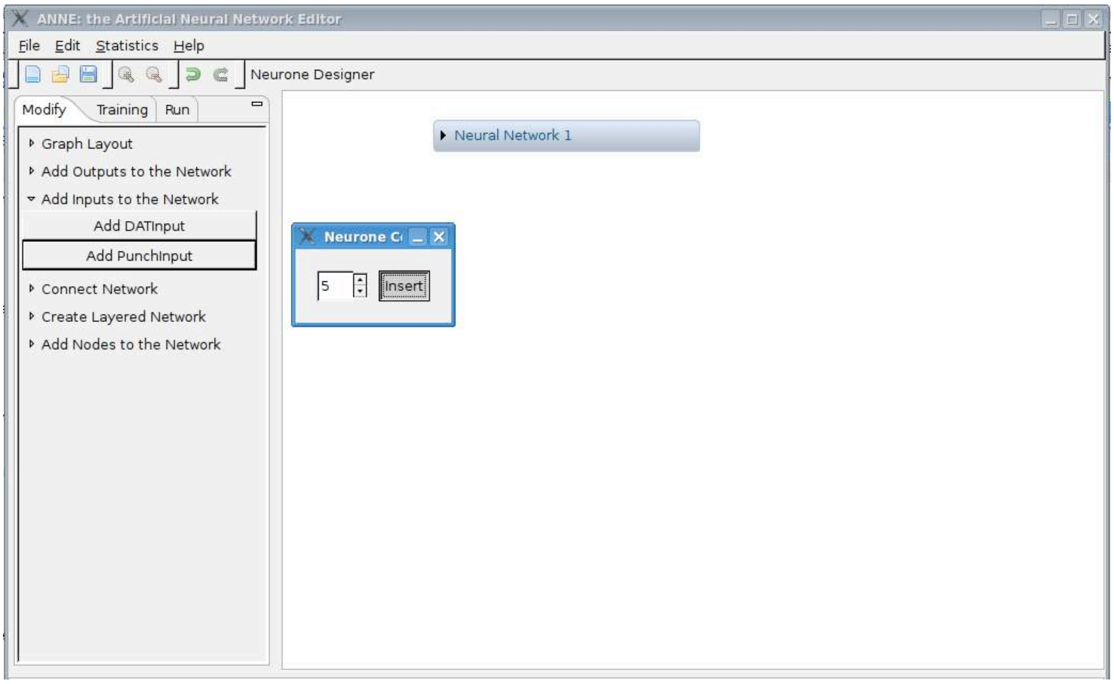
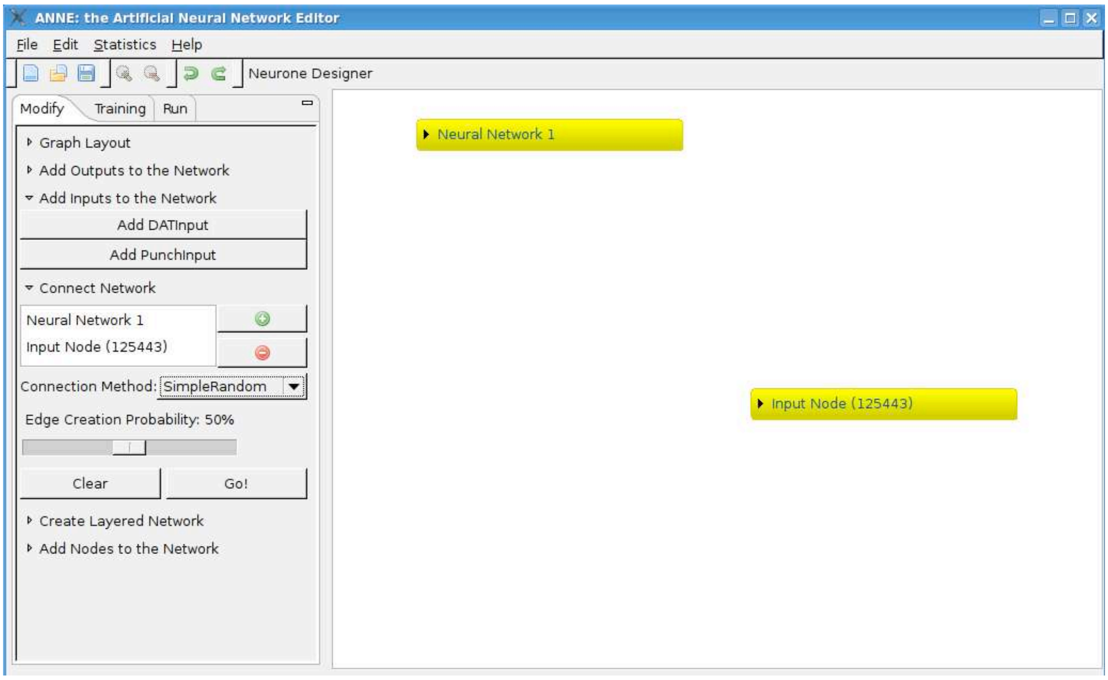


Figure 2: The Add Inputs to Network Panel

1.2 Creating Input Neurones

To create input nodes, select the “Add Inputs to Network” option in the Modify Tab. Next, select the type of input node you would like to use. If “PunchInput” is selected, a dialog box will appear prompting you to select the number of input nodes you would like to insert. If “DATInput” is selected, you will be prompted to with an open dialog box to select a *.dat* file to use.



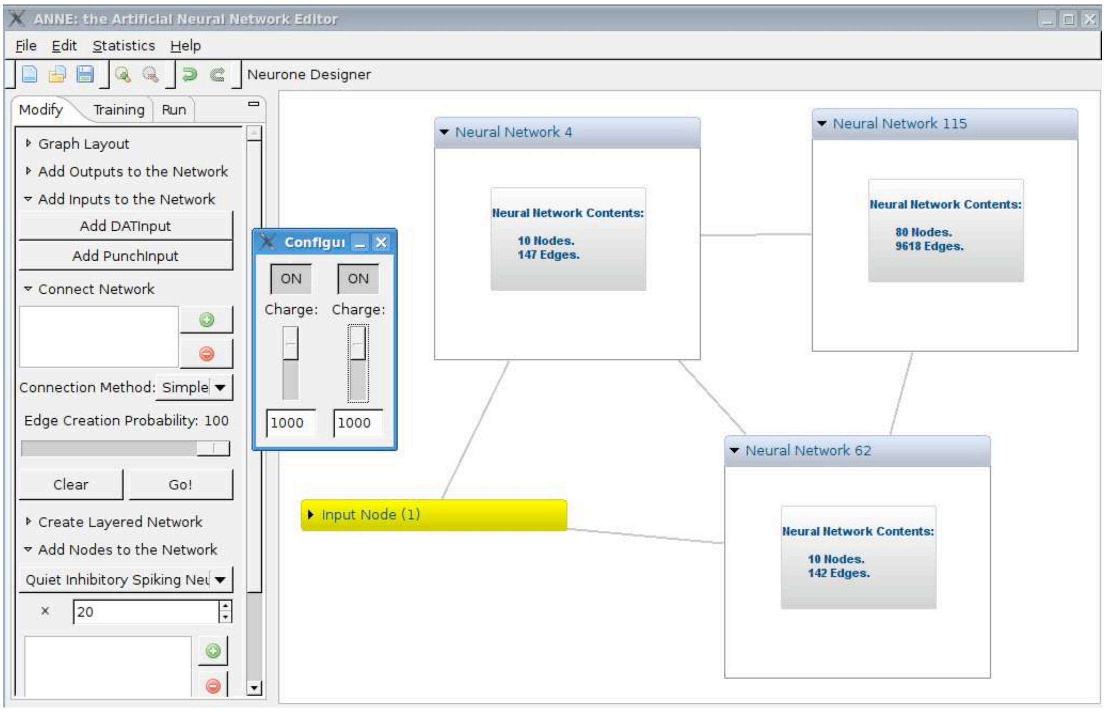
Connection Method: SimpleRandom

Edge Creation Probability: 50%

Figure 3: Connecting Networks

1.3 Connecting Networks

To connect a group of neurones or neural networks, you will need to use the “Connect Network” panel in the Modify tab. To select which nodes you wish to connect, click the add button and then select the nodes you would like to connect together by dragging a box around them with the mouse, or by clicking on them directly. You can select the probablity of the edges being generated by using the slider, and the algorithm to use from the drop-down menu. Once you are happy with your selection and options, click the “Go” button to generate the edges that connect the nodes together.



Connection Method: Simple

Edge Creation Probability: 100

Quiet Inhibitory Spiking Neu

× 20

1000 1000

Neural Network Contents:
10 Nodes.
147 Edges.

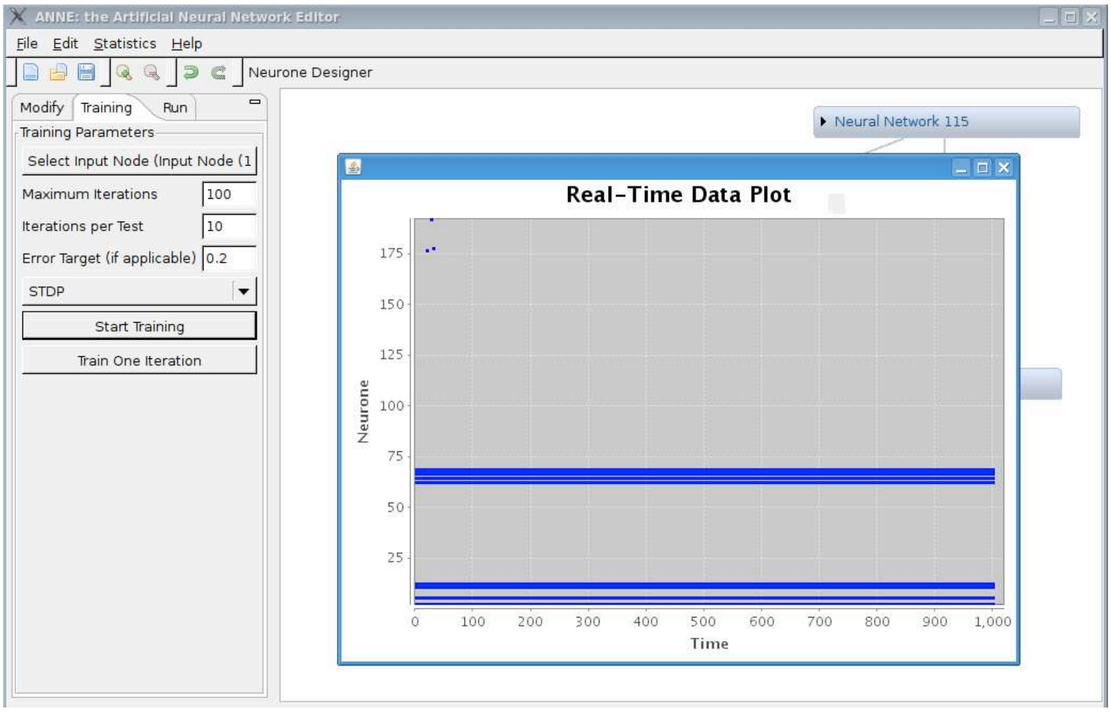
Neural Network Contents:
80 Nodes.
9618 Edges.

Neural Network Contents:
10 Nodes.
142 Edges.

Figure 4: Example Network

1.4 Training and Simulating a Network

Once you have finished editing your network, it can be trained and simulated. The example network we are using consists of two PunchingInput neurones, two groups of quiet excitatory spiking neurones, and a group containing a mixture of quiet excitatory and quiet inhibitory spiking neurones.



Maximum Iterations 100
Iterations per Test 10
Error Target (if applicable) 0.2
STDP

Figure 5: Training a Network

1.5 Training

Now that the network has been created, a real-time plot of its activity can be made by selecting "Statistics" then enabling "RealTimePlot" from the menu. When you train and run your network, its activity will be shown in the plot window that is created. This can be navigated using your arrow keys; ↑ and ↓ zoom in and out respectively, and ← and → will pan the plot left and right.

To train the network, first open the Train tab in the sidebar. Click the "Select Input Node" button and click an input node, then change any training parameters as required. The drop-down box in this tab contains the different training algorithms available; in our example, we will use STDP. Click the Start Training button, then wait for the network to finish training. The Real-Time plot window can show its progress.

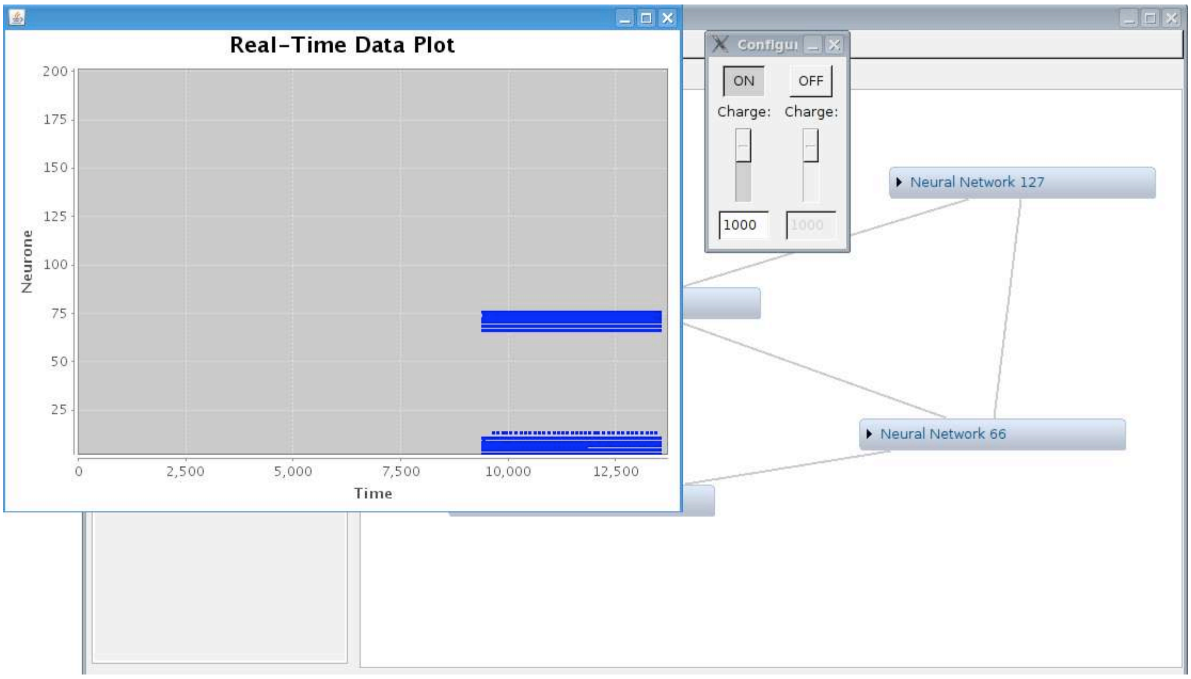
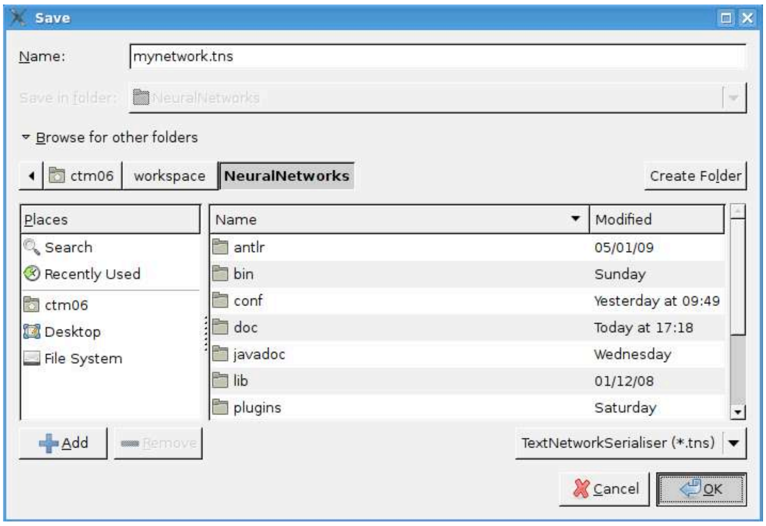


Figure 6: Running a Network

1.6 Running Networks

To run your trained network, click on the Run tab in the sidebar. There are options for simulating the network one step at a time, but for now just click the Run button to make the network start running. Now that the simulation has begun, it is possible make changes to the input nodes and observe the results. In our example, clicking the button by one of the input neurones stops it from firing, and the real-time plot shows that the neural network has been sufficiently trained to keep responding as if it were still turned on; it has learned the spatio-temporal association between these neurones.



Name: mynetwork.tns

Save in folder: NeuralNetworks

Name	Modified
antlr	05/01/09
bin	Sunday
conf	Yesterday at 09:49
doc	Today at 17:18
javadoc	Wednesday
lib	01/12/08
plugins	Saturday

TextNetworkSerialiser (*.tns)

Figure 7: Saving and Loading Networks

1.7 Saving and Loading

Finally, save your changes to the neural network and close the program. Either click “File” then “Save as” from the menu, or click the save icon on the toolbar. Once the network has finished saving, close ANNE by either clicking the “X” icon in the top-right corner of the screen or choosing “File” then “Close” from the menu.

Contents

Chapter 1

Package uk.ac.ic.doc.neuralnets.gui

Package Contents *Page*

Classes

1.1 Classes

1.1.1 Class **GUILayout**

This class lays out the GUI skeleton in a given a shell giving access to the main pane, side pane and bottom pane.

Declaration

public class GUILayout
extends java.lang.Object

Constructors

- *GUILayout*
  `public` **GUILayout**`( org.eclipse.swt.widgets.Shell` **shell** `)`
  - **Usage**
    - Adds layout containers to the shell.
  - **Parameters**
    - `shell` -

Methods

- *getBottomContainer*
  `public Composite` **getBottomContainer**`( )`
  - **Usage**
    - Get the bottom pane
  - **Returns** - the Composite for the bottom container

- *getGraphContainer*
  `public Composite` **getGraphContainer**`( )`
  - **Usage**
    - Gets the main window pane
  - **Returns** - the Composite for the graph container

- *getSidebarContainer*
  `public Composite` **getSidebarContainer**`( )`
  - **Usage**
    - Gets the side pane
  - **Returns** - the Composite for the side container

- *getToolbar*
  `public CoolBar` **getToolbar**`( )`

- **Usage**
  - Get the toolbar
- **Returns** - the application toolbar as a CoolBar

1.1.2 Class GUILog

Creates the log box in the bottom bar

Declaration

public class GUILog
extends java.lang.Object

Constructors

- *GUILog*
  `public` **GUILog**`( org.eclipse.swt.widgets.Composite` **container** `)`

1.1.3 Class GUIMain

Bootstrap.

Declaration

public class GUIMain
extends java.lang.Object

Constructors

- *GUIMain*
  `public` **GUIMain**`( )`

Methods

- *main*
  `public static void` **main**`( java.lang.String [] ` **args** `)`
  - **Parameters**
    - `args` -

1.1.4 Class **GUIManager**

Manages the GUI representation of a layered neural network. Controls importing and exporting networks to and from their standard model representation, zooming into and out of layers of the network, and tooltips. Listens synchronously for GraphUpdateEvents, NewNeuroneTypeEvents, NeuralNetworkTickEvents and NeuralNetworkSimulationEvents

Declaration

public class GUIManager
extends uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager

Constructors

- *GUIManager*
  `public` **GUIManager**`( org.eclipse.zest.core.widgets.IContainer` **graph**, `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **network** `)`
  - **Usage**
    - Creates a GUIManager to display a given Neural Network on a given SWT IContainer canvas.
  - **Parameters**
    - `graph` - the canvas on which to display the network
    - `network` - the network to be displayed in the GUI
- *GUIManager*
  `public` **GUIManager**`( org.eclipse.zest.core.widgets.IContainer` **graph**, `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **network**, `uk.ac.ic.doc.neuralnets.persistence.FileSpecification` **location** `)`
  - **Usage**
    - Creates a GUIManager to display a given Neural Network, from a given location, on a given SWT IContainer canvas.
  - **Parameters**
    - `graph` - the canvas on which to display the network
    - `network` - the network to be displayed in the GUI
    - `location` - the location of the network

Methods

- *addConnection*
  `public void` **addConnection**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
- *canZoomIn*
  `public boolean` **canZoomIn**`( )`
- *canZoomOut*
  `public boolean` **canZoomOut**`( )`

- *disableGraph*
  `public void` **disableGraph**`( )`
  - **Usage**
    - Disable clicks to the graph area.

- *enableGraph*
  `public void` **enableGraph**`( )`
  - **Usage**
    - Enable clicks to the graph area

- *getCurrentNetwork*
  `public NeuralNetwork` **getCurrentNetwork**`( )`

- *getGraph*
  `public Graph` **getGraph**`( )`

- *getNode*
  `public GraphItem` **getNode**`( uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **n** `)`

- *getZoomIDs*
  `public Stack` **getZoomIDs**`( )`

- *getZoomLevels*
  `public Stack` **getZoomLevels**`( )`

- *persistLocations*
  `public void` **persistLocations**`( )`

- *redrawCurrentView*
  `public void` **redrawCurrentView**`( )`

- *remove*
  `public void` **remove**`( org.eclipse.zest.core.widgets.GraphItem` **i** `)`

- *removeNetwork*
  `public void` **removeNetwork**`(`
  `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    - Removes the given neural network from the current view, and redraws the screen as necessary.
  - **Parameters**
    - `n` - the neural network to add to the current section of the neural network

- *reset*
  `protected void` **reset**`( )`

- *updateInterfaceHints*
  `public void` **updateInterfaceHints**`( )`

- *zoomIn*
  `public void` **zoomIn**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`

- *zoomOut*
  `public void` **zoomOut**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager`

(in 19.1.2, page 192)

- *canZoomIn*
  `public abstract boolean` **`canZoomIn`**`( )`
  - **Usage**
    - Checks whether or not it is possible to zoom in. It is only possible to zoom in if exactly one internal network layer is selected.
  - **Returns** - whether or not it is possible to zoom in

- *canZoomOut*
  `public abstract boolean` **`canZoomOut`**`( )`
  - **Usage**
    - Checks whether or not it is possible to zoom out. It is always possible to zoom out unless the current view is the root network.
  - **Returns** - whether or not it is possible to zoom out

- *getZoomIDs*
  `public abstract Stack` **`getZoomIDs`**`( )`
  - **Usage**
    - Returns a stack containing the IDs of each network layer that has currently been zoomed into. This can be used to trace the current zoom path from the root of the neural network.
  - **Returns** - a stack of IDs of each network layer that is currently zoomed into

- *getZoomLevels*
  `public abstract Stack` **`getZoomLevels`**`( )`
  - **Usage**
    - Returns a stack containing each network layer that has currently been zoomed into, starting with the root network.
  - **Returns** - a stack containing each network layer that has currently been zoomed into.

- *zoomIn*
  `public abstract void` **`zoomIn`**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **`n`** `)`
  - **Usage**
    - Zooms into the selected network layer. Clears the current view, and instead shows the contents of the selected network layer.
  - **Parameters**
    - `n` - the network to zoom into.

- *zoomOut*
  `public abstract void` **`zoomOut`**`( )`
  - **Usage**
    - Zooms out one layer. Clears the current view, and instead shows the contents of the current layer's parent. If the current view is the root network, then nothing happens as it is not possible to zoom out further.

Methods inherited from class `uk.ac.ic.doc.neuralnets.coreui.InterfaceManager`

(in 19.1.1, page 189)

- *addConnection*
  `public void` **addConnection**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - Adds the given edge to the current view, and redraws the screen as necessary.
  - **Parameters**
    - `e` -

- *addNetwork*
  `public void` **addNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    - Adds the given neural network to the current view, and redraws the screen as necessary.
  - **Parameters**
    - `n` - the neural network to add to the current section of the neural network

- *addNeurone*
  `public void` **addNeurone**`( uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **n** `)`
  - **Usage**
    - Adds the given neurone to the current view, and redraws the screen as necessary.
  - **Parameters**
    - `n` - the neurone to add to the current section of the neural network

- *addNode*
  `public void` **addNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**
    - Adds the given node to the current view, and redraws the screen as necessary.
  - **Parameters**
    - `n` - the node to add to the current section of the neural network

- *addNode*
  `public void` **addNode**`( uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification` **spec** `)`
  - **Usage**
    - Creates a node from the give specification, adds to the current view, and redraws the screen as necessary.
  - **Parameters**
    - `spec` - the specification of the node to add to the current section of the neural network

- *getCommandControl*
  `public CommandControl` **getCommandControl**`( )`
  - **Usage**
    - Gets the command control used by the GUIManager. This object handles the undo and redo stacks as commands are executed and undone.
  - **Returns** - the CommandControl object used by the GUIManager

- *getCurrentNetwork*
  `public abstract NeuralNetwork` **getCurrentNetwork**`( )`
  - **Usage**
    - Returns the neural network layer currently being viewed in the GUIManager.

  - **Returns** - the current neural network layer

- *getGraph*
  `public abstract Object` **getGraph**`( )`
  - **Usage**
    - Returns the Graph representation used by this UI Manager.
  - **Returns** - the Graph that the Manager draws onto

- *getNode*
  `public abstract Object` **getNode**`( uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **n** `)`
  - **Usage**
    - Finds the GUINode in the GUI corresponding to the given Neurone and returns it. Returns null if the given Neurone is not loaded in the GUI.
  - **Parameters**
    - `n` - the Neurone to look up in the GUI
  - **Returns** - the GUINode in the GUI corresponding to the given Neurone

---

- *getRootNetwork*
  `public NeuralNetwork` **getRootNetwork**`( )`
  - **Usage**
    - Gets the root of the layered neural network stored in the GUIManager.
  - **Returns** - the root of the main neural network

- *getSaveLocation*
  `public FileSpecification` **getSaveLocation**`( )`
  - **Usage**
    - Gets the location to save the network to, or null if no such location exists.
  - **Returns** - the network's save location, or null if none exists

---

- *getUtils*
  `public InteractionUtils` **getUtils**`( )`
  - **Usage**
    - Returns the GUIManager's interaction utilities.
  - **Returns** - the InteractionUtils object used by the GUIManager

---

- *persistLocations*
  `public abstract void` **persistLocations**`( )`
  - **Usage**
    - Pushes down the locations of all Nodes to the model. Allows positions to be persisted to storage and reloaded.

- *redrawCurrentView*
  `public abstract void` **redrawCurrentView**`( )`
  - **Usage**
    - Draws the current view of the graph. Imports the current network layer from the internal model and applies the current layout.

- *remove*
  `public abstract void` **remove**`( java.lang.Object` **i** `)`
  - **Usage**
    - Removes the given GraphItem from the view.
  - **Parameters**

    - `i` - the graphitem to be removed from the view

- *removeNetwork*
  `public void` **removeNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    - Removes the given neural network from the current view, and redraws the screen as necessary.
  - **Parameters**
    - `n` - the neural network to remove from the current section of the neural network

- *reset*
  `protected abstract void` **reset**`( )`
  - **Usage**
    - Reset the current manager, e.g. when a new network is loaded

- *setNetwork*
  `public void` **setNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **network**`, uk.ac.ic.doc.neuralnets.persistence.FileSpecification` **location** `)`
  - **Usage**
    - Loads the given neural network into the GUIManager, from the given location.
  - **Parameters**
    - `network` - the network to be loaded into the GUIManager
    - `location` - the location to load the network from

- *setSaveLocation*
  `public void` **setSaveLocation**`( uk.ac.ic.doc.neuralnets.persistence.FileSpecification` **saveLoc** `)`
  - **Usage**
    - Sets the network's save location.
  - **Parameters**
    - `saveLoc` -

- *updateInterfaceHints*
  `public abstract void` **updateInterfaceHints**`( )`
  - **Usage**
    - Updates the tooltips or other UI hints of all graph elements in the current view.

1.1.5 Class **GUIMenu**

Constructs the application menu. Looks for `MenuPlugins`, sorts them according to priority, then loads them into the menu.

Declaration

public class GUIMenu
extends java.lang.Object

Constructors

- *GUIMenu*
  `public` **GUIMenu**`(` `org.eclipse.swt.widgets.Shell` **rootShell**`,` `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm** `)`
  - **Usage**
    - Creates the application menu by requesting `MenuPlugins` from the PluginManager.
  - **Parameters**
    - `rootShell` - - the shell the menu is for
    - `gm` - - the graph manager.
  - **See Also**
    - `uk.ac.ic.doc.neuralnets.util.plugins.PluginManager` (in 7.2.3, page 60)

Methods

- *addMenuItem*
  `public MenuItem` **addMenuItem**`(` `java.lang.String` **parent**`,` `java.lang.String` **name** `)`
  - **Usage**
    - Adds a named menu item to a parent menu
  - **Parameters**
    - `parent` - - the menu to add the item to. If the parent menu isn't found then the root menu is used.
    - `name` - - the name for the new menu item.
  - **Returns** - the newly created `MenuItem`

- *addMenuSeparator*
  `public void` **addMenuSeparator**`(` `java.lang.String` **parent** `)`
  - **Usage**
    - Add a separator to parent menu
  - **Parameters**
    - `parent` - - menu to separate

- *addSubMenu*
  `public MenuItem` **addSubMenu**`(` `java.lang.String` **parent**`,` `java.lang.String` **name** `)`
  - **Usage**
    - Adds a menu item to the parent menu and connects an empty menu to it. The highest level menu is **"root"** which is automatically created.
  - **Parameters**
    - `parent` - - name of the parent menu, e.g. "root", if the parent menu is not found then the root menu will be used.
    - `name` - - name of the new submenu
  - **Returns** - `MenuItem` for the new submenu, if the submenu already exists then that `MenuItem` is returned.

- *getManager*
  `public ZoomingInterfaceManager` **`getManager`**`( )`
  - **Usage**
    - Get the graph manager.
  - **Returns** - the ZoomingInterfaceManager for the graph.

- *getShell*
  `public Shell` **`getShell`**`( )`
  - **Usage**
    - Get the parent shell of the menu.
  - **Returns** - the main program shell

1.1.6 Class **GUISideBar**

Controls the Sidebar of the UI.

Declaration

public class GUISideBar
extends java.lang.Object

Constructors

- *GUISideBar*
  `public` **`GUISideBar`**`( org.eclipse.swt.widgets.Composite` **`container`**`, uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **`gm`** `)`
  - **Usage**
    - Create the Sidebar.
  - **Parameters**
    - `container` - - sidebar container
    - `gm` - - graph manager.

1.1.7 Class **GUIToolbar**

Constructs the application toolbar from `ToolbarPlugins`. The toolbar is a collection of groups which can each contain a number of buttons/controls.

Declaration

public class GUIToolbar
extends java.lang.Object

Constructors

- *GUIToolbar*
  `public` **GUIToolbar**`( org.eclipse.swt.widgets.CoolBar` **coolbar**`, uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm** `)`
  - **Usage**
    - ∗ Creates the application toolbar by requesting `ToolbarPlugins` from the plugin manager.
  - **Parameters**
    - ∗ `coolbar` -
    - ∗ `gm` -

Methods

- *addButton*
  `public ToolItem` **addButton**`( java.lang.String` **parent**`, org.eclipse.swt.graphics.Image` **icon** `)`
  - **Usage**
    - ∗ Add a button to a parent group with an icon.
  - **Parameters**
    - ∗ `parent` - - the parent group.
    - ∗ `icon` - - the icon `Image`.
  - **Returns** - - the new button

- *addButton*
  `public ToolItem` **addButton**`( java.lang.String` **parent**`, java.lang.String` **name** `)`
  - **Usage**
    - ∗ Add a button to a parent group with text
  - **Parameters**
    - ∗ `parent` - - the name parent group
    - ∗ `name` - - text to appear on the button
  - **Returns** - - the new button

- *addButton*
  `public ToolItem` **addButton**`( java.lang.String` **parent**`, java.lang.String` **name**`, int` **type** `)`
  - **Usage**
    - ∗ Add a radio/toggle button to a parent group.
  - **Parameters**
    - ∗ `parent` - - the parent group
    - ∗ `name` - - the button name
    - ∗ `type` - - the button type SWT.CHECK/SWT.RADIO/SWT.SEPARATOR
  - **Returns** - - the new button

- *addGroup*
  `public CoolItem` **addGroup**`( java.lang.String` **name** `)`
  - **Usage**
    - Add a new group to the toolbar.
  - **Parameters**
    - `name` - - name of the new toolbar.

- *getManager*
  `public ZoomingInterfaceManager` **getManager**`( )`
  - **Usage**
    - Get the graph manager. Allows toolbar buttons to have listeners which modify the graph.
  - **Returns** - - the manager for the graph.

- *getShell*
  `public Shell` **getShell**`( )`
  - **Usage**
    - Get the parent shell. Allows toolbar buttons to have listeners which create new shells.
  - **Returns** - - the toolbars parent shell

- *repackGroup*
  `public void` **repackGroup**`( java.lang.String` **itemGroup** `)`
  - **Usage**
    - Recalculate the size of the toolbar group
  - **Parameters**
    - `itemGroup` -

1.1.8 Class **ImageHandler**

The ImageHandleris responsible for retrieving `Image` instances for named image files.

Declaration

public class ImageHandler
extends java.lang.Object

Methods

- *get*
  `public static ImageHandler` **get**`( )`
  - **Usage**
    - Get the ImageHandler.

  - **Returns** - the ImageHandler

- *getIcon*
  `public Image` **getIcon**`( java.lang.String` **name** `)`
  - **Usage**
    - Create an SWT Image for the named icon file from the *res/icons* folder
  - **Parameters**
    - `name` - - Icon file name with or without .png extension
  - **Returns** - Image object for file or null if the file is not found.

1.1.9 Class MenuPlugin

Menu plugins create the application menu structure. See GUIMenu for the interface used to create menus.

Declaration

public abstract class MenuPlugin
extends uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin

Constructors

- *MenuPlugin*
  `public` **MenuPlugin**`( )`

Methods

- *load*
  `public abstract void` **load**`( uk.ac.ic.doc.neuralnets.gui.GUIMenu` **menu** `)`
  - **Usage**
    - Creates the menu for the plugin.
  - **Parameters**
    - `menu` -

Methods inherited from class `uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin`

(in 7.2.4, page 62)

- *compareTo*
  `public int` **compareTo**`( uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin` **o** `)`
- *getPriority*
  `public abstract int` **getPriority**`( )`
  - **Usage**
    - The plugin's priority.
  - **Returns** - the priority

1.1.10 Class **NetworkModifier**

Network Modifiers are pluggable units in the Modify tab.

Declaration

public abstract class NetworkModifier
extends uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin

Constructors

- *NetworkModifier*
  `public` **NetworkModifier**`( )`

Methods

- *getConfigurationGUI*
  `public abstract Composite` **getConfigurationGUI**`(`
  `org.eclipse.swt.widgets.Composite` **parent**,
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm**,
  `org.eclipse.swt.widgets.ExpandItem` **ei** `)`
  - **Usage**
    - Create the UI for the unit, called during the initialization of the modify tab.
  - **Parameters**
    - `parent` - - the expand bar for modifiers
    - `gm` - - the graph manager
    - `ei` - - the expand item for the modifier.
  - **Returns** - composite containing the UI components for the modifier

- *toString*
  `public abstract String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin`

(in 7.2.4, page 62)

- *compareTo*
  `public int` **compareTo**`( uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin` **o** `)`
- *getPriority*
  `public abstract int` **getPriority**`( )`
  - **Usage**
    - The plugin's priority.
  - **Returns** - the priority

1.1.11 Class **RunPanel**

Creates the user interface for the Run tab. The Run tab listens syncronously for NeuralNetworkSimulationEvents and NeuralNetworkTickEvents.

Declaration

public class RunPanel
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.events.EventHandler

Constructors

- *RunPanel*
  `public` **RunPanel**`( org.eclipse.swt.widgets.Composite` **parent**`,`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm** `)`
  - **Usage**
    - Create the Run tab.
  - **Parameters**
    - `parent` - - the tab container
    - `gm` - - the graph manager

Methods

- *flush*
  `public void` **flush**`( )`
- *getName*
  `public String` **getName**`( )`
- *handle*
  `public void` **handle**`( uk.ac.ic.doc.neuralnets.events.Event` **e** `)`
- *isValid*
  `public boolean` **isValid**`( )`

1.1.12 Class **ScrollingTextAppender**

Declaration

public class ScrollingTextAppender
extends org.apache.log4j.AppenderSkeleton

Constructors

- *ScrollingTextAppender*
  `public` **ScrollingTextAppender**`( )`

Methods

- *append*
  `protected void` **append**`( org.apache.log4j.spi.LoggingEvent` **e** `)`
- *close*
  `public void` **close**`( )`
- *requiresLayout*
  `public boolean` **requiresLayout**`( )`
- *setText*
  `public static void` **setText**`( org.eclipse.swt.custom.StyledText` **t** `)`

Methods inherited from class `org.apache.log4j.AppenderSkeleton`

- *activateOptions*
  `public void` **activateOptions**`( )`
- *addFilter*
  `public void` **addFilter**`( org.apache.log4j.spi.Filter` **arg0** `)`
- *append*
  `protected abstract void` **append**`( org.apache.log4j.spi.LoggingEvent` **arg0** `)`
- *clearFilters*
  `public void` **clearFilters**`( )`
- *doAppend*
  `public synchronized void` **doAppend**`( org.apache.log4j.spi.LoggingEvent` **arg0** `)`
- *finalize*
  `public void` **finalize**`( )`
- *getErrorHandler*
  `public ErrorHandler` **getErrorHandler**`( )`
- *getFilter*
  `public Filter` **getFilter**`( )`
- *getFirstFilter*
  `public final Filter` **getFirstFilter**`( )`
- *getLayout*
  `public Layout` **getLayout**`( )`
- *getName*
  `public final String` **getName**`( )`
- *getThreshold*
  `public Priority` **getThreshold**`( )`
- *isAsSevereAsThreshold*
  `public boolean` **isAsSevereAsThreshold**`( org.apache.log4j.Priority` **arg0** `)`
- *setErrorHandler*
  `public synchronized void` **setErrorHandler**`( org.apache.log4j.spi.ErrorHandler` **arg0** `)`

- *setLayout*
  `public void` **setLayout**`( org.apache.log4j.Layout` **arg0** `)`
- *setName*
  `public void` **setName**`( java.lang.String` **arg0** `)`
- *setThreshold*
  `public void` **setThreshold**`( org.apache.log4j.Priority` **arg0** `)`

1.1.13 Class **ToolbarPlugin**

ToolbarPlugins add buttons to the application toolbar.

Declaration

public abstract class ToolbarPlugin
extends uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin

Constructors

- *ToolbarPlugin*
  `public` **ToolbarPlugin**`( )`

Methods

- *create*
  `public abstract void` **create**`( uk.ac.ic.doc.neuralnets.gui.GUIToolbar` **toolbar** `)`
  - **Usage**
    - Create buttons to add to the toolbar.
      For example: `toolbar.addItem("MyItem"); toolbar.addButton("MyItem", "MyButton");`
  - **Parameters**
    - `toolbar` - - the application toolbar to which to add buttons

Methods inherited from class `uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin`

(in 7.2.4, page 62)

- *compareTo*
  `public int` **compareTo**`( uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin` **o** `)`
- *getPriority*
  `public abstract int` **getPriority**`( )`
  - **Usage**
    - The plugin's priority.
  - **Returns** - the priority

1.1.14 Class **TrainingPanel**

Create the Training Panel

Declaration

public class TrainingPanel
extends java.lang.Object

Constructors

- *TrainingPanel*
  `public` **TrainingPanel**`( org.eclipse.swt.widgets.Composite` **c**`,`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm** `)`

Chapter 2

Package uk.ac.ic.doc.neuralnets.graph.neural.manip

Package Contents *Page*

Classes

2.1 Classes

2.1.1 Class **EdgeCreatedEvent**

Declaration

public class EdgeCreatedEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *EdgeCreatedEvent*
  `public` **EdgeCreatedEvent**`( int` **num**`, int` **count** `)`

Methods

- *getEdgeCount*
  `public int` **getEdgeCount**`( )`
- *getEdgeNumber*
  `public int` **getEdgeNumber**`( )`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

2.1.2 Class **EdgeFactory**

Declaration

public class EdgeFactory
extends java.lang.Object
implements java.io.Serializable

Constructors

- *EdgeFactory*
  `public` **EdgeFactory**`( )`

Methods

- *create*
  `public Edge` **create**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeSpecification` **s** `)`
- *create*
  `public Edge` **create**`( uk.ac.ic.doc.neuralnets.graph.Node` **f**`, uk.ac.ic.doc.neuralnets.graph.Node` **t** `)`
- *get*
  `public static EdgeFactory` **get**`( )`

2.1.3 Class **GraphFactory**

Declaration

public class GraphFactory
extends java.lang.Object

Fields

- public static final int EVENT_RESOLUTION
  –

Constructors

- *GraphFactory*
  `public` **GraphFactory**`( )`

Methods

- *create*
  `public Graph` **create**`( java.lang.Class` **type**`, uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification` **ntype**`, int` **quantity** `)`
- *create*
  `public Graph` **create**`( uk.ac.ic.doc.neuralnets.graph.neural.manipulation.GraphSpecification` **spec** `)`
- *get*
  `public static GraphFactory` **get**`( )`
- *makeNetwork*
  `public NeuralNetwork` **makeNetwork**`( int` **n**`, double` **edgeProb** `)`

2.1.4 Class **GraphSpecification**

Declaration

public abstract class GraphSpecification
extends java.lang.Object

Constructors

- *GraphSpecification*
  `public` **GraphSpecification**`( )`
- *GraphSpecification*
  `public` **GraphSpecification**`( java.util.List` **nodes** `)`
- *GraphSpecification*
  `public` **GraphSpecification**`( java.util.List` **s**`, java.util.List` **ns**`,`
  `uk.ac.ic.doc.neuralnets.util.Transformer` **builder** `)`
- *GraphSpecification*
  `public` **GraphSpecification**`( uk.ac.ic.doc.neuralnets.util.Transformer` **builder**
  `)`

Methods

- *getEdgeBuilder*
  `public Transformer` **getEdgeBuilder**`( )`
- *getNodes*
  `public List` **getNodes**`( )`
- *getSpecifications*
  `public List` **getSpecifications**`( )`
- *getTarget*
  `public abstract Class` **getTarget**`( )`
- *separateNetworks*
  `public abstract boolean` **separateNetworks**`( )`

2.1.5 Class **HomogenousNetworkSpecification**

Declaration

public class HomogenousNetworkSpecification
extends uk.ac.ic.doc.neuralnets.graph.neural.manipulation.GraphSpecification

Constructors

- *HomogenousNetworkSpecification*
  `public` **HomogenousNetworkSpecification**`( java.lang.Integer` **nodes**`, double` **edgeProb** `)`
- *HomogenousNetworkSpecification*
  `public` **HomogenousNetworkSpecification**`( java.util.List` **nodes**`, double` **edgeProb** `)`
- *HomogenousNetworkSpecification*
  `public` **HomogenousNetworkSpecification**`( java.util.List` **specs**`, java.util.List` **nodes** `)`
- *HomogenousNetworkSpecification*
  `public` **HomogenousNetworkSpecification**`( java.util.List` **specs**`, java.util.List` **nodes**`, double` **edgeProb** `)`
- *HomogenousNetworkSpecification*
  `public` **HomogenousNetworkSpecification**`( uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification` **spec**`, double` **edgeProb** `)`
- *HomogenousNetworkSpecification*
  `public` **HomogenousNetworkSpecification**`( uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification` **spec**`, java.lang.Integer` **nodes** `)`
- *HomogenousNetworkSpecification*
  `public` **HomogenousNetworkSpecification**`( uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification` **spec**`, java.lang.Integer` **nodes**`, double` **edgeProb** `)`

Methods

- *getTarget*
  `public Class` **getTarget**`( )`
- *separateNetworks*
  `public boolean` **separateNetworks**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.manipulation.GraphSpecification`

(in 2.1.4, page 28)

- *getEdgeBuilder*
  `public Transformer` **getEdgeBuilder**`( )`
- *getNodes*
  `public List` **getNodes**`( )`
- *getSpecifications*
  `public List` **getSpecifications**`( )`

- *getTarget*
  `public abstract Class` **getTarget**`( )`
- *separateNetworks*
  `public abstract boolean` **separateNetworks**`( )`

2.1.6 Class **InhibitoryNodeSpecification**

Declaration

public class InhibitoryNodeSpecification
extends uk.ac.ic.doc.neuralnets.graph.neural.manipulation.SpikingNodeSpecification

Constructors

- *InhibitoryNodeSpecification*
  `public` **InhibitoryNodeSpecification**`( )`

Methods inherited from class
`uk.ac.ic.doc.neuralnets.graph.neural.manipulation.SpikingNodeSpecification`

(in 2.1.12, page 41)

Methods inherited from class
`uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification`

(in 17.2.15, page 169)

- *get*
  `public ASTExpression` **get**`( java.lang.String` **param** `)`
  - **Usage**
    - Get the AST expression for input parameter.
  - **Parameters**
    - `param` - String
  - **Returns** - AST expression
- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration**`( )`
  - **Usage**
    - Get the edge decoration for the node specification.
  - **Returns** - The edge decoration.
- *getName*
  `public String` **getName**`( )`
  - **Usage**
    - Get the name of the node specification.
  - **Returns** - The name.

- *getParameters*
  `public Set` **getParameters**`( )`
  - **Usage**
    - Get the parameter key set.
  - **Returns** - Parameter key set.
- *getTarget*
  `public Class` **getTarget**`( )`
  - **Usage**
    - Get target of node specification.
  - **Returns** - Target
- *set*
  `public NodeSpecification` **set**`( java.lang.String` **param**`, uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **target** `)`
  - **Usage**
    - Set a parameter to an AST expresion.
  - **Parameters**
    - `param` - Parameter name
    - `target` - AST expression value.
  - **Returns** - Itself.
- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration` **ed** `)`
  - **Usage**
    - Set the edge decorator for the node specification.
  - **Parameters**
    - `ed` - The edge decoration.
- *setName*
  `public void` **setName**`( java.lang.String` **n** `)`
  - **Usage**
    - Set name of node specification.
  - **Parameters**
    - `n` - Name

2.1.7 Class **InteractionUtils**

Declaration

public class InteractionUtils
extends java.lang.Object

Constructors

- *InteractionUtils*
  `public` **InteractionUtils**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Parameters**
    - ∗ `n` - The NeuralNetwork to operate over

Methods

- *bifurcate*
  `public NeuralNetwork` **bifurcate**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n**`, uk.ac.ic.doc.neuralnets.util.Transformer` **knife** `)`
  - **Usage**
    - ∗ Extract the nodes from n that are selected by the knife, removing them from the network and instead creating a new network.
      Any edges in n that are into or out of knife are instead routed via a NetworkBridge.
      The resultant network is added to the parent network of n automatically.
  - **Parameters**
    - ∗ `n` - The network to bifurcate
    - ∗ `knife` - A transformer to select the nodes to remove
  - **Returns** - The resultant (new) bifurcated network

- *connect*
  `public Collection` **connect**`( java.util.Collection` **f**`, java.util.Collection` **t** `)`
  - **Usage**
    - ∗ Fully connect the given sets of nodes in the network
  - **Parameters**
    - ∗ `f` - The source node
    - ∗ `t` - The target node
  - **Returns** - The collection of created edges

- *connect*
  `public Collection` **connect**`( java.util.Collection` **f**`, java.util.Collection` **t**`, double` **edgeProb** `)`
  - **Usage**
    - ∗ Connect the given sets of nodes in the network with the chosen probability of edge creation
  - **Parameters**
    - ∗ `f` - The source node
    - ∗ `t` - The target node
    - ∗ `edgeProb` - The probability a given edge is created
  - **Returns** - The collection of created edges

- *connect*
  `public Edge` **`connect`**`( uk.ac.ic.doc.neuralnets.graph.Node` **`f`**`, uk.ac.ic.doc.neuralnets.graph.Node` **`t`** `)`
  - **Usage**
    - Connect the given nodes in any networks. If the network of f is the same as the network of t, return a synpase in that network. Otherwise, create a bridge from network of f to network of t, and route a synapse through its bundle. If network of f is a super-node of the network of t, then bridges are still created. Bridges and synapses are always re-used where possible.
      Given a network with two sub-networks, n1 and n2, and n2 containing n3, a synapse from a neurone in n1 to a neurone in n3 most route over a network bridge to n2, then a network bridge from n2 to n3, and finally act as a synapse from n3's input to the synapse.
      Connecting a network to its parent results in a null connection, as it is not necessary.
  - **Parameters**
    - `f` - The node to connect from
    - `t` - The node to connect to
  - **Returns** - The edge that connects these nodes, or null if no such connection is possible

---

- *connect1to1*
  `public Collection` **`connect1to1`**`( java.util.Collection` **`f`**`, java.util.Collection` **`t`** `)`
  - **Usage**
    - Connect the given sets of nodes in the network with a 1-1 connection mapping (i.e. each node in f connects to one node in t) to as great an extent as possible. If there are insufficient nodes in t, some may be re-used
  - **Parameters**
    - `f` - The source node
    - `t` - The target node
  - **Returns** - The collection of created edges

- *createNodes*
  `public NeuralNetwork` **`createNodes`**`( uk.ac.ic.doc.neuralnets.graph.neural.manipulation.GraphSpecification` **`spec`** `)`
  - **Usage**
    - Create some nodes in the network
  - **Parameters**
    - `spec` - The specification of how to add nodes and edges
  - **Returns** - The nodes added, as a new network

- *createNodes*
  `public NeuralNetwork` **`createNodes`**`( int` **`nodes`**`, double` **`edgeProb`** `)`
  - **Usage**

    - Create some nodes in the network
  - **Parameters**
    - `nodes` - The number of nodes to create
    - `edgeProb` - The probability a given edge should be made
  - **Returns** - The nodes added, as a new network

- *findNetwork*
  `public NeuralNetwork` **findNetwork**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**
    - Find the network which contains the given node. NB: Our semantics of containment dictate that the root network is contained by itself.
  - **Parameters**
    - `n` - The node to seek
  - **Returns** - The NeuralNetwork that contains it, or null if such could not be found

---

- *getNetwork*
  `public NeuralNetwork` **getNetwork**`( )`
  - **Returns** - The NeuralNetwork that backs these utils

- *isSuper*
  `public boolean` **isSuper**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **a**`, uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **b** `)`
  - **Usage**
    - Answers whether network a is a parent of network b
  - **Parameters**
    - `a` - The parent node to test
    - `b` - The child node to seek
  - **Returns** - true iff a is a parent of b

- *isSuper*
  `public boolean` **isSuper**`( uk.ac.ic.doc.neuralnets.graph.Node` **a**`, uk.ac.ic.doc.neuralnets.graph.Node` **b** `)`
  - **Usage**
    - Answers whether Node a is a super-node of node b (i.e. a parent)
  - **Parameters**
    - `a` - The parent node to test
    - `b` - The child node to seek
  - **Returns** - true iff a is a parent of b

- *lowestCommonAncestor*
  `public NeuralNetwork` **lowestCommonAncestor**`( uk.ac.ic.doc.neuralnets.graph.Node` **a**`, uk.ac.ic.doc.neuralnets.graph.Node` **b** `)`
  - **Usage**

    - Find the lowest common ancestor of Nodes a and b; i.e. the deepest NeuralNetwork in the tree of networks that contains both a and b.
      Algorithm: Iterate up the parents of a and b until an intersection in the sets of their ancestors is found; at that point, we have th lowest common ancestor and can return
  - **Parameters**
    - `a` - The first node to seek
    - `b` - The second node to seek
  - **Returns** - The lowest common ancestor of a and b, or null if it could not be found (in a correct network, this shouldn't be possible)

- *pauseNetwork*
  `public void` **pauseNetwork**`( )`
  - **Usage**
    - Pause the network from running

- *prettyPrintNetwork*
  `public void` **prettyPrintNetwork**`( java.io.PrintStream` **out** `)`
  - **Usage**
    - Print out the network to the given PrintStream
  - **Parameters**
    - `out` - The PrintStream to which to print

- *resetNetwork*
  `public void` **resetNetwork**`( )`

- *runNetwork*
  `public void` **runNetwork**`( )`
  - **Usage**
    - Run the network from the last tick state (i.e. resume)

- *runNetwork*
  `public void` **runNetwork**`( int` **ticks** `)`
  - **Usage**
    - Run the network for the given number of ticks
  - **Parameters**
    - `ticks` - How long to run for, or <0 for ”forever”

- *setNetwork*
  `public void` **setNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Parameters**
    - `n` - The NeuralNetwork to operate over

- *teardown*
  `public void` **teardown**`( )`
  - **Usage**
    - Cause this instance to stop any threads it may have spawned, and release its resources. Any further operations have undefined behaviour.

2.1.8 Class **InteractionUtils.NetworkRunner**

The thread used to run the network asynchronously with the UI

Declaration

protected class InteractionUtils.NetworkRunner
extends java.lang.Thread

Constructors

- *InteractionUtils.NetworkRunner*
  `protected` **InteractionUtils.NetworkRunner**`( )`

Methods

- *getRemainingTicks*
  `public int` **getRemainingTicks**`( )`
- *kill*
  `public void` **kill**`( )`
- *pauseNetwork*
  `public void` **pauseNetwork**`( )`
- *run*
  `public void` **run**`( )`
- *runNetwork*
  `public void` **runNetwork**`( )`
- *runNetwork*
  `public void` **runNetwork**`( int` **ticks** `)`
- *setTicks*
  `public void` **setTicks**`( int` **ticks** `)`

Methods inherited from class `java.lang.Thread`

- *activeCount*
  `public static int` **activeCount**`( )`
- *checkAccess*
  `public final void` **checkAccess**`( )`
- *countStackFrames*
  `public native int` **countStackFrames**`( )`
- *currentThread*
  `public static native Thread` **currentThread**`( )`
- *destroy*
  `public void` **destroy**`( )`

- *dumpStack*
  `public static void` **dumpStack**`( )`
- *enumerate*
  `public static int` **enumerate**`( java.lang.Thread [] ` **arg0** `)`
- *getAllStackTraces*
  `public static Map` **getAllStackTraces**`( )`
- *getContextClassLoader*
  `public ClassLoader` **getContextClassLoader**`( )`
- *getDefaultUncaughtExceptionHandler*
  `public static Thread.UncaughtExceptionHandler` **getDefaultUncaughtExceptionHandler**`( )`
- *getId*
  `public long` **getId**`( )`
- *getName*
  `public final String` **getName**`( )`
- *getPriority*
  `public final int` **getPriority**`( )`
- *getStackTrace*
  `public StackTraceElement` **getStackTrace**`( )`
- *getState*
  `public Thread.State` **getState**`( )`
- *getThreadGroup*
  `public final ThreadGroup` **getThreadGroup**`( )`
- *getUncaughtExceptionHandler*
  `public Thread.UncaughtExceptionHandler` **getUncaughtExceptionHandler**`( )`
- *holdsLock*
  `public static native boolean` **holdsLock**`( java.lang.Object ` **arg0** `)`
- *interrupt*
  `public void` **interrupt**`( )`
- *interrupted*
  `public static boolean` **interrupted**`( )`
- *isAlive*
  `public final native boolean` **isAlive**`( )`
- *isDaemon*
  `public final boolean` **isDaemon**`( )`
- *isInterrupted*
  `public boolean` **isInterrupted**`( )`
- *join*
  `public final void` **join**`( )`
- *join*
  `public final synchronized void` **join**`( long ` **arg0** `)`
- *join*
  `public final synchronized void` **join**`( long ` **arg0**`, int ` **arg1** `)`
- *resume*
  `public final void` **resume**`( )`
- *run*
  `public void` **run**`( )`
- *setContextClassLoader*
  `public void` **setContextClassLoader**`( java.lang.ClassLoader ` **arg0** `)`
- *setDaemon*
  `public final void` **setDaemon**`( boolean ` **arg0** `)`

- *setDefaultUncaughtExceptionHandler*
  `public static void` **setDefaultUncaughtExceptionHandler**`(`
  `java.lang.Thread.UncaughtExceptionHandler` **arg0** `)`
- *setName*
  `public final void` **setName**`( java.lang.String` **arg0** `)`
- *setPriority*
  `public final void` **setPriority**`( int` **arg0** `)`
- *setUncaughtExceptionHandler*
  `public void` **setUncaughtExceptionHandler**`(`
  `java.lang.Thread.UncaughtExceptionHandler` **arg0** `)`
- *sleep*
  `public static native void` **sleep**`( long` **arg0** `)`
- *sleep*
  `public static void` **sleep**`( long` **arg0**`, int` **arg1** `)`
- *start*
  `public synchronized void` **start**`( )`
- *stop*
  `public final void` **stop**`( )`
- *stop*
  `public final synchronized void` **stop**`( java.lang.Throwable` **arg0** `)`
- *suspend*
  `public final void` **suspend**`( )`
- *toString*
  `public String` **toString**`( )`
- *yield*
  `public static native void` **yield**`( )`

2.1.9 Class NodeCreatedEvent

Declaration

public class NodeCreatedEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *NodeCreatedEvent*
  `public` **NodeCreatedEvent**`( int` **num**`, int` **count** `)`

Methods

- *getNodeCount*
  `public int` **getNodeCount**`( )`
- *getNodeNumber*
  `public int` **getNodeNumber**`( )`
- *toString*
  `public String` **toString**`( )`

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

2.1.10 CLASS **NodeFactory**

DECLARATION

public class NodeFactory
extends java.lang.Object
implements java.io.Serializable

CONSTRUCTORS

- *NodeFactory*
  `public` **NodeFactory**`( )`

METHODS

- *create*
  `public Neurone` **create**`( )`
- *create*
  `public Node` **create**`( uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification  s )`
- *get*
  `public static NodeFactory` **get**`( )`

2.1.11 CLASS **PerceptronSpecification**

DECLARATION

public class PerceptronSpecification
extends uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification

CONSTRUCTORS

- *PerceptronSpecification*
  `public` **PerceptronSpecification**`( )`

Methods inherited from class
`uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification`

(in 17.2.15, page 169)

- *get*
  `public ASTExpression` **get**`( java.lang.String` **param** `)`
  - **Usage**
    - Get the AST expression for input parameter.
  - **Parameters**
    - `param` - String
  - **Returns** - AST expression
- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration**`( )`
  - **Usage**
    - Get the edge decoration for the node specification.
  - **Returns** - The edge decoration.
- *getName*
  `public String` **getName**`( )`
  - **Usage**
    - Get the name of the node specification.
  - **Returns** - The name.
- *getParameters*
  `public Set` **getParameters**`( )`
  - **Usage**
    - Get the parameter key set.
  - **Returns** - Parameter key set.
- *getTarget*
  `public Class` **getTarget**`( )`
  - **Usage**
    - Get target of node specification.
  - **Returns** - Target
- *set*
  `public NodeSpecification` **set**`( java.lang.String` **param**`,`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **target** `)`
  - **Usage**
    - Set a parameter to an AST expresion.
  - **Parameters**
    - `param` - Parameter name
    - `target` - AST expression value.
  - **Returns** - Itself.
- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration`
  ed `)`
  - **Usage**
    - Set the edge decorator for the node specification.

- **Parameters**
  - `ed` - The edge decoration.

- *setName*
  `public void` **setName**`( java.lang.String` **n** `)`
  - **Usage**
    - Set name of node specification.
  - **Parameters**
    - `n` - Name

2.1.12 Class SpikingNodeSpecification

Default NodeSpecification for SpikingNeurones

Declaration

public class SpikingNodeSpecification
extends uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification

Constructors

- *SpikingNodeSpecification*
  `public` **SpikingNodeSpecification**`( )`

Methods inherited from class
`uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification`

(in 17.2.15, page 169)

- *get*
  `public ASTExpression` **get**`( java.lang.String` **param** `)`
  - **Usage**
    - Get the AST expression for input parameter.
  - **Parameters**
    - `param` - String
  - **Returns** - AST expression

- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration**`( )`
  - **Usage**
    - Get the edge decoration for the node specification.
  - **Returns** - The edge decoration.

- *getName*
  `public String` **getName**`( )`
  - **Usage**
    - Get the name of the node specification.
  - **Returns** - The name.

- *getParameters*
  `public Set` **getParameters**`( )`
  - **Usage**
    - Get the parameter key set.
  - **Returns** - Parameter key set.

- *getTarget*
  `public Class` **getTarget**`( )`
  - **Usage**
    - Get target of node specification.
  - **Returns** - Target

- *set*
  `public NodeSpecification` **set**`( java.lang.String` **param**`,`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **target** `)`
  - **Usage**
    - Set a parameter to an AST expresion.
  - **Parameters**
    - `param` - Parameter name
    - `target` - AST expression value.
  - **Returns** - Itself.

- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration`
  ed `)`
  - **Usage**
    - Set the edge decorator for the node specification.
  - **Parameters**
    - `ed` - The edge decoration.

- *setName*
  `public void` **setName**`( java.lang.String` **n** `)`
  - **Usage**
    - Set name of node specification.
  - **Parameters**
    - `n` - Name

Chapter 3

Package uk.ac.ic.doc.neuralnets.gui.graph.events

Package Contents *Page*

Classes

3.1 Classes

3.1.1 Class **ChargeUpdateHandler**

Declaration

public class ChargeUpdateHandler
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.events.EventHandler

Constructors

- *ChargeUpdateHandler*
  `public` **ChargeUpdateHandler**`( )`
- *ChargeUpdateHandler*
  `public` **ChargeUpdateHandler**`(`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **m** `)`

Methods

- *flush*
  `public void` **flush**`( )`
- *getName*
  `public String` **getName**`( )`
- *handle*
  `public void` **handle**`( uk.ac.ic.doc.neuralnets.events.Event` **e** `)`
- *isValid*
  `public boolean` **isValid**`( )`
- *setGUIManager*
  `public void` **setGUIManager**`(`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **m** `)`

3.1.2 Class **NeuroneTypesPersister**

Declaration

public class NeuroneTypesPersister
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.events.EventHandler

Constructors

- *NeuroneTypesPersister*
  public **NeuroneTypesPersister**( )

Methods

- *flush*
  public void **flush**( )
- *getName*
  public String **getName**( )
- *handle*
  public void **handle**(uk.ac.ic.doc.neuralnets.events.Event **e**)
- *isValid*
  public boolean **isValid**( )

3.1.3 Class **NodeLocationUpdater**

Declaration

public class NodeLocationUpdater
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.events.EventHandler

Constructors

- *NodeLocationUpdater*
  public **NodeLocationUpdater**(
  uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager **gm**)

Methods

- *flush*
  public void **flush**( )
- *getName*
  public String **getName**( )
- *handle*
  public void **handle**(uk.ac.ic.doc.neuralnets.events.Event **e**)
- *isValid*
  public boolean **isValid**( )

3.1.4 Class **ToolTipUpdater**

Declaration

public class ToolTipUpdater
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.events.EventHandler

Constructors

- *ToolTipUpdater*
  `public` **ToolTipUpdater**`(`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm** `)`

Methods

- *flush*
  `public void` **flush**`( )`
- *getName*
  `public String` **getName**`( )`
- *handle*
  `public void` **handle**`( uk.ac.ic.doc.neuralnets.events.Event` **e** `)`
- *isValid*
  `public boolean` **isValid**`( )`

Chapter 4

Package uk.ac.ic.doc.neuralnets.gui.statistics

Package Contents *Page*

Classes

4.1 Classes

4.1.1 Class **StatisticianConfig**

Basic Statistician Configuration interface. Statisticians are EventHandlers designed to harvest data from events during the running of a neural network. StatisticianConfigs can be used to configure/disable Statisticians.

Declaration

public abstract class StatisticianConfig
extends uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin

Constructors

- *StatisticianConfig*
  `public` **StatisticianConfig**`( )`

Methods

- *configure*
  `public abstract EventHandler` **configure**`( org.eclipse.swt.widgets.Shell` **parent** `)`
  - **Usage**
    - Perform an operations required to configure a new statistician.
  - **Parameters**
    - `parent` - - shell access, for user interaction
  - **Returns** - the configured event handler

- *disable*
  `public abstract void` **disable**`( uk.ac.ic.doc.neuralnets.events.EventHandler` **h** `)`
  - **Usage**
    - Disable a statistician
  - **Parameters**
    - `h` - the event handler to disable

- *getTargetEvents*
  `public Class` **getTargetEvents**`( )`
  - **Usage**
    - Defines which events this statistician listens for.
  - **Returns** -

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin`

(in 7.2.4, page 62)

- *compareTo*
  `public int` **compareTo**`( uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin` **o** `)`
- *getPriority*
  `public abstract int` **getPriority**`( )`
  - **Usage**
    - The plugin's priority.
  - **Returns** - the priority

Chapter 5

Package uk.ac.ic.doc.neuralnets.util

Package Contents *Page*

Interfaces

Classes

5.1 Interfaces

5.1.1 Interface **Transformer**

General purpose Transformer from one data-type to another

Declaration

public interface Transformer
implements java.io.Serializable

Methods

- *transform*
  `public Object` **transform**`( java.lang.Object` **input** `)`
  - **Usage**
    - Transform input object
  - **Parameters**
    - `input` - - the object to transform
  - **Returns** - the transformed object

5.2 Classes

5.2.1 Class **Container**

Simple container for another object, for use when a final object is required but cannot be furnished yet

Declaration

public class Container
extends java.lang.Object

Constructors

- *Container*
  `public` **Container**`( )`
  - **Usage**
    - Create an empty container

- *Container*
  `public` **Container**`( java.lang.Object` **contents** `)`
  - **Usage**

    - * Create a container with contents of type T.
  - **Parameters**
    - * `contents` -

Methods

- *get*
  `public Object get( )`
  - **Usage**
    - * Get the content of the container.
  - **Returns** - the container contents

- *set*
  `public void set( java.lang.Object t )`
  - **Usage**
    - * Set the content of the container.
  - **Parameters**
    - * `t` - - the object to store in the container

Chapter 6

Package uk.ac.ic.doc.neuralnets.util.configuration

6.1 Interfaces

6.1.1 Interface **Configurator**

Configurators are Plugins that are run once at application load-time. They are intended for configuring external libraries such as Log4J.

Declaration

```
public interface Configurator
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin
```

Methods

- *configure*
  `public void` **configure**`( )`
  - **Usage**
    - Perform any required actions for configuration

6.2 Classes

6.2.1 Class **ConfigurationManager**

The ConfigurationManager controls Configurator objects, calling their `configure` methods at application load time.

Declaration

```
public class ConfigurationManager
extends java.lang.Object
```

Fields

- public static final File config
  - Master configuration file.

Constructors

- *ConfigurationManager*
  `public` **ConfigurationManager**`( )`

Methods

- *configure*
  `public static void` **configure**`( )`
  - **Usage**
    - Configure all configurators found in conf/configurator.cfg.

Chapter 7

Package uk.ac.ic.doc.neuralnets.util.plugins

Package Contents *Page*

Interfaces

Classes

7.1 Interfaces

7.1.1 Interface **Plugin**

Generic Plugin interface. All plugin types must extend or implement this interface. The class name of an extending plugin type must be unique. Plugins can not directly implement the Plugin interface, i.e. a plugin must be a descendant of a sub-type of Plugin.

Declaration

```
public interface Plugin
```

Methods

- *getName*
  `public String` **getName**`( )`
  - **Usage**
    - Get the canonical name of this Plugin, used to identify it
  - **Returns** - The canonical name of the loaded plugin

7.2 Classes

7.2.1 Class **PluginLoader**

The PluginLoader is responsible for loading plugin class files from the /plugin directory into the virtual machine.

Declaration

```
public class PluginLoader
extends java.lang.ClassLoader
```

Constructors

- *PluginLoader*
  `public` **PluginLoader**`( java.lang.String` **searchPath** `)`

Methods

- *findClass*
  `public Class` **findClass**`( java.lang.String` **name** `)`

Methods inherited from class `java.lang.ClassLoader`

- *clearAssertionStatus*
  `public synchronized void` **clearAssertionStatus**`( )`
- *defineClass*
  `protected final Class` **defineClass**`( byte [] ` **arg0**`, int ` **arg1**`, int ` **arg2** `)`
- *defineClass*
  `protected final Class` **defineClass**`( java.lang.String ` **arg0**`, byte [] ` **arg1**`, int ` **arg2**`, int ` **arg3** `)`
- *defineClass*
  `protected final Class` **defineClass**`( java.lang.String ` **arg0**`, byte [] ` **arg1**`, int ` **arg2**`, int ` **arg3**`, java.security.ProtectionDomain ` **arg4** `)`
- *defineClass*
  `protected final Class` **defineClass**`( java.lang.String ` **arg0**`, java.nio.ByteBuffer ` **arg1**`, java.security.ProtectionDomain ` **arg2** `)`
- *definePackage*
  `protected Package` **definePackage**`( java.lang.String ` **arg0**`, java.lang.String ` **arg1**`, java.lang.String ` **arg2**`, java.lang.String ` **arg3**`, java.lang.String ` **arg4**`, java.lang.String ` **arg5**`, java.lang.String ` **arg6**`, java.net.URL ` **arg7** `)`
- *findClass*
  `protected Class` **findClass**`( java.lang.String ` **arg0** `)`
- *findLibrary*
  `protected String` **findLibrary**`( java.lang.String ` **arg0** `)`
- *findLoadedClass*
  `protected final Class` **findLoadedClass**`( java.lang.String ` **arg0** `)`
- *findResource*
  `protected URL` **findResource**`( java.lang.String ` **arg0** `)`
- *findResources*
  `protected Enumeration` **findResources**`( java.lang.String ` **arg0** `)`
- *findSystemClass*
  `protected final Class` **findSystemClass**`( java.lang.String ` **arg0** `)`
- *getPackage*
  `protected Package` **getPackage**`( java.lang.String ` **arg0** `)`
- *getPackages*
  `protected Package` **getPackages**`( )`
- *getParent*
  `public final ClassLoader` **getParent**`( )`
- *getResource*
  `public URL` **getResource**`( java.lang.String ` **arg0** `)`
- *getResourceAsStream*
  `public InputStream` **getResourceAsStream**`( java.lang.String ` **arg0** `)`
- *getResources*
  `public Enumeration` **getResources**`( java.lang.String ` **arg0** `)`
- *getSystemClassLoader*
  `public static ClassLoader` **getSystemClassLoader**`( )`
- *getSystemResource*
  `public static URL` **getSystemResource**`( java.lang.String ` **arg0** `)`
- *getSystemResourceAsStream*
  `public static InputStream` **getSystemResourceAsStream**`( java.lang.String ` **arg0** `)`
- *getSystemResources*
  `public static Enumeration` **getSystemResources**`( java.lang.String ` **arg0** `)`

- *loadClass*
  `public Class` **loadClass**`( java.lang.String` **arg0** `)`
- *loadClass*
  `protected synchronized Class` **loadClass**`( java.lang.String` **arg0**`, boolean` **arg1** `)`
- *resolveClass*
  `protected final void` **resolveClass**`( java.lang.Class` **arg0** `)`
- *setClassAssertionStatus*
  `public synchronized void` **setClassAssertionStatus**`( java.lang.String` **arg0**`, boolean` **arg1** `)`
- *setDefaultAssertionStatus*
  `public synchronized void` **setDefaultAssertionStatus**`( boolean` **arg0** `)`
- *setPackageAssertionStatus*
  `public synchronized void` **setPackageAssertionStatus**`( java.lang.String` **arg0**`, boolean` **arg1** `)`
- *setSigners*
  `protected final void` **setSigners**`( java.lang.Class` **arg0**`, java.lang.Object []` **arg1** `)`

7.2.2 Class **PluginLoadException**

Throw when there are unrecoverable errors whilst attempting to instantiate a plugin.

Declaration

public class PluginLoadException
extends java.lang.Exception

Fields

- public static final long serialVersionUID
  –

Constructors

- *PluginLoadException*
  `public` **PluginLoadException**`( java.lang.String` **m** `)`
- *PluginLoadException*
  `public` **PluginLoadException**`( java.lang.String` **m**`, java.lang.Throwable` **e** `)`
- *PluginLoadException*
  `public` **PluginLoadException**`( java.lang.Throwable` **e** `)`

Methods inherited from class `java.lang.Exception`

Methods inherited from class `java.lang.Throwable`

- *fillInStackTrace*
  `public synchronized native Throwable` **fillInStackTrace**`( )`
- *getCause*
  `public Throwable` **getCause**`( )`
- *getLocalizedMessage*
  `public String` **getLocalizedMessage**`( )`
- *getMessage*
  `public String` **getMessage**`( )`
- *getStackTrace*
  `public StackTraceElement` **getStackTrace**`( )`
- *initCause*
  `public synchronized Throwable` **initCause**`( java.lang.Throwable` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( )`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintStream` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintWriter` **arg0** `)`
- *setStackTrace*
  `public void` **setStackTrace**`( java.lang.StackTraceElement []` **arg0** `)`
- *toString*
  `public String` **toString**`( )`

7.2.3 Class **PluginManager**

The PluginManager is responsible for managing the class loading and instantiation of plugins from the plugins directory. Plugins are loaded and cached by the PluginLoader.

Declaration

public class PluginManager
extends java.lang.Object

Fields

- public static final File searchPath
  - Path to plugin directory

Methods

- *checkValidity*
  `public void` **checkValidity**`( )`
  - **Usage**
    - Check the validity of all the plugins in this PluginManager. If any have been loaded that are invalid, remove them from this PluginManager

---

- *checkValidity*
  `public void` **checkValidity**`( java.lang.Class` **clazz** `)`
  - **Usage**
    - Check the validity of all the plugins of the given type. If any have been loaded that are invalid, remove them from this PluginManager
  - **Parameters**
    - `clazz` - The class of the plugin type

- *checkValidity*
  `public void` **checkValidity**`( java.lang.String` **type** `)`
  - **Usage**
    - Check the validity of all the plugins of the given type. If any have been loaded that are invalid, remove them from this PluginManager
  - **Parameters**
    - `type` - The type name of the plugin

- *get*
  `public static PluginManager` **get**`( )`
  - **Usage**
    - Retrieve the instance of the PluginManager.
  - **Returns** - the PluginManager instance
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.util.plugins.PluginLoadException` -

- *getPlugin*
  `public Plugin` **getPlugin**`( java.lang.String` **name**`, java.lang.Class` **clazz** `)`
  - **Usage**
    - Load the requested plugin and cast it to the given class
  - **Parameters**
    - `name` - The name of the plugin
    - `clazz` - The class to which it must be cast
  - **Returns** - A Plugin object of type T

- *getPlugin*
  `public Plugin` **getPlugin**`( java.lang.String` **name**`, java.lang.String` **type** `)`
  - **Usage**

    - Load the requested plugin and cast it to the given class
  - **Parameters**
    - `name` - The name of the plugin
    - `type` - The type of the plugin to fetch
  - **Returns** - A Plugin object of the given name and type

- *getPluginsOftype*
  `public Set` **getPluginsOftype**`( java.lang.Class` **clazz** `)`
  - **Usage**
    - Answer all the plugins of the given type
  - **Parameters**
    - `clazz` - The class of the type of plugin to find
  - **Returns** - A set of plugin names

- *getPluginsOfType*
  `public Set` **getPluginsOfType**`( java.lang.String` **type** `)`
  - **Usage**
    - Answer all the plugins of the given type
  - **Parameters**
    - `type` - The type of the plugin to find
  - **Returns** - A set of plugin names

- *refreshPlugins*
  `public void` **refreshPlugins**`( )`

7.2.4 Class PriorityPlugin

PriorityPlugin extends the plugin interface allowing an ordering to be applied. The ordering can be achieved in two ways: by implementing the `getPriority` to return the plugin's priority, or by overriding the `compareTo` method if more detailed comparison is required.

Declaration

public abstract class PriorityPlugin
extends java.lang.Object
implements java.lang.Comparable, Plugin

Constructors

- *PriorityPlugin*
  `public` **PriorityPlugin**`( )`

Methods

- *compareTo*
  `public int` **compareTo**`( uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin` **o** `)`

- *getPriority*
  `public abstract int` **getPriority**`( )`
  - **Usage**
    - The plugin's priority.
  - **Returns** - the priority

Chapter 8

Package uk.ac.ic.doc.neuralnets.graph.neural.io

8.1 Interfaces

8.1.1 Interface **Foldable**

Declaration

public interface Foldable

Methods

- *fold*
  `public void` **fold**`( int` **foldNumber**`, int` **folds** `)`
  - **Usage**
    - Instruct this foldable to prepare for the next fold
  - **Parameters**
    - `foldNumber` - The number of the current fold to prepare
    - `folds` - The number of folds total

8.2 Classes

8.2.1 Class **InputNode**

Declaration

public abstract class InputNode
extends uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin, Foldable

Constructors

- *InputNode*
  `public` **InputNode**`( )`

Methods

- *configure*
  `public abstract void` **configure**`( )`
- *destroy*
  `public abstract void` **destroy**`( )`

- *fold*
  `public void` **fold**`( int` **foldNumber**`, int` **folds** `)`

- *getData*
  `public PartitionableMatrix` **getData**`( )`

- *getTargets*
  `public PartitionableMatrix` **getTargets**`( )`

- *recreate*
  `public abstract void` **recreate**`( )`

- *setRow*
  `public void` **setRow**`( int` **row** `)`
  - **Usage**
    * Set the current row of data to use for input. Is fold-sensitive (row N is different per fold).
  - **Parameters**
    * `row` - The number of the row to seek to

- *toNetwork*
  `public NeuralNetwork` **toNetwork**`( )`
  - **Usage**
    * Sends data to the network.
  - **Returns** - Itself.

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork`

(in 17.2.5, page 154)

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.neural.NetworkBridge` **e** `)`
- *getIncoming*
  `public Collection` **getIncoming**`( )`
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
- *getTicks*
  `public int` **getTicks**`( )`
- *getX*
  `public int` **getX**`( )`
- *getY*
  `public int` **getY**`( )`
- *getZ*
  `public int` **getZ**`( )`
- *resetTicks*
  `public void` **resetTicks**`( )`

- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
- *setPos*
  `public void` **setPos**`( int` **x**`, int` **y**`, int` **z** `)`
- *tick*
  `public Node` **tick**`( )`
- *type*
  `protected String` **type**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.Graph`

(in 18.2.1, page 184)

- *addAllNodes*
  `public Graph` **addAllNodes**`( java.util.Collection` **ns** `)`
  - **Usage**
    - Adds a collection of nodes to the graph, only if that collection doesn't contain itself.
  - **Parameters**
    - `ns` - Collection of nodes to add.
  - **Returns** - Itself with the nodes added or not added.
- *addEdge*
  `public Graph` **addEdge**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - Adds an edge to the graph and adds its start and end nodes to the graph.
  - **Parameters**
    - `e` - Edge to add.
  - **Returns** - Itself
- *addNode*
  `public Graph` **addNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**
    - Adds input node to the graph as long as input node is not itself, returns itself.
  - **Parameters**
    - `n` - Node to add.
  - **Returns** - Itself with the node added or not added.
- *forEachEdge*
  `public Graph` **forEachEdge**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c** `)`
  - **Usage**
    - Conducts a command on each edge within the graph.
  - **Parameters**
    - `c` - Command to execute.
  - **Returns** - Itself.
- *forEachNode*
  `public Graph` **forEachNode**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c** `)`
  - **Usage**
    - Conducts a command on each node within the graph.
  - **Parameters**
    - `c` - Command to execute.

  - **Returns** - Itself.

- *getEdges*
  `public Collection` **getEdges**`( )`
  - **Usage**
    - Gets the edges from within.
  - **Returns** - The edges.

- *getFreshID*
  `public void` **getFreshID**`( )`
  - **Usage**
    - Sets the id of the object to a new fresh id.

- *getID*
  `public int` **getID**`( )`
  - **Usage**
    - Gets the id of the object.
  - **Returns** - The id.

- *getNodes*
  `public Collection` **getNodes**`( )`
  - **Usage**
    - Gets the nodes from within.
  - **Returns** - The nodes.

- *merge*
  `public Graph` **merge**`( uk.ac.ic.doc.neuralnets.graph.Graph` **o** `)`
  - **Usage**
    - Merges one graph with its self, as all the edges and nodes.
  - **Parameters**
    - `o` - Graph to merge with.
  - **Returns** - Itself

- *setID*
  `public void` **setID**`( int` **id** `)`
  - **Usage**
    - Sets the id of the object to parameter.
  - **Parameters**
    - `int` - New id.

- *toString*
  `public String` **toString**`( )`

- *type*
  `protected String` **type**`( )`
  - **Usage**
    - Returns the object type.
  - **Returns** - Object type.

8.2.2 Class **IONeurone**

Purely a class to ”mark” a neurone as being for I/O purposes.

Declaration

public class IONeurone
extends uk.ac.ic.doc.neuralnets.graph.neural.Neurone

Serializable Fields

- private boolean concrete
  –

Constructors

- *IONeurone*
  `public` **IONeurone**`( )`

Methods

- *getCharge*
  `public double` **getCharge**`( )`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.Neurone`

(in 17.2.8, page 159)

- *charge*
  `public Neurone` **charge**`( double` **amt** `)`
- *getCharge*
  `public double` **getCharge**`( )`
- *getCurrentCharge*
  `public Double` **getCurrentCharge**`( )`
- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration**`( )`
- *getFreshID*
  `public void` **getFreshID**`( )`
- *getID*
  `public int` **getID**`( )`
- *getSquashFunction*
  `public ASTExpression` **getSquashFunction**`( )`
- *getTrigger*
  `public double` **getTrigger**`( )`
- *reset*
  `public void` **reset**`( )`

- *setCharge*
  `public void` **setCharge**`( double` **charge** `)`
- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration` **ed** `)`
- *setID*
  `public void` **setID**`( int` **id** `)`
- *setInitialCharge*
  `public void` **setInitialCharge**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **c** `)`
- *setSquashFunction*
  `public void` **setSquashFunction**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setTrigger*
  `public void` **setTrigger**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **t** `)`
- *setTrigger*
  `public void` **setTrigger**`( double` **d** `)`
- *tick*
  `public Node` **tick**`( )`
  - **Usage**
    - Ticks the neurone one step forward. Fires the neurone is appropriate.
  - **Returns** - Itself.
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.NodeBase`

(in 17.2.12, page 164)

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - Connect this node up with the input edge.
- *getIncoming*
  `public Collection` **getIncoming**`( )`
  - **Usage**
    - Get incoming edges.
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
  - **Usage**
    - Returns the meta data for the key input.
  - **Parameters**
    - `key` - To look for.
  - **Returns** - item Found.
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
  - **Usage**
    - Get outgoing edges.

- *getX*
  `public int` **getX**`( )`
  - **Usage**
    - Returns the position of the node on the x axis.
  - **Returns** - x axis position.
- *getY*
  `public int` **getY**`( )`
  - **Usage**
    - Returns the position of the node on the y axis.
  - **Returns** - y axis position.
- *getZ*
  `public int` **getZ**`( )`
  - **Usage**
    - Returns the position of the node on the z axis.
  - **Returns** - z axis position.
- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
  - **Usage**
    - Set meta data for the object.
  - **Parameters**
    - `key` - String key
    - `item` - String item
- *setPos*
  `public void` **setPos**`( int` **x**`, int` **y**`, int` **z** `)`
  - **Usage**
    - Sets the position of the node.
  - **Parameters**
    - `x` - Position on x axis.
    - `y` - Position on y axis.
    - `z` - Position on z axis.
- *setX*
  `public void` **setX**`( int` **x** `)`
  - **Usage**
    - Sets the position of the node on the x axis.
  - **Parameters**
    - `x` - Position on x axis.
- *setY*
  `public void` **setY**`( int` **y** `)`
  - **Usage**
    - Sets the position of the node on the y axis.
  - **Parameters**
    - `y` - Position on y axis.
- *setZ*
  `public void` **setZ**`( int` **z** `)`

- **Usage**
  - Sets the position of the node on the z axis.
- **Parameters**
  - `z` - Position on z axis.

- *tick*
  `public abstract Node` **tick**`( )`
- *toString*
  `public abstract String` **toString**`( )`

8.2.3 Class **OutputNode**

Declaration

public abstract class OutputNode
extends uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin

Constructors

- *OutputNode*
  `public` **OutputNode**`( )`
- *OutputNode*
  `public` **OutputNode**`( int` **nodes** `)`

Methods

- *destroy*
  `public abstract void` **destroy**`( )`
- *fire*
  `protected abstract void` **fire**`( int` **n**`, java.lang.Double` **amt** `)`
- *recreate*
  `public abstract void` **recreate**`( )`
- *setNodes*
  `protected abstract void` **setNodes**`( int` **n** `)`
- *toNetwork*
  `public NeuralNetwork` **toNetwork**`( int` **nodes** `)`
- *toString*
  `public String` **toString**`( )`

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork`

(in 17.2.5, page 154)

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.neural.NetworkBridge` **e** `)`
- *getIncoming*
  `public Collection` **getIncoming**`( )`
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
- *getTicks*
  `public int` **getTicks**`( )`
- *getX*
  `public int` **getX**`( )`
- *getY*
  `public int` **getY**`( )`
- *getZ*
  `public int` **getZ**`( )`
- *resetTicks*
  `public void` **resetTicks**`( )`
- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
- *setPos*
  `public void` **setPos**`( int` **x**`, int` **y**`, int` **z** `)`
- *tick*
  `public Node` **tick**`( )`
- *type*
  `protected String` **type**`( )`

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.graph.Graph`

(in 18.2.1, page 184)

- *addAllNodes*
  `public Graph` **addAllNodes**`( java.util.Collection` **ns** `)`
  - **Usage**
    - Adds a collection of nodes to the graph, only if that collection doesn't contain itself.
  - **Parameters**
    - `ns` - Collection of nodes to add.
  - **Returns** - Itself with the nodes added or not added.
- *addEdge*
  `public Graph` **addEdge**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - Adds an edge to the graph and adds its start and end nodes to the graph.
  - **Parameters**
    - `e` - Edge to add.
  - **Returns** - Itself

- *addNode*
  `public Graph` **addNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**
    - Adds input node to the graph as long as input node is not itself, returns itself.
  - **Parameters**
    - `n` - Node to add.
  - **Returns** - Itself with the node added or not added.

- *forEachEdge*
  `public Graph` **forEachEdge**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c** `)`
  - **Usage**
    - Conducts a command on each edge within the graph.
  - **Parameters**
    - `c` - Command to execute.
  - **Returns** - Itself.

- *forEachNode*
  `public Graph` **forEachNode**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c** `)`
  - **Usage**
    - Conducts a command on each node within the graph.
  - **Parameters**
    - `c` - Command to execute.
  - **Returns** - Itself.

- *getEdges*
  `public Collection` **getEdges**`( )`
  - **Usage**
    - Gets the edges from within.
  - **Returns** - The edges.

- *getFreshID*
  `public void` **getFreshID**`( )`
  - **Usage**
    - Sets the id of the object to a new fresh id.

- *getID*
  `public int` **getID**`( )`
  - **Usage**
    - Gets the id of the object.
  - **Returns** - The id.

- *getNodes*
  `public Collection` **getNodes**`( )`
  - **Usage**
    - Gets the nodes from within.
  - **Returns** - The nodes.

- *merge*
  `public Graph` **merge**`( uk.ac.ic.doc.neuralnets.graph.Graph` **o** `)`
  - **Usage**
    - Merges one graph with its self, as all the edges and nodes.

  - **Parameters**
    - `o` - Graph to merge with.
  - **Returns** - Itself

- *setID*
  `public void` **setID**`( int` **id** `)`
  - **Usage**
    - Sets the id of the object to parameter.
  - **Parameters**
    - `int` - New id.

- *toString*
  `public String` **toString**`( )`

- *type*
  `protected String` **type**`( )`
  - **Usage**
    - Returns the object type.
  - **Returns** - Object type.

8.2.4 Class **ValueReportingOutputNode**

Declaration

public class ValueReportingOutputNode
extends uk.ac.ic.doc.neuralnets.graph.neural.io.OutputNode

Serializable Fields

- private List values
  -

Constructors

- *ValueReportingOutputNode*
  `public` **ValueReportingOutputNode**`( )`

Methods

- *destroy*
  `public void` **destroy**`( )`

- *fire*
  `protected void` **fire**`( int` **n**`, java.lang.Double` **amt** `)`

- *getName*
  `public String` **getName**`( )`
- *getValues*
  `public List` **getValues**`( )`
- *recreate*
  `public void` **recreate**`( )`
- *setNodes*
  `protected void` **setNodes**`( int` **n** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.io.OutputNode`

(in 8.2.3, page 72)

- *destroy*
  `public abstract void` **destroy**`( )`
- *fire*
  `protected abstract void` **fire**`( int` **n**`, java.lang.Double` **amt** `)`
- *recreate*
  `public abstract void` **recreate**`( )`
- *setNodes*
  `protected abstract void` **setNodes**`( int` **n** `)`
- *toNetwork*
  `public NeuralNetwork` **toNetwork**`( int` **nodes** `)`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork`

(in 17.2.5, page 154)

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.neural.NetworkBridge` **e** `)`
- *getIncoming*
  `public Collection` **getIncoming**`( )`
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
- *getTicks*
  `public int` **getTicks**`( )`
- *getX*
  `public int` **getX**`( )`
- *getY*
  `public int` **getY**`( )`
- *getZ*
  `public int` **getZ**`( )`
- *resetTicks*
  `public void` **resetTicks**`( )`
- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`

- *setPos*
  `public void` **setPos**`( int  x, int  y, int  z )`
- *tick*
  `public Node` **tick**`( )`
- *type*
  `protected String` **type**`( )`

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.graph.Graph`

(in 18.2.1, page 184)

- *addAllNodes*
  `public Graph` **addAllNodes**`( java.util.Collection  ns )`
  - **Usage**
    - Adds a collection of nodes to the graph, only if that collection doesn't contain itself.
  - **Parameters**
    - `ns` - Collection of nodes to add.
  - **Returns** - Itself with the nodes added or not added.
- *addEdge*
  `public Graph` **addEdge**`( uk.ac.ic.doc.neuralnets.graph.Edge  e )`
  - **Usage**
    - Adds an edge to the graph and adds its start and end nodes to the graph.
  - **Parameters**
    - `e` - Edge to add.
  - **Returns** - Itself
- *addNode*
  `public Graph` **addNode**`( uk.ac.ic.doc.neuralnets.graph.Node  n )`
  - **Usage**
    - Adds input node to the graph as long as input node is not itself, returns itself.
  - **Parameters**
    - `n` - Node to add.
  - **Returns** - Itself with the node added or not added.
- *forEachEdge*
  `public Graph` **forEachEdge**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command  c )`
  - **Usage**
    - Conducts a command on each edge within the graph.
  - **Parameters**
    - `c` - Command to execute.
  - **Returns** - Itself.
- *forEachNode*
  `public Graph` **forEachNode**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command  c )`
  - **Usage**
    - Conducts a command on each node within the graph.
  - **Parameters**
    - `c` - Command to execute.
  - **Returns** - Itself.

- *getEdges*
  `public Collection` **getEdges**`( )`
  - **Usage**
    - Gets the edges from within.
  - **Returns** - The edges.

- *getFreshID*
  `public void` **getFreshID**`( )`
  - **Usage**
    - Sets the id of the object to a new fresh id.

- *getID*
  `public int` **getID**`( )`
  - **Usage**
    - Gets the id of the object.
  - **Returns** - The id.

- *getNodes*
  `public Collection` **getNodes**`( )`
  - **Usage**
    - Gets the nodes from within.
  - **Returns** - The nodes.

- *merge*
  `public Graph` **merge**`( uk.ac.ic.doc.neuralnets.graph.Graph` **o** `)`
  - **Usage**
    - Merges one graph with its self, as all the edges and nodes.
  - **Parameters**
    - `o` - Graph to merge with.
  - **Returns** - Itself

- *setID*
  `public void` **setID**`( int` **id** `)`
  - **Usage**
    - Sets the id of the object to parameter.
  - **Parameters**
    - `int` - New id.

- *toString*
  `public String` **toString**`( )`

- *type*
  `protected String` **type**`( )`
  - **Usage**
    - Returns the object type.
  - **Returns** - Object type.

Chapter 9

Package uk.ac.ic.doc.neuralnets.graph.neural.train

9.1 Interfaces

9.1.1 Interface **Trainer**

Declaration

public interface Trainer
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin

Methods

- *setInputs*
  `public void` **setInputs**`( java.util.Collection` **in** `)`

- *setInputs*
  `public void` **setInputs**`( uk.ac.ic.doc.neuralnets.graph.neural.io.InputNode` **in** `)`

- *setTestLength*
  `public void` **setTestLength**`( int` **it** `)`

- *trainFully*
  `public double` **trainFully**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n**`, double` **errorTarget**`, int` **maxIt** `)`
  - **Usage**
    - Train this network until the accuracy >= target
  - **Parameters**
    - `n` - The network to train
    - `errorTarget` - The target accuracy
    - `maxIt` - The maximum number of iterations
  - **Returns** - The accuracy of the network after training

- *trainOnce*
  `public double` **trainOnce**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    - Train this network with one iteration
  - **Parameters**
    - `n` - The network to train
  - **Returns** - The accuracy of the network after training

Chapter 10

Package uk.ac.ic.doc.neuralnets.gui.connector

Package Contents *Page*

10.1 Classes

10.1.1 Class **NetworkConnector**

Declaration

public abstract class NetworkConnector
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin

Constructors

- *NetworkConnector*
  `public` **NetworkConnector**`( )`
- *NetworkConnector*
  `public` **NetworkConnector**`(`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm** `)`

Methods

- *connect*
  `public abstract Collection` **connect**`( java.util.List` **nodes** `)`
- *getConfigurationPanel*
  `public abstract Composite` **getConfigurationPanel**`(`
  `org.eclipse.swt.widgets.Composite` **parent** `)`
- *setGUIManager*
  `public void` **setGUIManager**`(`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **gm** `)`

Chapter 11

Package uk.ac.ic.doc.neuralnets.persistence

Package Contents *Page*

11.1 Interfaces

11.1.1 Interface **LoadSpecification**

LoadSpecifications provide an abstract method for parameterising a LoadService in order to load a neural network in to the program. To load a network a LoadSpecification is created which names the LoadService to use as the load process. The specification is passed to the LoadManager which retrieves the requested LoadService and passes the specification on to it.

Declaration

```
public interface LoadSpecification
```

Methods

- *getServiceName*
  `public String` **getServiceName**`( )`
  - **Usage**
    - The LoadService used by this specification.
  - **Returns** - the load service plugin name.

11.1.2 Interface **SaveSpecification**

SaveSpecification provide an abstract way of parameterising a SaveService in order to save a network. To save a network a SaveSpecification is created which names the SaveService to use as the save process. The specification is passed to the SaveManager which retrieves the requested SaveService and passes the specification on to it.

Declaration

```
public interface SaveSpecification
```

Methods

- *getServiceName*
  `public String` **getServiceName**`( )`
  - **Usage**
    - The SaveService used by this specification.
  - **Returns** - the save service plugin name.

11.2 Classes

11.2.1 Class **FileSpecification**

The FileSpecification provides parameters for persistence of networks to/from the file system, i.e. a file path.

Declaration

public class FileSpecification
extends java.lang.Object
implements SaveSpecification, LoadSpecification

Constructors

- *FileSpecification*
  `public` **FileSpecification**`( java.lang.String` **pathname**`, java.lang.String` **serviceName** `)`
  - **Usage**
    - Create a new specification.
  - **Parameters**
    - `pathname` - - path to save/load to from
    - `serviceName` - - the service to use.

Methods

- *getSavePath*
  `public String` **getSavePath**`( )`
  - **Usage**
    - Get the file system location.
  - **Returns** - the file path

- *getServiceName*
  `public String` **getServiceName**`( )`

- *setPath*
  `public void` **setPath**`( java.lang.String` **savePath** `)`
  - **Usage**
    - Set the file system location
  - **Parameters**
    - `savePath` - the new file path

11.2.2 Class **LoadException**

Denotes an error whilst attempting to load a network.

Declaration

public class LoadException
extends java.lang.Exception

Constructors

- *LoadException*
  `public` **LoadException**`( )`
- *LoadException*
  `public` **LoadException**`( java.lang.String` **message** `)`
- *LoadException*
  `public` **LoadException**`( java.lang.String` **message**`, java.lang.Throwable` **cause** `)`
- *LoadException*
  `public` **LoadException**`( java.lang.Throwable` **cause** `)`

Methods inherited from class `java.lang.Exception`

Methods inherited from class `java.lang.Throwable`

- *fillInStackTrace*
  `public synchronized native Throwable` **fillInStackTrace**`( )`
- *getCause*
  `public Throwable` **getCause**`( )`
- *getLocalizedMessage*
  `public String` **getLocalizedMessage**`( )`
- *getMessage*
  `public String` **getMessage**`( )`
- *getStackTrace*
  `public StackTraceElement` **getStackTrace**`( )`
- *initCause*
  `public synchronized Throwable` **initCause**`( java.lang.Throwable` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( )`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintStream` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintWriter` **arg0** `)`
- *setStackTrace*
  `public void` **setStackTrace**`( java.lang.StackTraceElement []` **arg0** `)`
- *toString*
  `public String` **toString**`( )`

11.2.3 Class **LoadManager**

The LoadManager is responsible for creating networks for use in the application from data in persistable storage using pluggable LoadServices, which are parameterised by LoadSpecifications.

Declaration

public class LoadManager
extends java.lang.Object

Methods

- *get*
  `public static LoadManager` **`get`**`( )`
  - **Usage**
    - Retrieve the instance of the LoadManager.
  - **Returns** - the LoadManager instance.
- *load*
  `public Saveable` **`load`**`( uk.ac.ic.doc.neuralnets.persistence.LoadSpecification` **`spec`** `)`
  - **Usage**
    - Reads in a external object using a load service parameterised by a load specification.
  - **Parameters**
    - `spec` - paramaters for loading
  - **Returns** - the loaded Saveable object.
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.persistence.LoadException` -

11.2.4 Class **LoadService**

Classes that implement this interface should be able to create neural networks for use in the application from data in persistable storage. They can be fully parameterised through the use of a LoadSpecification.

Declaration

public abstract class LoadService
extends uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin

Constructors

- *LoadService*
  `public` **`LoadService`**`( )`

METHODS

- *getFileType*
  `public abstract String` **getFileType**`( )`
  - **Usage**
    - Get the string form of the file type that this load service should seek e.g. "*.xml"
  - **Returns** - The lexical form of the file extension

- *load*
  `public abstract Saveable` **load**`(`
  `uk.ac.ic.doc.neuralnets.persistence.LoadSpecification` **spec** `)`
  - **Usage**
    - Imports a neural network from persistent storage.
  - **Parameters**
    - `spec` - - the load service parameters
  - **Returns** - the loaded network
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.persistence.LoadException` - in event of error during loading.

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin`

(in 7.2.4, page 62)

- *compareTo*
  `public int` **compareTo**`( uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin` **o** `)`
- *getPriority*
  `public abstract int` **getPriority**`( )`
  - **Usage**
    - The plugin's priority.
  - **Returns** - the priority

11.2.5 CLASS MethodSelector

DECLARATION

public class MethodSelector
extends java.lang.Object

CONSTRUCTORS

- *MethodSelector*
  `public` **MethodSelector**`( )`

Methods

- *getPersistableFields*
  `public Set` **getPersistableFields**(`java.lang.Class` **c**)
- *getPersistableMethods*
  `public Set` **getPersistableMethods**(`java.lang.Class` **c**)
- *getPersistableMethodsAndFields*
  `public Set` **getPersistableMethodsAndFields**(`java.lang.Class` **c**)

11.2.6 Class SaveException

Denotes there was an error whilst attempting to save a network.

Declaration

public class SaveException
extends java.lang.Exception

Constructors

- *SaveException*
  `public` **SaveException**( )
- *SaveException*
  `public` **SaveException**(`java.lang.String` **message**)
- *SaveException*
  `public` **SaveException**(`java.lang.String` **message**, `java.lang.Throwable` **cause**)
- *SaveException*
  `public` **SaveException**(`java.lang.Throwable` **cause**)

Methods inherited from class `java.lang.Exception`

Methods inherited from class `java.lang.Throwable`

- *fillInStackTrace*
  `public synchronized native Throwable` **fillInStackTrace**( )
- *getCause*
  `public Throwable` **getCause**( )
- *getLocalizedMessage*
  `public String` **getLocalizedMessage**( )

- *getMessage*
  `public String` **getMessage**`( )`
- *getStackTrace*
  `public StackTraceElement` **getStackTrace**`( )`
- *initCause*
  `public synchronized Throwable` **initCause**`( java.lang.Throwable` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( )`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintStream` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintWriter` **arg0** `)`
- *setStackTrace*
  `public void` **setStackTrace**`( java.lang.StackTraceElement []` **arg0** `)`
- *toString*
  `public String` **toString**`( )`

11.2.7 Class **SaveManager**

The SaveManager is responsible for persisting a given network via parameters specified in a SaveSpecification using pluggable SaveServices.

Declaration

public class SaveManager
extends java.lang.Object

Methods

- *get*
  `public static SaveManager` **get**`( )`
  - **Usage**
    - Retrieves the instance of the SaveManager.
  - **Returns** - the SaveManager instance.

- *save*
  `public void` **save**`( uk.ac.ic.doc.neuralnets.graph.Saveable` **net**`,`
  `uk.ac.ic.doc.neuralnets.persistence.SaveSpecification` **spec** `)`
  - **Usage**
    - Saves a network through the SaveService named in the SaveSpecification.
  - **Parameters**
    - `net` - the Neural Network to save.
    - `spec` - SaveSpecification, which contains parameters for the save service.
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.persistence.SaveException` - in the event something goes wrong during saving.

11.2.8 Class **SaveService**

Classes that implement this interface should be able to create a persistent representation of a given neural network in some format. They can be fully parameterised through the use of a SaveSpecification.

Declaration

public abstract class SaveService
extends uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin

Constructors

- *SaveService*
  `public` **SaveService**`( )`

Methods

- *getFileType*
  `public abstract String` **getFileType**`( )`
  - **Usage**
    - Get the string form of the file type that this save service should seek e.g. "*.xml"
  - **Returns** - The lexical form of the file extension

- *save*
  `public abstract void` **save**`( uk.ac.ic.doc.neuralnets.graph.Saveable` **network**`, uk.ac.ic.doc.neuralnets.persistence.SaveSpecification` **spec** `)`
  - **Usage**
    - Exports the given neural network to persistent storage in a given format
  - **Parameters**
    - `network` - - the network to save
    - `spec` - - the save service parameters
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.persistence.SaveException` - in the event of error during saving

Methods inherited from class `uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin`

(in 7.2.4, page 62)

- *compareTo*
  `public int` **compareTo**`( uk.ac.ic.doc.neuralnets.util.plugins.PriorityPlugin` **o** `)`
- *getPriority*
  `public abstract int` **getPriority**`( )`
  - **Usage**
    - The plugin's priority.
  - **Returns** - the priority

Chapter 12

Package uk.ac.ic.doc.neuralnets.matrix

12.1 Interfaces

12.1.1 Interface **Matrix.Command**

Declaration

public static interface Matrix.Command

Methods

- *exec*
  `public void exec( int x, int y, java.lang.Object item )`

12.2 Classes

12.2.1 Class **Matrix**

Matrix class that almost supports dynamic resizing May not be needed for our use cases, so didn't invest any more effort Resizing half-works (specify no-bound with width or height == 0), can put effort in if it's needed Wherever possible, instead of returning void from a public method, returns itself instead to permit chaining of calls

Declaration

public class Matrix
extends java.lang.Object
implements java.io.Serializable

Constructors

- *Matrix*
  `public Matrix( int width, int height )`

Methods

- *add*
  `public synchronized Matrix add( java.lang.Object item )`
- *add*
  `public synchronized Matrix add( java.lang.Object item, int x )`
- *bounds*
  `protected final void bounds( int x, int y )`

- *boundsX*
  `protected final void` **boundsX**`( int` **x** `)`
- *boundsY*
  `protected final void` **boundsY**`( int` **y** `)`
- *forEach*
  `public synchronized Matrix` **forEach**`(`
  `uk.ac.ic.doc.neuralnets.matrix.Matrix.Command` **c** `)`
- *get*
  `public synchronized Object` **get**`( int` **x**`, int` **y** `)`
- *getHeight*
  `public int` **getHeight**`( )`
- *getWidth*
  `public int` **getWidth**`( )`
- *set*
  `public synchronized Matrix` **set**`( java.lang.Object` **item**`, int` **x**`, int` **y** `)`
- *toString*
  `public synchronized String` **toString**`( )`

12.2.2 Class **PartitionableMatrix**

Declaration

public class PartitionableMatrix
extends uk.ac.ic.doc.neuralnets.matrix.Matrix

Serializable Fields

- private int pX1
  –
- private int pY1
  –
- private int pX2
  –
- private int pY2
  –

Constructors

- *PartitionableMatrix*
  `public` **PartitionableMatrix**`( int` **width**`, int` **height** `)`

Methods

- *clearPartition*
  `public synchronized PartitionableMatrix` **clearPartition**`( )`
- *forEachPartitioned*
  `public synchronized PartitionableMatrix` **forEachPartitioned**`( uk.ac.ic.doc.neuralnets.matrix.Matrix.Command` **c** `)`
- *getPartitioned*
  `public synchronized Object` **getPartitioned**`( int` **x**`, int` **y** `)`
- *getPartitionedMatrix*
  `public synchronized PartitionableMatrix` **getPartitionedMatrix**`( )`
- *newMatrix*
  `protected PartitionableMatrix` **newMatrix**`( int` **w**`, int` **h** `)`
- *partition*
  `public synchronized PartitionableMatrix` **partition**`( int` **x1**`, int` **y1**`, int` **x2**`, int` **y2** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.matrix.Matrix`

(in 12.2.1, page 93)

- *add*
  `public synchronized Matrix` **add**`( java.lang.Object` **item** `)`
- *add*
  `public synchronized Matrix` **add**`( java.lang.Object` **item**`, int` **x** `)`
- *bounds*
  `protected final void` **bounds**`( int` **x**`, int` **y** `)`
- *boundsX*
  `protected final void` **boundsX**`( int` **x** `)`
- *boundsY*
  `protected final void` **boundsY**`( int` **y** `)`
- *forEach*
  `public synchronized Matrix` **forEach**`( uk.ac.ic.doc.neuralnets.matrix.Matrix.Command` **c** `)`
- *get*
  `public synchronized Object` **get**`( int` **x**`, int` **y** `)`
- *getHeight*
  `public int` **getHeight**`( )`
- *getWidth*
  `public int` **getWidth**`( )`
- *set*
  `public synchronized Matrix` **set**`( java.lang.Object` **item**`, int` **x**`, int` **y** `)`
- *toString*
  `public synchronized String` **toString**`( )`

12.2.3 Class **RollUpMatrix**

Declaration

public class RollUpMatrix
extends uk.ac.ic.doc.neuralnets.matrix.PartitionableMatrix

Constructors

- *RollUpMatrix*
  `public` **RollUpMatrix**`( int` **width**`, int` **height** `)`

Methods

- *newMatrix*
  `protected PartitionableMatrix` **newMatrix**`( int` **w**`, int` **h** `)`
- *rollUp*
  `public synchronized RollUpMatrix` **rollUp**`( int` **width**`, int` **height** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.matrix.PartitionableMatrix`

(in 12.2.2, page 94)

- *clearPartition*
  `public synchronized PartitionableMatrix` **clearPartition**`( )`
- *forEachPartitioned*
  `public synchronized PartitionableMatrix` **forEachPartitioned**`( uk.ac.ic.doc.neuralnets.matrix.Matrix.Command` **c** `)`
- *getPartitioned*
  `public synchronized Object` **getPartitioned**`( int` **x**`, int` **y** `)`
- *getPartitionedMatrix*
  `public synchronized PartitionableMatrix` **getPartitionedMatrix**`( )`
- *newMatrix*
  `protected PartitionableMatrix` **newMatrix**`( int` **w**`, int` **h** `)`
- *partition*
  `public synchronized PartitionableMatrix` **partition**`( int` **x1**`, int` **y1**`, int` **x2**`, int` **y2** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.matrix.Matrix`

(in 12.2.1, page 93)

- *add*
  `public synchronized Matrix` **add**`( java.lang.Object` **item** `)`
- *add*
  `public synchronized Matrix` **add**`( java.lang.Object` **item**`, int` **x** `)`
- *bounds*
  `protected final void` **bounds**`( int` **x**`, int` **y** `)`
- *boundsX*
  `protected final void` **boundsX**`( int` **x** `)`

- *boundsY*
  `protected final void` **boundsY**`( int` **y** `)`
- *forEach*
  `public synchronized Matrix` **forEach**`( uk.ac.ic.doc.neuralnets.matrix.Matrix.Command` **c** `)`
- *get*
  `public synchronized Object` **get**`( int` **x**`, int` **y** `)`
- *getHeight*
  `public int` **getHeight**`( )`
- *getWidth*
  `public int` **getWidth**`( )`
- *set*
  `public synchronized Matrix` **set**`( java.lang.Object` **item**`, int` **x**`, int` **y** `)`
- *toString*
  `public synchronized String` **toString**`( )`

Chapter 13

Package uk.ac.ic.doc.neuralnets.expressions

Package Contents *Page*

Interfaces

Classes

13.1 Interfaces

13.1.1 Interface **BindVariable**

Declaration

public interface BindVariable
implements java.lang.annotation.Annotation

Methods

- *rebind*
  `public boolean` **`rebind`**`( )`
  - **Usage**
    - Whether or not an Expression should rebind this method each time it is evaluated. Defaults to false.
- *value*
  `public String` **`value`**`( )`
  - **Usage**
    - The variable name to bind the annotated method to

13.2 Classes

13.2.1 Class **CalculationLexer**

Declaration

public class CalculationLexer
extends org.antlr.runtime.Lexer

Fields

- public static final int MOD
  –
- public static final int GRAND
  –
- public static final int INT
  –

- public static final int COSH
  –
- public static final int MULT
  –
- public static final int MINUS
  –
- public static final int EOF
  –
- public static final int SINH
  –
- public static final int LPAREN
  –
- public static final int RPAREN
  –
- public static final int TANH
  –
- public static final int WS
  –
- public static final int POW
  –
- public static final int NEWLINE
  –
- public static final int SIN
  –
- public static final int COS
  –
- public static final int TAN
  –
- public static final int RAND
  –
- public static final int DOUBLE
  –
- public static final int PLUS
  –

- public static final int VAR
  -
- public static final int DIV
  -

Constructors

- *CalculationLexer*
  `public` **CalculationLexer**`( )`
- *CalculationLexer*
  `public` **CalculationLexer**`( org.antlr.runtime.CharStream` **input** `)`
- *CalculationLexer*
  `public` **CalculationLexer**`( org.antlr.runtime.CharStream` **input**`,`
  `org.antlr.runtime.RecognizerSharedState` **state** `)`

Methods

- *getGrammarFileName*
  `public String` **getGrammarFileName**`( )`
- *mCOS*
  `public final void` **mCOS**`( )`
- *mCOSH*
  `public final void` **mCOSH**`( )`
- *mDIV*
  `public final void` **mDIV**`( )`
- *mDOUBLE*
  `public final void` **mDOUBLE**`( )`
- *mGRAND*
  `public final void` **mGRAND**`( )`
- *mINT*
  `public final void` **mINT**`( )`
- *mLPAREN*
  `public final void` **mLPAREN**`( )`
- *mMINUS*
  `public final void` **mMINUS**`( )`
- *mMOD*
  `public final void` **mMOD**`( )`
- *mMULT*
  `public final void` **mMULT**`( )`

- *mNEWLINE*
  `public final void` **mNEWLINE**`( )`
- *mPLUS*
  `public final void` **mPLUS**`( )`
- *mPOW*
  `public final void` **mPOW**`( )`
- *mRAND*
  `public final void` **mRAND**`( )`
- *mRPAREN*
  `public final void` **mRPAREN**`( )`
- *mSIN*
  `public final void` **mSIN**`( )`
- *mSINH*
  `public final void` **mSINH**`( )`
- *mTAN*
  `public final void` **mTAN**`( )`
- *mTANH*
  `public final void` **mTANH**`( )`
- *mTokens*
  `public void` **mTokens**`( )`
- *mVAR*
  `public final void` **mVAR**`( )`
- *mWS*
  `public final void` **mWS**`( )`

Methods inherited from class `org.antlr.runtime.Lexer`

- *emit*
  `public Token` **emit**`( )`
- *emit*
  `public void` **emit**`( org.antlr.runtime.Token` **arg0** `)`
- *getCharErrorDisplay*
  `public String` **getCharErrorDisplay**`( int` **arg0** `)`
- *getCharIndex*
  `public int` **getCharIndex**`( )`
- *getCharPositionInLine*
  `public int` **getCharPositionInLine**`( )`
- *getCharStream*
  `public CharStream` **getCharStream**`( )`
- *getErrorMessage*
  `public String` **getErrorMessage**`( org.antlr.runtime.RecognitionException` **arg0**, `java.lang.String [] ` **arg1** `)`
- *getLine*
  `public int` **getLine**`( )`

- *getSourceName*
  `public String` **getSourceName**`( )`
- *getText*
  `public String` **getText**`( )`
- *match*
  `public void` **match**`( int` **arg0** `)`
- *match*
  `public void` **match**`( java.lang.String` **arg0** `)`
- *matchAny*
  `public void` **matchAny**`( )`
- *matchRange*
  `public void` **matchRange**`( int` **arg0**`, int` **arg1** `)`
- *mTokens*
  `public abstract void` **mTokens**`( )`
- *nextToken*
  `public Token` **nextToken**`( )`
- *recover*
  `public void` **recover**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *reportError*
  `public void` **reportError**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *reset*
  `public void` **reset**`( )`
- *setCharStream*
  `public void` **setCharStream**`( org.antlr.runtime.CharStream` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`
- *skip*
  `public void` **skip**`( )`
- *traceIn*
  `public void` **traceIn**`( java.lang.String` **arg0**`, int` **arg1** `)`
- *traceOut*
  `public void` **traceOut**`( java.lang.String` **arg0**`, int` **arg1** `)`

Methods inherited from class `org.antlr.runtime.BaseRecognizer`

- *alreadyParsedRule*
  `public boolean` **alreadyParsedRule**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1** `)`
- *beginResync*
  `public void` **beginResync**`( )`
- *combineFollows*
  `protected BitSet` **combineFollows**`( boolean` **arg0** `)`
- *computeContextSensitiveRuleFOLLOW*
  `protected BitSet` **computeContextSensitiveRuleFOLLOW**`( )`
- *computeErrorRecoverySet*
  `protected BitSet` **computeErrorRecoverySet**`( )`
- *consumeUntil*
  `public void` **consumeUntil**`( org.antlr.runtime.IntStream` **arg0**`,`
  `org.antlr.runtime.BitSet` **arg1** `)`
- *consumeUntil*
  `public void` **consumeUntil**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1** `)`

- *displayRecognitionError*
  `public void` **displayRecognitionError**`( java.lang.String [] ` **arg0**`, org.antlr.runtime.RecognitionException ` **arg1** `)`
- *emitErrorMessage*
  `public void` **emitErrorMessage**`( java.lang.String ` **arg0** `)`
- *endResync*
  `public void` **endResync**`( )`
- *getBacktrackingLevel*
  `public int` **getBacktrackingLevel**`( )`
- *getCurrentInputSymbol*
  `protected Object` **getCurrentInputSymbol**`( org.antlr.runtime.IntStream ` **arg0** `)`
- *getErrorHeader*
  `public String` **getErrorHeader**`( org.antlr.runtime.RecognitionException ` **arg0** `)`
- *getErrorMessage*
  `public String` **getErrorMessage**`( org.antlr.runtime.RecognitionException ` **arg0**`, java.lang.String [] ` **arg1** `)`
- *getGrammarFileName*
  `public String` **getGrammarFileName**`( )`
- *getMissingSymbol*
  `protected Object` **getMissingSymbol**`( org.antlr.runtime.IntStream ` **arg0**`, org.antlr.runtime.RecognitionException ` **arg1**`, int ` **arg2**`, org.antlr.runtime.BitSet ` **arg3** `)`
- *getNumberOfSyntaxErrors*
  `public int` **getNumberOfSyntaxErrors**`( )`
- *getRuleInvocationStack*
  `public List` **getRuleInvocationStack**`( )`
- *getRuleInvocationStack*
  `public static List` **getRuleInvocationStack**`( java.lang.Throwable ` **arg0**`, java.lang.String ` **arg1** `)`
- *getRuleMemoization*
  `public int` **getRuleMemoization**`( int ` **arg0**`, int ` **arg1** `)`
- *getRuleMemoizationCacheSize*
  `public int` **getRuleMemoizationCacheSize**`( )`
- *getSourceName*
  `public abstract String` **getSourceName**`( )`
- *getTokenErrorDisplay*
  `public String` **getTokenErrorDisplay**`( org.antlr.runtime.Token ` **arg0** `)`
- *getTokenNames*
  `public String` **getTokenNames**`( )`
- *match*
  `public Object` **match**`( org.antlr.runtime.IntStream ` **arg0**`, int ` **arg1**`, org.antlr.runtime.BitSet ` **arg2** `)`
- *matchAny*
  `public void` **matchAny**`( org.antlr.runtime.IntStream ` **arg0** `)`
- *memoize*
  `public void` **memoize**`( org.antlr.runtime.IntStream ` **arg0**`, int ` **arg1**`, int ` **arg2** `)`
- *mismatch*
  `protected void` **mismatch**`( org.antlr.runtime.IntStream ` **arg0**`, int ` **arg1**`, org.antlr.runtime.BitSet ` **arg2** `)`
- *mismatchIsMissingToken*
  `public boolean` **mismatchIsMissingToken**`( org.antlr.runtime.IntStream ` **arg0**`, org.antlr.runtime.BitSet ` **arg1** `)`

- *mismatchIsUnwantedToken*
  `public boolean` **mismatchIsUnwantedToken**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1** `)`
- *pushFollow*
  `protected void` **pushFollow**`( org.antlr.runtime.BitSet` **arg0** `)`
- *recover*
  `public void` **recover**`( org.antlr.runtime.IntStream` **arg0**`, org.antlr.runtime.RecognitionException` **arg1** `)`
- *recoverFromMismatchedSet*
  `public Object` **recoverFromMismatchedSet**`( org.antlr.runtime.IntStream` **arg0**`, org.antlr.runtime.RecognitionException` **arg1**`, org.antlr.runtime.BitSet` **arg2** `)`
- *recoverFromMismatchedToken*
  `protected Object` **recoverFromMismatchedToken**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1**`, org.antlr.runtime.BitSet` **arg2** `)`
- *reportError*
  `public void` **reportError**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *reset*
  `public void` **reset**`( )`
- *toStrings*
  `public List` **toStrings**`( java.util.List` **arg0** `)`
- *traceIn*
  `public void` **traceIn**`( java.lang.String` **arg0**`, int` **arg1**`, java.lang.Object` **arg2** `)`
- *traceOut*
  `public void` **traceOut**`( java.lang.String` **arg0**`, int` **arg1**`, java.lang.Object` **arg2** `)`

13.2.2 Class **CalculationParser**

Declaration

public class CalculationParser
extends org.antlr.runtime.Parser

Fields

- public static final String tokenNames
  –
- public static final int MOD
  –
- public static final int INT
  –
- public static final int GRAND
  –
- public static final int COSH

–

- public static final int MULT

  –

- public static final int MINUS

  –

- public static final int EOF

  –

- public static final int SINH

  –

- public static final int LPAREN

  –

- public static final int RPAREN

  –

- public static final int TANH

  –

- public static final int WS

  –

- public static final int POW

  –

- public static final int NEWLINE

  –

- public static final int SIN

  –

- public static final int COS

  –

- public static final int RAND

  –

- public static final int TAN

  –

- public static final int DOUBLE

  –

- public static final int PLUS

  –

- public static final int VAR

  –

- public static final int DIV
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_stat191
  –
- public static final BitSet FOLLOW_NEWLINE_in_stat193
  –
- public static final BitSet FOLLOW_multLevelExpr_in_lowLevelExpr220
  –
- public static final BitSet FOLLOW_PLUS_in_lowLevelExpr234
  –
- public static final BitSet FOLLOW_multLevelExpr_in_lowLevelExpr238
  –
- public static final BitSet FOLLOW_MINUS_in_lowLevelExpr252
  –
- public static final BitSet FOLLOW_multLevelExpr_in_lowLevelExpr256
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr294
  –
- public static final BitSet FOLLOW_MULT_in_multLevelExpr314
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr318
  –
- public static final BitSet FOLLOW_DIV_in_multLevelExpr329
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr333
  –
- public static final BitSet FOLLOW_MOD_in_multLevelExpr344
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr348
  –
- public static final BitSet FOLLOW_unary_in_powLevelExpr378
  –
- public static final BitSet FOLLOW_POW_in_powLevelExpr386

  –

- public static final BitSet FOLLOW_unary_in_powLevelExpr390
  –
- public static final BitSet FOLLOW_atom_in_unary414
  –
- public static final BitSet FOLLOW_MINUS_in_unary421
  –
- public static final BitSet FOLLOW_atom_in_unary425
  –
- public static final BitSet FOLLOW_INT_in_atom446
  –
- public static final BitSet FOLLOW_VAR_in_atom453
  –
- public static final BitSet FOLLOW_DOUBLE_in_atom460
  –
- public static final BitSet FOLLOW_RAND_in_atom468
  –
- public static final BitSet FOLLOW_GRAND_in_atom476
  –
- public static final BitSet FOLLOW_LPAREN_in_atom486
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom488
  –
- public static final BitSet FOLLOW_RPAREN_in_atom490
  –
- public static final BitSet FOLLOW_SINH_in_atom497
  –
- public static final BitSet FOLLOW_LPAREN_in_atom499
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom503
  –
- public static final BitSet FOLLOW_RPAREN_in_atom506
  –
- public static final BitSet FOLLOW_COSH_in_atom511

–

- public static final BitSet FOLLOW_LPAREN_in_atom513
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom517
  –
- public static final BitSet FOLLOW_RPAREN_in_atom520
  –
- public static final BitSet FOLLOW_TANH_in_atom525
  –
- public static final BitSet FOLLOW_LPAREN_in_atom527
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom531
  –
- public static final BitSet FOLLOW_RPAREN_in_atom534
  –
- public static final BitSet FOLLOW_SIN_in_atom539
  –
- public static final BitSet FOLLOW_LPAREN_in_atom541
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom545
  –
- public static final BitSet FOLLOW_RPAREN_in_atom548
  –
- public static final BitSet FOLLOW_COS_in_atom553
  –
- public static final BitSet FOLLOW_LPAREN_in_atom555
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom559
  –
- public static final BitSet FOLLOW_RPAREN_in_atom562
  –
- public static final BitSet FOLLOW_TAN_in_atom567
  –
- public static final BitSet FOLLOW_LPAREN_in_atom569

–

- public static final BitSet FOLLOW_lowLevelExpr_in_atom573

  –

- public static final BitSet FOLLOW_RPAREN_in_atom576

  –

Constructors

- *CalculationParser*
  `public` **CalculationParser**`( org.antlr.runtime.TokenStream` **input** `)`
- *CalculationParser*
  `public` **CalculationParser**`( org.antlr.runtime.TokenStream` **input**`,`
  `org.antlr.runtime.RecognizerSharedState` **state** `)`

Methods

- *atom*
  `public final Double` **atom**`( )`
- *bind*
  `public void` **bind**`( java.lang.String` **var**`, java.lang.Double` **val** `)`
- *displayRecognitionError*
  `public void` **displayRecognitionError**`( java.lang.String []` **tokenNames**`,`
  `org.antlr.runtime.RecognitionException` **e** `)`
- *evaluate*
  `public Double` **evaluate**`( )`
- *getGrammarFileName*
  `public String` **getGrammarFileName**`( )`
- *getTokenNames*
  `public String` **getTokenNames**`( )`
- *lowLevelExpr*
  `public final Double` **lowLevelExpr**`( )`
- *multLevelExpr*
  `public final Double` **multLevelExpr**`( )`
- *powLevelExpr*
  `public final Double` **powLevelExpr**`( )`
- *stat*
  `public final Double` **stat**`( )`
- *unary*
  `public final Double` **unary**`( )`

Methods inherited from class `org.antlr.runtime.Parser`

- *getCurrentInputSymbol*
  `protected Object` **getCurrentInputSymbol**(`org.antlr.runtime.IntStream` **arg0**)
- *getMissingSymbol*
  `protected Object` **getMissingSymbol**(`org.antlr.runtime.IntStream` **arg0**, `org.antlr.runtime.RecognitionException` **arg1**, `int` **arg2**, `org.antlr.runtime.BitSet` **arg3**)
- *getSourceName*
  `public String` **getSourceName**( )
- *getTokenStream*
  `public TokenStream` **getTokenStream**( )
- *reset*
  `public void` **reset**( )
- *setTokenStream*
  `public void` **setTokenStream**(`org.antlr.runtime.TokenStream` **arg0**)
- *traceIn*
  `public void` **traceIn**(`java.lang.String` **arg0**, `int` **arg1**)
- *traceOut*
  `public void` **traceOut**(`java.lang.String` **arg0**, `int` **arg1**)

Methods inherited from class `org.antlr.runtime.BaseRecognizer`

- *alreadyParsedRule*
  `public boolean` **alreadyParsedRule**(`org.antlr.runtime.IntStream` **arg0**, `int` **arg1**)
- *beginResync*
  `public void` **beginResync**( )
- *combineFollows*
  `protected BitSet` **combineFollows**(`boolean` **arg0**)
- *computeContextSensitiveRuleFOLLOW*
  `protected BitSet` **computeContextSensitiveRuleFOLLOW**( )
- *computeErrorRecoverySet*
  `protected BitSet` **computeErrorRecoverySet**( )
- *consumeUntil*
  `public void` **consumeUntil**(`org.antlr.runtime.IntStream` **arg0**, `org.antlr.runtime.BitSet` **arg1**)
- *consumeUntil*
  `public void` **consumeUntil**(`org.antlr.runtime.IntStream` **arg0**, `int` **arg1**)
- *displayRecognitionError*
  `public void` **displayRecognitionError**(`java.lang.String []` **arg0**, `org.antlr.runtime.RecognitionException` **arg1**)
- *emitErrorMessage*
  `public void` **emitErrorMessage**(`java.lang.String` **arg0**)
- *endResync*
  `public void` **endResync**( )
- *getBacktrackingLevel*
  `public int` **getBacktrackingLevel**( )
- *getCurrentInputSymbol*
  `protected Object` **getCurrentInputSymbol**(`org.antlr.runtime.IntStream` **arg0**)

- *getErrorHeader*
  public String **getErrorHeader**(org.antlr.runtime.RecognitionException **arg0**)
- *getErrorMessage*
  public String **getErrorMessage**(org.antlr.runtime.RecognitionException **arg0**, java.lang.String [] **arg1**)
- *getGrammarFileName*
  public String **getGrammarFileName**( )
- *getMissingSymbol*
  protected Object **getMissingSymbol**(org.antlr.runtime.IntStream **arg0**, org.antlr.runtime.RecognitionException **arg1**, int **arg2**, org.antlr.runtime.BitSet **arg3**)
- *getNumberOfSyntaxErrors*
  public int **getNumberOfSyntaxErrors**( )
- *getRuleInvocationStack*
  public List **getRuleInvocationStack**( )
- *getRuleInvocationStack*
  public static List **getRuleInvocationStack**(java.lang.Throwable **arg0**, java.lang.String **arg1**)
- *getRuleMemoization*
  public int **getRuleMemoization**(int **arg0**, int **arg1**)
- *getRuleMemoizationCacheSize*
  public int **getRuleMemoizationCacheSize**( )
- *getSourceName*
  public abstract String **getSourceName**( )
- *getTokenErrorDisplay*
  public String **getTokenErrorDisplay**(org.antlr.runtime.Token **arg0**)
- *getTokenNames*
  public String **getTokenNames**( )
- *match*
  public Object **match**(org.antlr.runtime.IntStream **arg0**, int **arg1**, org.antlr.runtime.BitSet **arg2**)
- *matchAny*
  public void **matchAny**(org.antlr.runtime.IntStream **arg0**)
- *memoize*
  public void **memoize**(org.antlr.runtime.IntStream **arg0**, int **arg1**, int **arg2**)
- *mismatch*
  protected void **mismatch**(org.antlr.runtime.IntStream **arg0**, int **arg1**, org.antlr.runtime.BitSet **arg2**)
- *mismatchIsMissingToken*
  public boolean **mismatchIsMissingToken**(org.antlr.runtime.IntStream **arg0**, org.antlr.runtime.BitSet **arg1**)
- *mismatchIsUnwantedToken*
  public boolean **mismatchIsUnwantedToken**(org.antlr.runtime.IntStream **arg0**, int **arg1**)
- *pushFollow*
  protected void **pushFollow**(org.antlr.runtime.BitSet **arg0**)
- *recover*
  public void **recover**(org.antlr.runtime.IntStream **arg0**, org.antlr.runtime.RecognitionException **arg1**)
- *recoverFromMismatchedSet*
  public Object **recoverFromMismatchedSet**(org.antlr.runtime.IntStream **arg0**, org.antlr.runtime.RecognitionException **arg1**, org.antlr.runtime.BitSet **arg2**)

- *recoverFromMismatchedToken*
  `protected Object` **recoverFromMismatchedToken**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1**`, org.antlr.runtime.BitSet` **arg2** `)`
- *reportError*
  `public void` **reportError**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *reset*
  `public void` **reset**`( )`
- *toStrings*
  `public List` **toStrings**`( java.util.List` **arg0** `)`
- *traceIn*
  `public void` **traceIn**`( java.lang.String` **arg0**`, int` **arg1**`, java.lang.Object` **arg2** `)`
- *traceOut*
  `public void` **traceOut**`( java.lang.String` **arg0**`, int` **arg1**`, java.lang.Object` **arg2** `)`

13.2.3 Class **Expression**

Declaration

public class Expression
extends java.lang.Object

Constructors

- *Expression*
  `public` **Expression**`( java.lang.Double` **value** `)`
  - **Usage**
    - Create an Expression to encode the given value
  - **Parameters**
    - `value` - The value returned by this Expression
- *Expression*
  `public` **Expression**`( java.lang.String` **expr** `)`
  - **Usage**
    - Create an Expression for the given string
  - **Parameters**
    - `expr` - The expression to represent

Methods

- *bind*
  `public void` **bind**`( java.lang.Object` **o** `)`
  - **Usage**
    - Bind variables according to BindVariable annotations present in this object, and all of its super-classes

- **Parameters**
  - `o` - The object to bind variables from

- *bind*
  `public void` **bind**`( java.lang.String` **var**`, java.lang.Double` **val** `)`
  - **Usage**
    - Manually bind a variable in the expression
  - **Parameters**
    - `var` - The variable to bind
    - `val` - The value to bind to

- *bind*
  `protected void` **bind**`( java.lang.String` **var**`, java.lang.reflect.Method` **m** `)`

- *evaluate*
  `public Double` **evaluate**`( )`
  - **Usage**
    - Evaluate the expression after refreshing its current bindings
  - **Returns** - The value this expression evaluates to
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.expressions.ExpressionException` -

- *evaluate*
  `public Double` **evaluate**`( java.lang.Object` **o** `)`
  - **Usage**
    - Re-bind variables, then evaluate the expression
  - **Parameters**
    - `o` - The object to bind variables from
  - **Returns** - The value this expression evaluates to
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.expressions.ExpressionException` -

- *getExpression*
  `public String` **getExpression**`( )`
  - **Usage**
    - Answer the input expression
  - **Returns** - The mathematical expression encoded by this object

---

- *getParser*
  `protected CalculationParser` **getParser**`( java.lang.String` **ex** `)`

- *toString*
  `public String` **toString**`( )`

13.2.4 Class **ExpressionException**

Declaration

public class ExpressionException
extends java.lang.Exception

Constructors

- *ExpressionException*
  `public` **ExpressionException**`( java.lang.Exception` **e** `)`
- *ExpressionException*
  `public` **ExpressionException**`( java.lang.String` **msg** `)`

Methods inherited from class `java.lang.Exception`

Methods inherited from class `java.lang.Throwable`

- *fillInStackTrace*
  `public synchronized native Throwable` **fillInStackTrace**`( )`
- *getCause*
  `public Throwable` **getCause**`( )`
- *getLocalizedMessage*
  `public String` **getLocalizedMessage**`( )`
- *getMessage*
  `public String` **getMessage**`( )`
- *getStackTrace*
  `public StackTraceElement` **getStackTrace**`( )`
- *initCause*
  `public synchronized Throwable` **initCause**`( java.lang.Throwable` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( )`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintStream` **arg0** `)`
- *printStackTrace*
  `public void` **printStackTrace**`( java.io.PrintWriter` **arg0** `)`
- *setStackTrace*
  `public void` **setStackTrace**`( java.lang.StackTraceElement []` **arg0** `)`
- *toString*
  `public String` **toString**`( )`

Chapter 14

Package uk.ac.ic.doc.neuralnets.commands

Package Contents *Page*

14.1 Classes

14.1.1 Class **Command**

Action that can be undone or redone.

Declaration

public abstract class Command
extends java.lang.Object
implements java.lang.Runnable

Constructors

- *Command*
  `public` **Command**`( )`

Methods

- *execute*
  `protected abstract void` **execute**`( )`

- *isUndo*
  `public boolean` **isUndo**`( )`
  - **Usage**
    * Returns the value of whether the command is set to undo.
  - **Returns** - Boolean commands undo state.

- *run*
  `public void` **run**`( )`
  - **Usage**
    * Runs the command, undone is undo state is true, else command executed.

- *setUndo*
  `public void` **setUndo**`( boolean` **undo** `)`
  - **Usage**
    * Sets the commands state of undo.
  - **Parameters**
    * `undo` - Boolean for undo state.

- *undo*
  `protected abstract void` **undo**`( )`

14.1.2 Class **CommandControl**

Implements undo and redo functionality. The addCommand() method adds a new stack and runs it, and the undo() and redo() methods can be called from the GUI.

Declaration

public class CommandControl
extends java.lang.Object

Constructors

- *CommandControl*
  `public` **CommandControl**`( )`

Methods

- *addCommand*
  `public void` **addCommand**`( uk.ac.ic.doc.neuralnets.commands.Command` **command** `)`
  - **Usage**
    - Executes a command and adds it to the stack so it can be undone and redone.
  - **Parameters**
    - `command` -
- *canRedo*
  `public boolean` **canRedo**`( )`
  - **Usage**
    - Returns boolean value of ability to redo.
  - **Returns** - Boolean of ability to redo.
- *canUndo*
  `public boolean` **canUndo**`( )`
  - **Usage**
    - Returns boolean value of ability to undo.
  - **Returns** - Boolean of ability to undo.
- *redo*
  `public void` **redo**`( )`
  - **Usage**
    - Redoes the last command that was undone.

- *reset*
  `public void` **reset**`( )`
- *stopDispatcher*
  `public void` **stopDispatcher**`( )`
- *undo*
  `public void` **undo**`( )`
  - **Usage**
    - Undoes the most recent command.

14.1.3 Class **CommandEvent**

Declaration

public class CommandEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *CommandEvent*
  `public` **CommandEvent**`( )`

Methods

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

Chapter 15

Package uk.ac.ic.doc.neuralnets.gui.graph.listener

15.1 Classes

15.1.1 Class **KeyboardPlugin**

Declaration

public abstract class KeyboardPlugin
extends java.lang.Object
implements org.eclipse.swt.events.KeyListener, uk.ac.ic.doc.neuralnets.util.plugins.Plugin

Constructors

- *KeyboardPlugin*
  `public KeyboardPlugin( )`

Methods

- *getName*
  `public abstract String getName( )`
- *keyPressed*
  `public void keyPressed( org.eclipse.swt.events.KeyEvent e )`
- *keyReleased*
  `public void keyReleased( org.eclipse.swt.events.KeyEvent e )`
- *setManager*
  `public void setManager( uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager g )`

15.1.2 Class **MouseItemListener**

Declaration

public class MouseItemListener
extends java.lang.Object
implements org.eclipse.swt.events.MouseListener

Constructors

- *MouseItemListener*
  `public MouseItemListener( )`
- *MouseItemListener*
  `public MouseItemListener( org.eclipse.zest.core.widgets.Graph g )`

Methods

- *getFigureAt*
  `protected IFigure` **getFigureAt**`( int` **x**`, int` **y** `)`

- *getGraph*
  `public Graph` **getGraph**`( )`

- *getItemAt*
  `protected GraphItem` **getItemAt**`( int` **x**`, int` **y** `)`

- *getItemFor*
  `protected GraphItem` **getItemFor**`( org.eclipse.draw2d.IFigure` **figure** `)`
  - **Usage**
    - This could be hideously slow, in theory. We're iterating over all the nodes, then all the edges. However, experimentally it is faster than the GUI update for a given size of network.
      We could store this data in a Map<IFigure,GraphItem>, but then there's a lot of housekeeping involved in keeping the map up to date - plus we end up with a big chunk of memory storing all the pointers again

- *handleClick*
  `protected void` **handleClick**`( org.eclipse.swt.events.MouseEvent` **e**`, org.eclipse.zest.core.widgets.GraphItem` **i** `)`

- *handleDoubleClick*
  `protected void` **handleDoubleClick**`( org.eclipse.swt.events.MouseEvent` **e**`, org.eclipse.zest.core.widgets.GraphItem` **i** `)`

- *handleDown*
  `protected void` **handleDown**`( org.eclipse.swt.events.MouseEvent` **e**`, org.eclipse.zest.core.widgets.GraphItem` **i** `)`

- *handleUp*
  `protected void` **handleUp**`( org.eclipse.swt.events.MouseEvent` **e**`, org.eclipse.zest.core.widgets.GraphItem` **i** `)`

- *mouseDoubleClick*
  `public void` **mouseDoubleClick**`( org.eclipse.swt.events.MouseEvent` **e** `)`

- *mouseDown*
  `public void` **mouseDown**`( org.eclipse.swt.events.MouseEvent` **e** `)`

- *mouseUp*
  `public void` **mouseUp**`( org.eclipse.swt.events.MouseEvent` **e** `)`

- *setGraph*
  `public void` **setGraph**`( org.eclipse.zest.core.widgets.Graph` **g** `)`

15.1.3 Class **MousePlugin**

Declaration

public abstract class MousePlugin
extends uk.ac.ic.doc.neuralnets.gui.graph.listener.MouseItemListener
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin

Constructors

- *MousePlugin*
  `public` **MousePlugin**`( )`

Methods

- *getName*
  `public abstract String` **getName**`( )`
- *setManager*
  `public void` **setManager**`(`
  `uk.ac.ic.doc.neuralnets.coreui.ZoomingInterfaceManager` **g** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.gui.graph.listener.MouseItemListener`

(in 15.1.2, page 121)

- *getFigureAt*
  `protected IFigure` **getFigureAt**`( int` **x**`, int` **y** `)`
- *getGraph*
  `public Graph` **getGraph**`( )`
- *getItemAt*
  `protected GraphItem` **getItemAt**`( int` **x**`, int` **y** `)`
- *getItemFor*
  `protected GraphItem` **getItemFor**`( org.eclipse.draw2d.IFigure` **figure** `)`
  - **Usage**
    - This could be hideously slow, in theory. We're iterating over all the nodes, then all the edges. However, experimentally it is faster than the GUI update for a given size of network. We could store this data in a Map<IFigure,GraphItem>, but then there's a lot of housekeeping involved in keeping the map up to date - plus we end up with a big chunk of memory storing all the pointers again
- *handleClick*
  `protected void` **handleClick**`( org.eclipse.swt.events.MouseEvent` **e**`,`
  `org.eclipse.zest.core.widgets.GraphItem` **i** `)`
- *handleDoubleClick*
  `protected void` **handleDoubleClick**`( org.eclipse.swt.events.MouseEvent` **e**`,`
  `org.eclipse.zest.core.widgets.GraphItem` **i** `)`
- *handleDown*
  `protected void` **handleDown**`( org.eclipse.swt.events.MouseEvent` **e**`,`
  `org.eclipse.zest.core.widgets.GraphItem` **i** `)`

- *handleUp*
  `protected void` **handleUp**`( org.eclipse.swt.events.MouseEvent` **e**`, org.eclipse.zest.core.widgets.GraphItem` **i** `)`
- *mouseDoubleClick*
  `public void` **mouseDoubleClick**`( org.eclipse.swt.events.MouseEvent` **e** `)`
- *mouseDown*
  `public void` **mouseDown**`( org.eclipse.swt.events.MouseEvent` **e** `)`
- *mouseUp*
  `public void` **mouseUp**`( org.eclipse.swt.events.MouseEvent` **e** `)`
- *setGraph*
  `public void` **setGraph**`( org.eclipse.zest.core.widgets.Graph` **g** `)`

Chapter 16

Package uk.ac.ic.doc.neuralnets.expressions.ast

Package Contents *Page*

Classes

16.1 Classes

16.1.1 Class **ASTExpression**

Declaration

public class ASTExpression
extends java.lang.Object

Constructors

- *ASTExpression*
  `public` **ASTExpression**`( java.lang.Double` **value** `)`
  - **Usage**
    - Create an Expression to encode the given value
  - **Parameters**
    - `value` - The value returned by this Expression

- *ASTExpression*
  `public` **ASTExpression**`( java.lang.String` **expr** `)`
  - **Usage**
    - Create an Expression for the given string
  - **Parameters**
    - `expr` - The expression to represent

Methods

- *bind*
  `public void` **bind**`( java.lang.Object` **o** `)`
  - **Usage**
    - Bind variables according to BindVariable annotations present in this object, and all of its super-classes
  - **Parameters**
    - `o` - The object to bind variables from

- *bind*
  `public void` **bind**`( java.lang.String` **var**`, java.lang.Double` **val** `)`
  - **Usage**
    - Manually bind a variable in the expression
  - **Parameters**
    - `var` - The variable to bind
    - `val` - The value to bind to

- *bind*
  `protected void` **bind**`( java.lang.String` **var**`, java.lang.reflect.Method` **m**`, java.lang.Object` **o** `)`

- *evaluate*
  `public Double` **evaluate**`( )`
  - **Usage**
    - Evaluate the expression after refreshing its current bindings
  - **Returns** - The value this expression evaluates to
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.expressions.ExpressionException` -

- *evaluate*
  `public Double` **evaluate**`( java.lang.Object` **o** `)`
  - **Usage**
    - Re-bind variables, then evaluate the expression
  - **Parameters**
    - `o` - The object to bind variables from
  - **Returns** - The value this expression evaluates to
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.expressions.ExpressionException` -

- *evaluateThis*
  `public Double` **evaluateThis**`( java.lang.Object` **o** `)`
  - **Usage**
    - Evaluate the expression after refreshing its current bindings from the supplied object. Will not seek new annotations.
  - **Parameters**
    - `o` - The object to bind on to
  - **Returns** - The value this expression evaluates to
  - **Exceptions**
    - `uk.ac.ic.doc.neuralnets.expressions.ExpressionException` -

- *getExpression*
  `public String` **getExpression**`( )`
  - **Usage**
    - Answer the input expression
  - **Returns** - The mathematical expression encoded by this object

- *parse*
  `protected Component` **parse**`( java.lang.String` **ex** `)`

- *toString*
  `public String` **toString**`( )`

16.1.2 Class **ASTExpressionFactory**

Declaration

public class ASTExpressionFactory
extends java.lang.Object

Methods

- *flushCache*
  `public void` **flushCache**`( )`
- *get*
  `public static ASTExpressionFactory` **get**`( )`
- *getExpression*
  `public ASTExpression` **getExpression**`( java.lang.Double` **d** `)`
- *getExpression*
  `public ASTExpression` **getExpression**`( java.lang.String` **expressionString** `)`

16.1.3 Class **BinaryOperator**

Declaration

public abstract class BinaryOperator
extends uk.ac.ic.doc.neuralnets.expressions.ast.Component

Constructors

- *BinaryOperator*
  `public` **BinaryOperator**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **l**`, uk.ac.ic.doc.neuralnets.expressions.ast.Component` **r**`, java.lang.String` **operation** `)`

Methods

- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public String` **getExpression**`( )`
- *getOperation*
  `public String` **getOperation**`( )`
- *getVariables*
  `public Set` **getVariables**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.expressions.ast.Component`

(in 16.1.4, page 129)

- *bracket*
  `public String` **bracket**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c** `)`
- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public abstract String` **getExpression**`( )`
- *getVariables*
  `public abstract Set` **getVariables**`( )`
- *order*
  `public int` **order**`( java.lang.String` **op** `)`

16.1.4 Class **Component**

Declaration

public abstract class Component
extends java.lang.Object

Constructors

- *Component*
  `public` **Component**`( )`

Methods

- *bracket*
  `public String` **bracket**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c** `)`
- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public abstract String` **getExpression**`( )`
- *getVariables*
  `public abstract Set` **getVariables**`( )`
- *order*
  `public int` **order**`( java.lang.String` **op** `)`

16.1.5 Class **ExpressionASTLexer**

Declaration

public class ExpressionASTLexer
extends org.antlr.runtime.Lexer

Fields

- public static final int MOD
  –
- public static final int GRAND
  –
- public static final int INT
  –
- public static final int COSH
  –
- public static final int MULT
  –
- public static final int MINUS
  –
- public static final int SQRT
  –
- public static final int EOF
  –
- public static final int SINH
  –
- public static final int LPAREN
  –
- public static final int RPAREN
  –
- public static final int TANH
  –
- public static final int WS
  –
- public static final int POW

  –

- public static final int NEWLINE
  –
- public static final int SIN
  –
- public static final int COS
  –
- public static final int TAN
  –
- public static final int RAND
  –
- public static final int DOUBLE
  –
- public static final int PLUS
  –
- public static final int VAR
  –
- public static final int DIV
  –

Constructors

- *ExpressionASTLexer*
  `public` **ExpressionASTLexer**`( )`
- *ExpressionASTLexer*
  `public` **ExpressionASTLexer**`( org.antlr.runtime.CharStream` **input** `)`
- *ExpressionASTLexer*
  `public` **ExpressionASTLexer**`( org.antlr.runtime.CharStream` **input**`,`
  `org.antlr.runtime.RecognizerSharedState` **state** `)`

Methods

- *getGrammarFileName*
  `public String` **getGrammarFileName**`( )`
- *mCOS*
  `public final void` **mCOS**`( )`

- *mCOSH*
  `public final void` **mCOSH**`( )`
- *mDIV*
  `public final void` **mDIV**`( )`
- *mDOUBLE*
  `public final void` **mDOUBLE**`( )`
- *mGRAND*
  `public final void` **mGRAND**`( )`
- *mINT*
  `public final void` **mINT**`( )`
- *mLPAREN*
  `public final void` **mLPAREN**`( )`
- *mMINUS*
  `public final void` **mMINUS**`( )`
- *mMOD*
  `public final void` **mMOD**`( )`
- *mMULT*
  `public final void` **mMULT**`( )`
- *mNEWLINE*
  `public final void` **mNEWLINE**`( )`
- *mPLUS*
  `public final void` **mPLUS**`( )`
- *mPOW*
  `public final void` **mPOW**`( )`
- *mRAND*
  `public final void` **mRAND**`( )`
- *mRPAREN*
  `public final void` **mRPAREN**`( )`
- *mSIN*
  `public final void` **mSIN**`( )`
- *mSINH*
  `public final void` **mSINH**`( )`
- *mSQRT*
  `public final void` **mSQRT**`( )`
- *mTAN*
  `public final void` **mTAN**`( )`
- *mTANH*
  `public final void` **mTANH**`( )`
- *mTokens*
  `public void` **mTokens**`( )`

- *mVAR*
  `public final void` **mVAR**`( )`
- *mWS*
  `public final void` **mWS**`( )`

Methods inherited from class `org.antlr.runtime.Lexer`

- *emit*
  `public Token` **emit**`( )`
- *emit*
  `public void` **emit**`( org.antlr.runtime.Token` **arg0** `)`
- *getCharErrorDisplay*
  `public String` **getCharErrorDisplay**`( int` **arg0** `)`
- *getCharIndex*
  `public int` **getCharIndex**`( )`
- *getCharPositionInLine*
  `public int` **getCharPositionInLine**`( )`
- *getCharStream*
  `public CharStream` **getCharStream**`( )`
- *getErrorMessage*
  `public String` **getErrorMessage**`( org.antlr.runtime.RecognitionException` **arg0**`, java.lang.String [ ]` **arg1** `)`
- *getLine*
  `public int` **getLine**`( )`
- *getSourceName*
  `public String` **getSourceName**`( )`
- *getText*
  `public String` **getText**`( )`
- *match*
  `public void` **match**`( int` **arg0** `)`
- *match*
  `public void` **match**`( java.lang.String` **arg0** `)`
- *matchAny*
  `public void` **matchAny**`( )`
- *matchRange*
  `public void` **matchRange**`( int` **arg0**`, int` **arg1** `)`
- *mTokens*
  `public abstract void` **mTokens**`( )`
- *nextToken*
  `public Token` **nextToken**`( )`
- *recover*
  `public void` **recover**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *reportError*
  `public void` **reportError**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *reset*
  `public void` **reset**`( )`
- *setCharStream*
  `public void` **setCharStream**`( org.antlr.runtime.CharStream` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`

- *skip*
  `public void` **skip**`( )`
- *traceIn*
  `public void` **traceIn**`( java.lang.String` **arg0**`, int` **arg1** `)`
- *traceOut*
  `public void` **traceOut**`( java.lang.String` **arg0**`, int` **arg1** `)`

Methods inherited from class `org.antlr.runtime.BaseRecognizer`

- *alreadyParsedRule*
  `public boolean` **alreadyParsedRule**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1** `)`
- *beginResync*
  `public void` **beginResync**`( )`
- *combineFollows*
  `protected BitSet` **combineFollows**`( boolean` **arg0** `)`
- *computeContextSensitiveRuleFOLLOW*
  `protected BitSet` **computeContextSensitiveRuleFOLLOW**`( )`
- *computeErrorRecoverySet*
  `protected BitSet` **computeErrorRecoverySet**`( )`
- *consumeUntil*
  `public void` **consumeUntil**`( org.antlr.runtime.IntStream` **arg0**`,`
  `org.antlr.runtime.BitSet` **arg1** `)`
- *consumeUntil*
  `public void` **consumeUntil**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1** `)`
- *displayRecognitionError*
  `public void` **displayRecognitionError**`( java.lang.String []` **arg0**`,`
  `org.antlr.runtime.RecognitionException` **arg1** `)`
- *emitErrorMessage*
  `public void` **emitErrorMessage**`( java.lang.String` **arg0** `)`
- *endResync*
  `public void` **endResync**`( )`
- *getBacktrackingLevel*
  `public int` **getBacktrackingLevel**`( )`
- *getCurrentInputSymbol*
  `protected Object` **getCurrentInputSymbol**`( org.antlr.runtime.IntStream` **arg0** `)`
- *getErrorHeader*
  `public String` **getErrorHeader**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *getErrorMessage*
  `public String` **getErrorMessage**`( org.antlr.runtime.RecognitionException` **arg0**`,`
  `java.lang.String []` **arg1** `)`
- *getGrammarFileName*
  `public String` **getGrammarFileName**`( )`
- *getMissingSymbol*
  `protected Object` **getMissingSymbol**`( org.antlr.runtime.IntStream` **arg0**`,`
  `org.antlr.runtime.RecognitionException` **arg1**`, int` **arg2**`, org.antlr.runtime.BitSet`
  arg3 `)`
- *getNumberOfSyntaxErrors*
  `public int` **getNumberOfSyntaxErrors**`( )`
- *getRuleInvocationStack*
  `public List` **getRuleInvocationStack**`( )`

- *getRuleInvocationStack*
  public static List **getRuleInvocationStack**(java.lang.Throwable **arg0**, java.lang.String **arg1**)
- *getRuleMemoization*
  public int **getRuleMemoization**(int **arg0**, int **arg1**)
- *getRuleMemoizationCacheSize*
  public int **getRuleMemoizationCacheSize**( )
- *getSourceName*
  public abstract String **getSourceName**( )
- *getTokenErrorDisplay*
  public String **getTokenErrorDisplay**(org.antlr.runtime.Token **arg0**)
- *getTokenNames*
  public String **getTokenNames**( )
- *match*
  public Object **match**(org.antlr.runtime.IntStream **arg0**, int **arg1**, org.antlr.runtime.BitSet **arg2**)
- *matchAny*
  public void **matchAny**(org.antlr.runtime.IntStream **arg0**)
- *memoize*
  public void **memoize**(org.antlr.runtime.IntStream **arg0**, int **arg1**, int **arg2**)
- *mismatch*
  protected void **mismatch**(org.antlr.runtime.IntStream **arg0**, int **arg1**, org.antlr.runtime.BitSet **arg2**)
- *mismatchIsMissingToken*
  public boolean **mismatchIsMissingToken**(org.antlr.runtime.IntStream **arg0**, org.antlr.runtime.BitSet **arg1**)
- *mismatchIsUnwantedToken*
  public boolean **mismatchIsUnwantedToken**(org.antlr.runtime.IntStream **arg0**, int **arg1**)
- *pushFollow*
  protected void **pushFollow**(org.antlr.runtime.BitSet **arg0**)
- *recover*
  public void **recover**(org.antlr.runtime.IntStream **arg0**, org.antlr.runtime.RecognitionException **arg1**)
- *recoverFromMismatchedSet*
  public Object **recoverFromMismatchedSet**(org.antlr.runtime.IntStream **arg0**, org.antlr.runtime.RecognitionException **arg1**, org.antlr.runtime.BitSet **arg2**)
- *recoverFromMismatchedToken*
  protected Object **recoverFromMismatchedToken**(org.antlr.runtime.IntStream **arg0**, int **arg1**, org.antlr.runtime.BitSet **arg2**)
- *reportError*
  public void **reportError**(org.antlr.runtime.RecognitionException **arg0**)
- *reset*
  public void **reset**( )
- *toStrings*
  public List **toStrings**(java.util.List **arg0**)
- *traceIn*
  public void **traceIn**(java.lang.String **arg0**, int **arg1**, java.lang.Object **arg2**)
- *traceOut*
  public void **traceOut**(java.lang.String **arg0**, int **arg1**, java.lang.Object **arg2**)

16.1.6 Class **ExpressionASTParser**

Declaration

```
public class ExpressionASTParser
extends org.antlr.runtime.Parser
```

Fields

- public static final String tokenNames
  –
- public static final int MOD
  –
- public static final int INT
  –
- public static final int GRAND
  –
- public static final int COSH
  –
- public static final int MULT
  –
- public static final int MINUS
  –
- public static final int SQRT
  –
- public static final int EOF
  –
- public static final int SINH
  –
- public static final int LPAREN
  –
- public static final int RPAREN
  –
- public static final int TANH

  –

- public static final int WS
  –
- public static final int POW
  –
- public static final int NEWLINE
  –
- public static final int SIN
  –
- public static final int COS
  –
- public static final int RAND
  –
- public static final int TAN
  –
- public static final int DOUBLE
  –
- public static final int PLUS
  –
- public static final int VAR
  –
- public static final int DIV
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_getTree199
  –
- public static final BitSet FOLLOW_NEWLINE_in_getTree201
  –
- public static final BitSet FOLLOW_multLevelExpr_in_lowLevelExpr223
  –
- public static final BitSet FOLLOW_PLUS_in_lowLevelExpr238
  –
- public static final BitSet FOLLOW_multLevelExpr_in_lowLevelExpr242
  –
- public static final BitSet FOLLOW_MINUS_in_lowLevelExpr257

  –

- public static final BitSet FOLLOW_multLevelExpr_in_lowLevelExpr261
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr295
  –
- public static final BitSet FOLLOW_MULT_in_multLevelExpr307
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr311
  –
- public static final BitSet FOLLOW_DIV_in_multLevelExpr323
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr327
  –
- public static final BitSet FOLLOW_MOD_in_multLevelExpr339
  –
- public static final BitSet FOLLOW_powLevelExpr_in_multLevelExpr343
  –
- public static final BitSet FOLLOW_unary_in_powLevelExpr372
  –
- public static final BitSet FOLLOW_POW_in_powLevelExpr380
  –
- public static final BitSet FOLLOW_unary_in_powLevelExpr384
  –
- public static final BitSet FOLLOW_atom_in_unary408
  –
- public static final BitSet FOLLOW_MINUS_in_unary415
  –
- public static final BitSet FOLLOW_atom_in_unary419
  –
- public static final BitSet FOLLOW_INT_in_atom440
  –
- public static final BitSet FOLLOW_DOUBLE_in_atom447
  –
- public static final BitSet FOLLOW_VAR_in_atom454

  –

- public static final BitSet FOLLOW_LPAREN_in_atom464
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom466
  –
- public static final BitSet FOLLOW_RPAREN_in_atom468
  –
- public static final BitSet FOLLOW_SQRT_in_atom475
  –
- public static final BitSet FOLLOW_LPAREN_in_atom477
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom481
  –
- public static final BitSet FOLLOW_RPAREN_in_atom484
  –
- public static final BitSet FOLLOW_RAND_in_atom490
  –
- public static final BitSet FOLLOW_GRAND_in_atom498
  –
- public static final BitSet FOLLOW_SINH_in_atom505
  –
- public static final BitSet FOLLOW_LPAREN_in_atom507
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom511
  –
- public static final BitSet FOLLOW_RPAREN_in_atom514
  –
- public static final BitSet FOLLOW_COSH_in_atom519
  –
- public static final BitSet FOLLOW_LPAREN_in_atom521
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom525
  –
- public static final BitSet FOLLOW_RPAREN_in_atom528

  –

- public static final BitSet FOLLOW_TANH_in_atom533
  –
- public static final BitSet FOLLOW_LPAREN_in_atom535
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom539
  –
- public static final BitSet FOLLOW_RPAREN_in_atom542
  –
- public static final BitSet FOLLOW_SIN_in_atom547
  –
- public static final BitSet FOLLOW_LPAREN_in_atom549
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom553
  –
- public static final BitSet FOLLOW_RPAREN_in_atom556
  –
- public static final BitSet FOLLOW_COS_in_atom561
  –
- public static final BitSet FOLLOW_LPAREN_in_atom563
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom567
  –
- public static final BitSet FOLLOW_RPAREN_in_atom570
  –
- public static final BitSet FOLLOW_TAN_in_atom575
  –
- public static final BitSet FOLLOW_LPAREN_in_atom577
  –
- public static final BitSet FOLLOW_lowLevelExpr_in_atom581
  –
- public static final BitSet FOLLOW_RPAREN_in_atom584
  –

Constructors

- *ExpressionASTParser*
  `public` **ExpressionASTParser**`( org.antlr.runtime.TokenStream` **input** `)`
- *ExpressionASTParser*
  `public` **ExpressionASTParser**`( org.antlr.runtime.TokenStream` **input**`, org.antlr.runtime.RecognizerSharedState` **state** `)`

Methods

- *atom*
  `public final Component` **atom**`( )`
- *getGrammarFileName*
  `public String` **getGrammarFileName**`( )`
- *getTokenNames*
  `public String` **getTokenNames**`( )`
- *getTree*
  `public final Component` **getTree**`( )`
- *getVariables*
  `public Map` **getVariables**`( )`
- *lowLevelExpr*
  `public final Component` **lowLevelExpr**`( )`
- *multLevelExpr*
  `public final Component` **multLevelExpr**`( )`
- *powLevelExpr*
  `public final Component` **powLevelExpr**`( )`
- *unary*
  `public final Component` **unary**`( )`

Methods inherited from class `org.antlr.runtime.Parser`

- *getCurrentInputSymbol*
  `protected Object` **getCurrentInputSymbol**`( org.antlr.runtime.IntStream` **arg0** `)`
- *getMissingSymbol*
  `protected Object` **getMissingSymbol**`( org.antlr.runtime.IntStream` **arg0**`, org.antlr.runtime.RecognitionException` **arg1**`, int` **arg2**`, org.antlr.runtime.BitSet` **arg3** `)`
- *getSourceName*
  `public String` **getSourceName**`( )`
- *getTokenStream*
  `public TokenStream` **getTokenStream**`( )`
- *reset*
  `public void` **reset**`( )`

- *setTokenStream*
  `public void` **setTokenStream**`( org.antlr.runtime.TokenStream` **arg0** `)`
- *traceIn*
  `public void` **traceIn**`( java.lang.String` **arg0**`, int` **arg1** `)`
- *traceOut*
  `public void` **traceOut**`( java.lang.String` **arg0**`, int` **arg1** `)`

Methods inherited from class `org.antlr.runtime.BaseRecognizer`

- *alreadyParsedRule*
  `public boolean` **alreadyParsedRule**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1** `)`
- *beginResync*
  `public void` **beginResync**`( )`
- *combineFollows*
  `protected BitSet` **combineFollows**`( boolean` **arg0** `)`
- *computeContextSensitiveRuleFOLLOW*
  `protected BitSet` **computeContextSensitiveRuleFOLLOW**`( )`
- *computeErrorRecoverySet*
  `protected BitSet` **computeErrorRecoverySet**`( )`
- *consumeUntil*
  `public void` **consumeUntil**`( org.antlr.runtime.IntStream` **arg0**`, org.antlr.runtime.BitSet` **arg1** `)`
- *consumeUntil*
  `public void` **consumeUntil**`( org.antlr.runtime.IntStream` **arg0**`, int` **arg1** `)`
- *displayRecognitionError*
  `public void` **displayRecognitionError**`( java.lang.String []` **arg0**`, org.antlr.runtime.RecognitionException` **arg1** `)`
- *emitErrorMessage*
  `public void` **emitErrorMessage**`( java.lang.String` **arg0** `)`
- *endResync*
  `public void` **endResync**`( )`
- *getBacktrackingLevel*
  `public int` **getBacktrackingLevel**`( )`
- *getCurrentInputSymbol*
  `protected Object` **getCurrentInputSymbol**`( org.antlr.runtime.IntStream` **arg0** `)`
- *getErrorHeader*
  `public String` **getErrorHeader**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *getErrorMessage*
  `public String` **getErrorMessage**`( org.antlr.runtime.RecognitionException` **arg0**`, java.lang.String []` **arg1** `)`
- *getGrammarFileName*
  `public String` **getGrammarFileName**`( )`
- *getMissingSymbol*
  `protected Object` **getMissingSymbol**`( org.antlr.runtime.IntStream` **arg0**`, org.antlr.runtime.RecognitionException` **arg1**`, int` **arg2**`, org.antlr.runtime.BitSet` **arg3** `)`
- *getNumberOfSyntaxErrors*
  `public int` **getNumberOfSyntaxErrors**`( )`
- *getRuleInvocationStack*
  `public List` **getRuleInvocationStack**`( )`

- *getRuleInvocationStack*
  `public static List` **getRuleInvocationStack**`( java.lang.Throwable` **arg0**, `java.lang.String` **arg1** `)`
- *getRuleMemoization*
  `public int` **getRuleMemoization**`( int` **arg0**, `int` **arg1** `)`
- *getRuleMemoizationCacheSize*
  `public int` **getRuleMemoizationCacheSize**`( )`
- *getSourceName*
  `public abstract String` **getSourceName**`( )`
- *getTokenErrorDisplay*
  `public String` **getTokenErrorDisplay**`( org.antlr.runtime.Token` **arg0** `)`
- *getTokenNames*
  `public String` **getTokenNames**`( )`
- *match*
  `public Object` **match**`( org.antlr.runtime.IntStream` **arg0**, `int` **arg1**, `org.antlr.runtime.BitSet` **arg2** `)`
- *matchAny*
  `public void` **matchAny**`( org.antlr.runtime.IntStream` **arg0** `)`
- *memoize*
  `public void` **memoize**`( org.antlr.runtime.IntStream` **arg0**, `int` **arg1**, `int` **arg2** `)`
- *mismatch*
  `protected void` **mismatch**`( org.antlr.runtime.IntStream` **arg0**, `int` **arg1**, `org.antlr.runtime.BitSet` **arg2** `)`
- *mismatchIsMissingToken*
  `public boolean` **mismatchIsMissingToken**`( org.antlr.runtime.IntStream` **arg0**, `org.antlr.runtime.BitSet` **arg1** `)`
- *mismatchIsUnwantedToken*
  `public boolean` **mismatchIsUnwantedToken**`( org.antlr.runtime.IntStream` **arg0**, `int` **arg1** `)`
- *pushFollow*
  `protected void` **pushFollow**`( org.antlr.runtime.BitSet` **arg0** `)`
- *recover*
  `public void` **recover**`( org.antlr.runtime.IntStream` **arg0**, `org.antlr.runtime.RecognitionException` **arg1** `)`
- *recoverFromMismatchedSet*
  `public Object` **recoverFromMismatchedSet**`( org.antlr.runtime.IntStream` **arg0**, `org.antlr.runtime.RecognitionException` **arg1**, `org.antlr.runtime.BitSet` **arg2** `)`
- *recoverFromMismatchedToken*
  `protected Object` **recoverFromMismatchedToken**`( org.antlr.runtime.IntStream` **arg0**, `int` **arg1**, `org.antlr.runtime.BitSet` **arg2** `)`
- *reportError*
  `public void` **reportError**`( org.antlr.runtime.RecognitionException` **arg0** `)`
- *reset*
  `public void` **reset**`( )`
- *toStrings*
  `public List` **toStrings**`( java.util.List` **arg0** `)`
- *traceIn*
  `public void` **traceIn**`( java.lang.String` **arg0**, `int` **arg1**, `java.lang.Object` **arg2** `)`
- *traceOut*
  `public void` **traceOut**`( java.lang.String` **arg0**, `int` **arg1**, `java.lang.Object` **arg2** `)`

16.1.7 Class **Literal**

Declaration

public class Literal
extends uk.ac.ic.doc.neuralnets.expressions.ast.Component

Constructors

- *Literal*
  `public` **Literal**`( java.lang.Double` **d** `)`
- *Literal*
  `public` **Literal**`( java.lang.String` **val** `)`

Methods

- *evaluate*
  `public Double` **evaluate**`( )`
- *getExpression*
  `public String` **getExpression**`( )`
- *getVariables*
  `public Set` **getVariables**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.expressions.ast.Component`

(in 16.1.4, page 129)

- *bracket*
  `public String` **bracket**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c** `)`
- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public abstract String` **getExpression**`( )`
- *getVariables*
  `public abstract Set` **getVariables**`( )`
- *order*
  `public int` **order**`( java.lang.String` **op** `)`

16.1.8 Class **NoOpComponent**

Declaration

public class NoOpComponent
extends uk.ac.ic.doc.neuralnets.expressions.ast.Component

Constructors

- *NoOpComponent*
  `public` **NoOpComponent**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **sub** `)`

Methods

- *evaluate*
  `public Double` **evaluate**`( )`
- *getExpression*
  `public String` **getExpression**`( )`
- *getVariables*
  `public Set` **getVariables**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.expressions.ast.Component`

(in 16.1.4, page 129)

- *bracket*
  `public String` **bracket**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c** `)`
- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public abstract String` **getExpression**`( )`
- *getVariables*
  `public abstract Set` **getVariables**`( )`
- *order*
  `public int` **order**`( java.lang.String` **op** `)`

16.1.9 Class NullaryOperator

Declaration

public abstract class NullaryOperator
extends uk.ac.ic.doc.neuralnets.expressions.ast.Component

Constructors

- *NullaryOperator*
  `public` **NullaryOperator**`( java.lang.String` **operation** `)`

METHODS

- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public String` **getExpression**`( )`
- *getVariables*
  `public Set` **getVariables**`( )`

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.expressions.ast.Component`

(in 16.1.4, page 129)

- *bracket*
  `public String` **bracket**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c** `)`
- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public abstract String` **getExpression**`( )`
- *getVariables*
  `public abstract Set` **getVariables**`( )`
- *order*
  `public int` **order**`( java.lang.String` **op** `)`

16.1.10 CLASS UnaryOperator

DECLARATION

public abstract class UnaryOperator
extends uk.ac.ic.doc.neuralnets.expressions.ast.Component

CONSTRUCTORS

- *UnaryOperator*
  `public` **UnaryOperator**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c**`, java.lang.String` **operation** `)`

METHODS

- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public String` **getExpression**`( )`
- *getVariables*
  `public Set` **getVariables**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.expressions.ast.Component`

(in 16.1.4, page 129)

- *bracket*
  `public String` **bracket**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c** `)`
- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public abstract String` **getExpression**`( )`
- *getVariables*
  `public abstract Set` **getVariables**`( )`
- *order*
  `public int` **order**`( java.lang.String` **op** `)`

16.1.11 Class **Variable**

Declaration

public class Variable
extends uk.ac.ic.doc.neuralnets.expressions.ast.Component

Constructors

- *Variable*
  `public` **Variable**`( java.lang.String` **name** `)`

Methods

- *bind*
  `public void` **bind**`( java.lang.Double` **val** `)`
- *evaluate*
  `public Double` **evaluate**`( )`
- *getExpression*
  `public String` **getExpression**`( )`
- *getVariables*
  `public Set` **getVariables**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.expressions.ast.Component`

(in 16.1.4, page 129)

- *bracket*
  `public String` **bracket**`( uk.ac.ic.doc.neuralnets.expressions.ast.Component` **c** `)`
- *evaluate*
  `public abstract Double` **evaluate**`( )`
- *getExpression*
  `public abstract String` **getExpression**`( )`
- *getVariables*
  `public abstract Set` **getVariables**`( )`
- *order*
  `public int` **order**`( java.lang.String` **op** `)`

Chapter 17

Package uk.ac.ic.doc.neuralnets.graph.neural

Package Contents *Page*

17.1 Interfaces

17.1.1 Interface **Persistable**

Declaration

public interface Persistable
implements java.lang.annotation.Annotation

17.2 Classes

17.2.1 Class **EdgeBase**

Declaration

public abstract class EdgeBase
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.graph.Edge

Serializable Fields

- private int id
  –

Constructors

- *EdgeBase*
  `public` **EdgeBase**`( uk.ac.ic.doc.neuralnets.graph.Node` **start**`, uk.ac.ic.doc.neuralnets.graph.Node` **end** `)`

Methods

- *getEnd*
  `public Node` **getEnd**`( )`
- *getFreshID*
  `public void` **getFreshID**`( )`
- *getID*
  `public int` **getID**`( )`
- *getStart*
  `public Node` **getStart**`( )`

- *setID*
  `public void` **setID**(`int` **id**)
- *setStart*
  `public Edge` **setStart**(`uk.ac.ic.doc.neuralnets.graph.Node` **start**)
- *setTo*
  `public Edge` **setTo**(`uk.ac.ic.doc.neuralnets.graph.Node` **end**)
- *tick*
  `public void` **tick**( )
- *toString*
  `public String` **toString**( )

17.2.2 Class **EdgeDecoration**

Declaration

public abstract class EdgeDecoration
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin, java.io.Serializable

Constructors

- *EdgeDecoration*
  `public` **EdgeDecoration**( )

Methods

- *getFigure*
  `public abstract Object` **getFigure**( )
- *getName*
  `public abstract String` **getName**( )

17.2.3 Class **EdgeSpecification**

Default EdgeSpecification

Declaration

public class EdgeSpecification
extends java.lang.Object
implements java.io.Serializable

Constructors

- *EdgeSpecification*
  `public` **EdgeSpecification**`( )`

Methods

- *getEnd*
  `public Node` **getEnd**`( )`
  - **Usage**
    - Get the end of the edge.
  - **Returns** - The end.

- *getStart*
  `public Node` **getStart**`( )`
  - **Usage**
    - Get the start of the edge.
  - **Returns** - The start.

- *getWeight*
  `public double` **getWeight**`( )`
  - **Usage**
    - Returns a random weight.
  - **Returns** - Random weight: 0 <w <1

17.2.4 Class **NetworkBridge**

Models a connection between two NeuralNetworks as a bundle of synapses

Declaration

public class NetworkBridge
extends uk.ac.ic.doc.neuralnets.graph.neural.EdgeBase

Serializable Fields

- private Set bundle
  -

Constructors

- *NetworkBridge*
  `public` **NetworkBridge**`( )`
- *NetworkBridge*
  `public` **NetworkBridge**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **start**`, uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **end** `)`

Methods

- *connect*
  `public Edge` **connect**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
- *getBundle*
  `public Collection` **getBundle**`( )`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.EdgeBase`

(in 17.2.1, page 151)

- *getEnd*
  `public Node` **getEnd**`( )`
- *getFreshID*
  `public void` **getFreshID**`( )`
- *getID*
  `public int` **getID**`( )`
- *getStart*
  `public Node` **getStart**`( )`
- *setID*
  `public void` **setID**`( int` **id** `)`
- *setStart*
  `public Edge` **setStart**`( uk.ac.ic.doc.neuralnets.graph.Node` **start** `)`
- *setTo*
  `public Edge` **setTo**`( uk.ac.ic.doc.neuralnets.graph.Node` **end** `)`
- *tick*
  `public void` **tick**`( )`
- *toString*
  `public String` **toString**`( )`

17.2.5 Class NeuralNetwork

Declaration

public class NeuralNetwork
extends uk.ac.ic.doc.neuralnets.graph.Graph
implements uk.ac.ic.doc.neuralnets.graph.Node, uk.ac.ic.doc.neuralnets.graph.Saveable

Serializable Fields

- private Set in
  –
- private Set out
  –
- private Map metadata
  –
- private int xpos
  –
- private int ypos
  –
- private int zpos
  –
- private int ticks
  –

Constructors

- *NeuralNetwork*
  `public` **NeuralNetwork**`( )`

Methods

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.neural.NetworkBridge` **e** `)`
- *getIncoming*
  `public Collection` **getIncoming**`( )`
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
- *getTicks*
  `public int` **getTicks**`( )`
- *getX*
  `public int` **getX**`( )`

- *getY*
  `public int` **getY**( )

- *getZ*
  `public int` **getZ**( )

- *resetTicks*
  `public void` **resetTicks**( )

- *setMetadata*
  `public Node` **setMetadata**(`java.lang.String` **key**, `java.lang.String` **item**)

- *setPos*
  `public void` **setPos**(`int` **x**, `int` **y**, `int` **z**)

- *tick*
  `public Node` **tick**( )

- *type*
  `protected String` **type**( )

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.Graph`

(in 18.2.1, page 184)

- *addAllNodes*
  `public Graph` **addAllNodes**(`java.util.Collection` **ns**)
  - **Usage**
    * Adds a collection of nodes to the graph, only if that collection doesn't contain itself.
  - **Parameters**
    * `ns` - Collection of nodes to add.
  - **Returns** - Itself with the nodes added or not added.

- *addEdge*
  `public Graph` **addEdge**(`uk.ac.ic.doc.neuralnets.graph.Edge` **e**)
  - **Usage**
    * Adds an edge to the graph and adds its start and end nodes to the graph.
  - **Parameters**
    * `e` - Edge to add.
  - **Returns** - Itself

- *addNode*
  `public Graph` **addNode**(`uk.ac.ic.doc.neuralnets.graph.Node` **n**)
  - **Usage**
    * Adds input node to the graph as long as input node is not itself, returns itself.
  - **Parameters**
    * `n` - Node to add.
  - **Returns** - Itself with the node added or not added.

- *forEachEdge*
  `public Graph` **forEachEdge**(`uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c**)
  - **Usage**
    * Conducts a command on each edge within the graph.
  - **Parameters**

    - ∗ `c` - Command to execute.
  - **Returns** - Itself.

- *forEachNode*
  `public Graph` **forEachNode**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c** `)`
  - **Usage**
    - ∗ Conducts a command on each node within the graph.
  - **Parameters**
    - ∗ `c` - Command to execute.
  - **Returns** - Itself.
- *getEdges*
  `public Collection` **getEdges**`( )`
  - **Usage**
    - ∗ Gets the edges from within.
  - **Returns** - The edges.
- *getFreshID*
  `public void` **getFreshID**`( )`
  - **Usage**
    - ∗ Sets the id of the object to a new fresh id.
- *getID*
  `public int` **getID**`( )`
  - **Usage**
    - ∗ Gets the id of the object.
  - **Returns** - The id.
- *getNodes*
  `public Collection` **getNodes**`( )`
  - **Usage**
    - ∗ Gets the nodes from within.
  - **Returns** - The nodes.
- *merge*
  `public Graph` **merge**`( uk.ac.ic.doc.neuralnets.graph.Graph` **o** `)`
  - **Usage**
    - ∗ Merges one graph with its self, as all the edges and nodes.
  - **Parameters**
    - ∗ `o` - Graph to merge with.
  - **Returns** - Itself
- *setID*
  `public void` **setID**`( int` **id** `)`
  - **Usage**
    - ∗ Sets the id of the object to parameter.
  - **Parameters**
    - ∗ `int` - New id.
- *toString*
  `public String` **toString**`( )`
- *type*
  `protected String` **type**`( )`
  - **Usage**
    - ∗ Returns the object type.
  - **Returns** - Object type.

17.2.6 Class **NeuralNetworkSimulationEvent**

Declaration

public class NeuralNetworkSimulationEvent
extends uk.ac.ic.doc.neuralnets.events.RevalidateStatisticiansEvent

Constructors

- *NeuralNetworkSimulationEvent*
  `public` **NeuralNetworkSimulationEvent**`( )`
- *NeuralNetworkSimulationEvent*
  `public` **NeuralNetworkSimulationEvent**`( boolean` **b** `)`

Methods

- *started*
  `public boolean` **started**`( )`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.RevalidateStatisticiansEvent`

(in 20.2.6, page 202)

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

17.2.7 Class **NeuralNetworkTickEvent**

Declaration

public class NeuralNetworkTickEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *NeuralNetworkTickEvent*
  `public` **NeuralNetworkTickEvent**`( int` **ticks** `)`

Methods

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

17.2.8 Class **Neurone**

Declaration

public class Neurone
extends uk.ac.ic.doc.neuralnets.graph.neural.NodeBase

Serializable Fields

- private String squashString
  –

Constructors

- *Neurone*
  `public` **Neurone**`( )`

Methods

- *charge*
  `public Neurone` **charge**`( double` **amt** `)`
- *getCharge*
  `public double` **getCharge**`( )`
- *getCurrentCharge*
  `public Double` **getCurrentCharge**`( )`

- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration**`( )`

- *getFreshID*
  `public void` **getFreshID**`( )`

- *getID*
  `public int` **getID**`( )`

- *getSquashFunction*
  `public ASTExpression` **getSquashFunction**`( )`

- *getTrigger*
  `public double` **getTrigger**`( )`

- *reset*
  `public void` **reset**`( )`

- *setCharge*
  `public void` **setCharge**`( double` **charge** `)`

- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`(`
  `uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration` **ed** `)`

- *setID*
  `public void` **setID**`( int` **id** `)`

- *setInitialCharge*
  `public void` **setInitialCharge**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **c** `)`

- *setSquashFunction*
  `public void` **setSquashFunction**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`

- *setTrigger*
  `public void` **setTrigger**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression`
  t `)`

- *setTrigger*
  `public void` **setTrigger**`( double` **d** `)`

- *tick*
  `public Node` **tick**`( )`
  - **Usage**
    - Ticks the neurone one step forward. Fires the neurone is appropriate.
  - **Returns** - Itself.

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.NodeBase`

(in 17.2.12, page 164)

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - Connect this node up with the input edge.
- *getIncoming*
  `public Collection` **getIncoming**`( )`
  - **Usage**
    - Get incoming edges.
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
  - **Usage**
    - Returns the meta data for the key input.
  - **Parameters**
    - `key` - To look for.
  - **Returns** - item Found.
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
  - **Usage**
    - Get outgoing edges.
- *getX*
  `public int` **getX**`( )`
  - **Usage**
    - Returns the position of the node on the x axis.
  - **Returns** - x axis position.
- *getY*
  `public int` **getY**`( )`
  - **Usage**
    - Returns the position of the node on the y axis.
  - **Returns** - y axis position.
- *getZ*
  `public int` **getZ**`( )`
  - **Usage**
    - Returns the position of the node on the z axis.
  - **Returns** - z axis position.
- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
  - **Usage**
    - Set meta data for the object.
  - **Parameters**
    - `key` - String key
    - `item` - String item

- *setPos*
  ```
  public void setPos( int  x, int  y, int  z )
  ```
  - **Usage**
    - ∗ Sets the position of the node.
  - **Parameters**
    - ∗ `x` - Position on x axis.
    - ∗ `y` - Position on y axis.
    - ∗ `z` - Position on z axis.

- *setX*
  ```
  public void setX( int  x )
  ```
  - **Usage**
    - ∗ Sets the position of the node on the x axis.
  - **Parameters**
    - ∗ `x` - Position on x axis.

- *setY*
  ```
  public void setY( int  y )
  ```
  - **Usage**
    - ∗ Sets the position of the node on the y axis.
  - **Parameters**
    - ∗ `y` - Position on y axis.

- *setZ*
  ```
  public void setZ( int  z )
  ```
  - **Usage**
    - ∗ Sets the position of the node on the z axis.
  - **Parameters**
    - ∗ `z` - Position on z axis.

- *tick*
  ```
  public abstract Node tick( )
  ```

- *toString*
  ```
  public abstract String toString( )
  ```

17.2.9 Class **NeuroneTypeConfig**

Configurator to load Statisticians

Declaration

public class NeuroneTypeConfig
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.util.configuration.Configurator

Constructors

- *NeuroneTypeConfig*
  ```
  public NeuroneTypeConfig( )
  ```

Methods

- *commitConfiguration*
  `public void` **commitConfiguration**`( )`
- *configure*
  `public void` **configure**`( )`
- *getName*
  `public String` **getName**`( )`

17.2.10 Class **NeuroneTypes**

Container object for the Neurone Types created by NeuroneTypeConfig

Declaration

public class NeuroneTypes
extends java.lang.Object

Fields

- public static final String EDGE_DECORATION_NAME
  - Magic keyword for edge decoration
- public static final Map nodeTypes
  - Map from node type name to class
- public static final Map nodeDecorations
  - Map from type name to edge decoration
- public static final Map nodeParams
  - Map from type name to list of the parameters
- public static final Map paramValues
  - Map from type name to list of the default parameter values

Constructors

- *NeuroneTypes*
  `public` **NeuroneTypes**`( )`

Methods

- *specFor*
  `public static NodeSpecification` **`specFor(`** `java.lang.String` **name** `)`
  - **Usage**
    - Build a NodeSpecification for the specified Neurone type
  - **Parameters**
    - `name` - The name of the Neurone (assumed to exist in nodeTypes)
  - **Returns** - The NodeSpecification for the given Neurone type

17.2.11 Class **NewNeuroneTypeEvent**

Indicates a new neurone type has been created

Declaration

public class NewNeuroneTypeEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *NewNeuroneTypeEvent*
  `public` **`NewNeuroneTypeEvent(`** `java.lang.String` **name** `)`

Methods

- *getName*
  `public String` **`getName(`** `)`
- *toString*
  `public String` **`toString(`** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **`toString(`** `)`

17.2.12 Class **NodeBase**

Basic Node implementation; should suffice for most Node purposes

Declaration

public abstract class NodeBase
extends java.lang.Object
implements uk.ac.ic.doc.neuralnets.graph.Node

Serializable Fields

- private Map metadata
  –
- private int xpos
  –
- private int ypos
  –
- private int zpos
  –

Constructors

- *NodeBase*
  `protected` **NodeBase**`( )`
- *NodeBase*
  `protected` **NodeBase**`( java.util.Set` **in**, `java.util.Set` **out** `)`

Methods

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - ∗ Connect this node up with the input edge.
- *getIncoming*
  `public Collection` **getIncoming**`( )`
  - **Usage**
    - ∗ Get incoming edges.
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
  - **Usage**
    - ∗ Returns the meta data for the key input.

  - **Parameters**
    - `key` - To look for.
  - **Returns** - item Found.

- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
  - **Usage**
    - Get outgoing edges.

- *getX*
  `public int` **getX**`( )`
  - **Usage**
    - Returns the position of the node on the x axis.
  - **Returns** - x axis position.

- *getY*
  `public int` **getY**`( )`
  - **Usage**
    - Returns the position of the node on the y axis.
  - **Returns** - y axis position.

- *getZ*
  `public int` **getZ**`( )`
  - **Usage**
    - Returns the position of the node on the z axis.
  - **Returns** - z axis position.

- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
  - **Usage**
    - Set meta data for the object.
  - **Parameters**
    - `key` - String key
    - `item` - String item

- *setPos*
  `public void` **setPos**`( int` **x**`, int` **y**`, int` **z** `)`
  - **Usage**
    - Sets the position of the node.
  - **Parameters**
    - `x` - Position on x axis.
    - `y` - Position on y axis.
    - `z` - Position on z axis.

- *setX*
  `public void` **setX**`( int` **x** `)`

  - **Usage**
    - Sets the position of the node on the x axis.
  - **Parameters**
    - `x` - Position on x axis.

- *setY*
  `public void` **setY**`( int` **y** `)`
  - **Usage**
    - Sets the position of the node on the y axis.
  - **Parameters**
    - `y` - Position on y axis.

- *setZ*
  `public void` **setZ**`( int` **z** `)`
  - **Usage**
    - Sets the position of the node on the z axis.
  - **Parameters**
    - `z` - Position on z axis.

- *tick*
  `public abstract Node` **tick**`( )`

- *toString*
  `public abstract String` **toString**`( )`

17.2.13 Class **NodeChargeUpdateEvent**

Declaration

public class NodeChargeUpdateEvent
extends uk.ac.ic.doc.neuralnets.events.SingletonEvent

Constructors

- *NodeChargeUpdateEvent*
  `public` **NodeChargeUpdateEvent**`(`
  `uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **n** `)`

Methods

- *equals*
  `public boolean` **equals**`( java.lang.Object` **o** `)`

- *getNeurone*
  `public Neurone` **getNeurone**`( )`

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.SingletonEvent`

(in 20.2.7, page 202)

- *equals*
  `public abstract boolean` **equals**`( java.lang.Object` **o** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

17.2.14 Class NodeFired

Declaration

public class NodeFired
extends uk.ac.ic.doc.neuralnets.events.NumericalEvent

Constructors

- *NodeFired*
  `public` **NodeFired**`( uk.ac.ic.doc.neuralnets.graph.Node` **node**`, int` **tick** `)`

Methods

- *get*
  `public double` **get**`( int` **idx** `)`
- *getNode*
  `public Node` **getNode**`( )`
- *getTick*
  `public int` **getTick**`( )`
- *numPoints*
  `public double` **numPoints**`( )`
- *push*
  `public void` **push**`( uk.ac.ic.doc.neuralnets.events.NumericalStatistician s )`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.NumericalEvent`

(in 20.2.4, page 200)

- *get*
  `public abstract double` **get**`( int` **idx** `)`
- *numPoints*
  `public abstract double` **numPoints**`( )`
- *push*
  `public abstract void` **push**`( uk.ac.ic.doc.neuralnets.events.NumericalStatistician s )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

17.2.15 Class **NodeSpecification**

Default NodeSpecification

Declaration

public class NodeSpecification
extends java.lang.Object
implements java.io.Serializable

Serializable Fields

- private Map parameters
  –
- private Class target
  –
- private EdgeDecoration ed
  –
- private String name
  –

Constructors

- *NodeSpecification*
  `public` **NodeSpecification**`( )`
- *NodeSpecification*
  `public` **NodeSpecification**`( java.lang.Class` **target** `)`

Methods

- *get*
  `public ASTExpression` **get(** `java.lang.String` **param)**
  - **Usage**
    - Get the AST expression for input parameter.
  - **Parameters**
    - `param` - String
  - **Returns** - AST expression

- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration( )**
  - **Usage**
    - Get the edge decoration for the node specification.
  - **Returns** - The edge decoration.

- *getName*
  `public String` **getName( )**
  - **Usage**
    - Get the name of the node specification.
  - **Returns** - The name.

- *getParameters*
  `public Set` **getParameters( )**
  - **Usage**
    - Get the parameter key set.
  - **Returns** - Parameter key set.

- *getTarget*
  `public Class` **getTarget( )**
  - **Usage**
    - Get target of node specification.
  - **Returns** - Target

- *set*
  `public NodeSpecification` **set(** `java.lang.String` **param,**
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **target)**
  - **Usage**
    - Set a parameter to an AST expresion.
  - **Parameters**
    - `param` - Parameter name
    - `target` - AST expression value.
  - **Returns** - Itself.

- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration` **ed** `)`
  - **Usage**
    - Set the edge decorator for the node specification.
  - **Parameters**
    - `ed` - The edge decoration.

- *setName*
  `public void` **setName**`( java.lang.String` **n** `)`
  - **Usage**
    - Set name of node specification.
  - **Parameters**
    - `n` - Name

17.2.16 Class **Perceptron**

Declaration

public class Perceptron
extends uk.ac.ic.doc.neuralnets.graph.neural.Neurone

Constructors

- *Perceptron*
  `public` **Perceptron**`( )`

Methods

- *getCharge*
  `public double` **getCharge**`( )`

- *tick*
  `public Node` **tick**`( )`

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.Neurone`

(in 17.2.8, page 159)

- *charge*
  `public Neurone` **charge**`( double` **amt** `)`
- *getCharge*
  `public double` **getCharge**`( )`
- *getCurrentCharge*
  `public Double` **getCurrentCharge**`( )`
- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration**`( )`
- *getFreshID*
  `public void` **getFreshID**`( )`
- *getID*
  `public int` **getID**`( )`
- *getSquashFunction*
  `public ASTExpression` **getSquashFunction**`( )`
- *getTrigger*
  `public double` **getTrigger**`( )`
- *reset*
  `public void` **reset**`( )`
- *setCharge*
  `public void` **setCharge**`( double` **charge** `)`
- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration` **ed** `)`
- *setID*
  `public void` **setID**`( int` **id** `)`
- *setInitialCharge*
  `public void` **setInitialCharge**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **c** `)`
- *setSquashFunction*
  `public void` **setSquashFunction**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setTrigger*
  `public void` **setTrigger**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **t** `)`
- *setTrigger*
  `public void` **setTrigger**`( double` **d** `)`
- *tick*
  `public Node` **tick**`( )`
  - **Usage**
    - Ticks the neurone one step forward. Fires the neurone is appropriate.
  - **Returns** - Itself.
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.NodeBase`

(in 17.2.12, page 164)

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - Connect this node up with the input edge.
- *getIncoming*
  `public Collection` **getIncoming**`( )`
  - **Usage**
    - Get incoming edges.
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
  - **Usage**
    - Returns the meta data for the key input.
  - **Parameters**
    - `key` - To look for.
  - **Returns** - item Found.
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
  - **Usage**
    - Get outgoing edges.
- *getX*
  `public int` **getX**`( )`
  - **Usage**
    - Returns the position of the node on the x axis.
  - **Returns** - x axis position.
- *getY*
  `public int` **getY**`( )`
  - **Usage**
    - Returns the position of the node on the y axis.
  - **Returns** - y axis position.
- *getZ*
  `public int` **getZ**`( )`
  - **Usage**
    - Returns the position of the node on the z axis.
  - **Returns** - z axis position.
- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
  - **Usage**
    - Set meta data for the object.
  - **Parameters**
    - `key` - String key
    - `item` - String item

- *setPos*
  `public void setPos( int x, int y, int z )`
  - **Usage**
    - Sets the position of the node.
  - **Parameters**
    - `x` - Position on x axis.
    - `y` - Position on y axis.
    - `z` - Position on z axis.
- *setX*
  `public void setX( int x )`
  - **Usage**
    - Sets the position of the node on the x axis.
  - **Parameters**
    - `x` - Position on x axis.
- *setY*
  `public void setY( int y )`
  - **Usage**
    - Sets the position of the node on the y axis.
  - **Parameters**
    - `y` - Position on y axis.
- *setZ*
  `public void setZ( int z )`
  - **Usage**
    - Sets the position of the node on the z axis.
  - **Parameters**
    - `z` - Position on z axis.
- *tick*
  `public abstract Node tick( )`
- *toString*
  `public abstract String toString( )`

17.2.17 Class SpikingNeurone

Declaration

public class SpikingNeurone
extends uk.ac.ic.doc.neuralnets.graph.neural.Neurone

Serializable Fields

- private double recoveryScale
  –
- private double recoverySensitivity
  –
- private double psr
  –
- private double u
  –
- private double psrRecovery
  –
- private double chargeUp
  –
- private String thalamicString
  –
- private String synapticDelayString
  –
- private int fired
  –
- private List delays
  –
- private Synapse outbound
  –

Constructors

- *SpikingNeurone*
  `public` **SpikingNeurone**`( )`

Methods

- *charge*
  `public Neurone` **charge**`( double` **amt** `)`
- *getPostSpikeReset*
  `public Double` **getPostSpikeReset**`( )`
- *getPSRRecovery*
  `public Double` **getPSRRecovery**`( )`
- *getRecoveryScale*
  `public Double` **getRecoveryScale**`( )`
- *getRecoverySensitivity*
  `public Double` **getRecoverySensitivity**`( )`
- *setPostSpikeReset*
  `public void` **setPostSpikeReset**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setPSRRecovery*
  `public void` **setPSRRecovery**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setRecoveryScale*
  `public void` **setRecoveryScale**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setRecoverySensitivity*
  `public void` **setRecoverySensitivity**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setSynapticDelay*
  `public void` **setSynapticDelay**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setThalamicInput*
  `public void` **setThalamicInput**`(`
  `uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *tick*
  `public Node` **tick**`( )`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.Neurone`

(in 17.2.8, page 159)

- *charge*
  `public Neurone` **charge**`( double` **amt** `)`
- *getCharge*
  `public double` **getCharge**`( )`

- *getCurrentCharge*
  `public Double` **getCurrentCharge**`( )`
- *getEdgeDecoration*
  `public EdgeDecoration` **getEdgeDecoration**`( )`
- *getFreshID*
  `public void` **getFreshID**`( )`
- *getID*
  `public int` **getID**`( )`
- *getSquashFunction*
  `public ASTExpression` **getSquashFunction**`( )`
- *getTrigger*
  `public double` **getTrigger**`( )`
- *reset*
  `public void` **reset**`( )`
- *setCharge*
  `public void` **setCharge**`( double` **charge** `)`
- *setEdgeDecoration*
  `public void` **setEdgeDecoration**`( uk.ac.ic.doc.neuralnets.graph.neural.EdgeDecoration` **ed** `)`
- *setID*
  `public void` **setID**`( int` **id** `)`
- *setInitialCharge*
  `public void` **setInitialCharge**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **c** `)`
- *setSquashFunction*
  `public void` **setSquashFunction**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **e** `)`
- *setTrigger*
  `public void` **setTrigger**`( uk.ac.ic.doc.neuralnets.expressions.ast.ASTExpression` **t** `)`
- *setTrigger*
  `public void` **setTrigger**`( double` **d** `)`
- *tick*
  `public Node` **tick**`( )`
  - **Usage**
    - Ticks the neurone one step forward. Fires the neurone is appropriate.
  - **Returns** - Itself.
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.NodeBase`

(in 17.2.12, page 164)

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    - Connect this node up with the input edge.
- *getIncoming*
  `public Collection` **getIncoming**`( )`
  - **Usage**

    - Get incoming edges.

- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
  - **Usage**
    - Returns the meta data for the key input.
  - **Parameters**
    - `key` - To look for.
  - **Returns** - item Found.

- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
  - **Usage**
    - Get outgoing edges.

- *getX*
  `public int` **getX**`( )`
  - **Usage**
    - Returns the position of the node on the x axis.
  - **Returns** - x axis position.

- *getY*
  `public int` **getY**`( )`
  - **Usage**
    - Returns the position of the node on the y axis.
  - **Returns** - y axis position.

- *getZ*
  `public int` **getZ**`( )`
  - **Usage**
    - Returns the position of the node on the z axis.
  - **Returns** - z axis position.

- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
  - **Usage**
    - Set meta data for the object.
  - **Parameters**
    - `key` - String key
    - `item` - String item

- *setPos*
  `public void` **setPos**`( int` **x**`, int` **y**`, int` **z** `)`
  - **Usage**
    - Sets the position of the node.
  - **Parameters**
    - `x` - Position on x axis.
    - `y` - Position on y axis.
    - `z` - Position on z axis.

- *setX*
  `public void` **setX**`( int` **x** `)`
  - **Usage**

    - Sets the position of the node on the x axis.
  - **Parameters**
    - `x` - Position on x axis.

- *setY*
  `public void` **setY**`( int` **y** `)`
  - **Usage**
    - Sets the position of the node on the y axis.
  - **Parameters**
    - `y` - Position on y axis.

- *setZ*
  `public void` **setZ**`( int` **z** `)`
  - **Usage**
    - Sets the position of the node on the z axis.
  - **Parameters**
    - `z` - Position on z axis.

- *tick*
  `public abstract Node` **tick**`( )`

- *toString*
  `public abstract String` **toString**`( )`

17.2.18 Class **Synapse**

Declaration

public class Synapse
extends uk.ac.ic.doc.neuralnets.graph.neural.EdgeBase

Serializable Fields

- private double weight
  –
- private int delay
  –

Constructors

- *Synapse*
  `public` **Synapse**`( )`

- *Synapse*
  `public` **Synapse**`( double` **weight**`,`
  `uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **start**`,`
  `uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **end** `)`

- *Synapse*
  `public` **Synapse**`( uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **start**`,`
  `uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **end** `)`

Methods

- *fire*
  `public Synapse` **fire**`( double` **amt** `)`
- *getDelay*
  `public int` **getDelay**`( )`
- *getWeight*
  `public double` **getWeight**`( )`
- *setDelay*
  `public Synapse` **setDelay**`( int` **d** `)`
- *setWeight*
  `public Synapse` **setWeight**`( double` **weight** `)`
- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.graph.neural.EdgeBase`

(in 17.2.1, page 151)

- *getEnd*
  `public Node` **getEnd**`( )`
- *getFreshID*
  `public void` **getFreshID**`( )`
- *getID*
  `public int` **getID**`( )`
- *getStart*
  `public Node` **getStart**`( )`
- *setID*
  `public void` **setID**`( int` **id** `)`
- *setStart*
  `public Edge` **setStart**`( uk.ac.ic.doc.neuralnets.graph.Node` **start** `)`
- *setTo*
  `public Edge` **setTo**`( uk.ac.ic.doc.neuralnets.graph.Node` **end** `)`
- *tick*
  `public void` **tick**`( )`
- *toString*
  `public String` **toString**`( )`

Chapter 18

Package uk.ac.ic.doc.neuralnets.graph

18.1 Interfaces

18.1.1 Interface **Edge**

Declaration

public interface Edge
implements java.io.Serializable, Identifiable

Methods

- *getEnd*
  `public Node` **getEnd**`( )`
- *getStart*
  `public Node` **getStart**`( )`
- *setStart*
  `public Edge` **setStart**`( uk.ac.ic.doc.neuralnets.graph.Node` **start** `)`
- *setTo*
  `public Edge` **setTo**`( uk.ac.ic.doc.neuralnets.graph.Node` **end** `)`
- *tick*
  `public void` **tick**`( )`

18.1.2 Interface **Graph.Command**

Declaration

public static interface Graph.Command

Methods

- *exec*
  `public void` **exec**`( java.lang.Object` **input** `)`

18.1.3 Interface **Identifiable**

Declaration

public interface Identifiable

Methods

- *getFreshID*
  `public void` **getFreshID**`( )`
- *getID*
  `public int` **getID**`( )`
- *setID*
  `public void` **setID**`( int` **id** `)`

18.1.4 Interface Node

Declaration

public interface Node
implements java.io.Serializable, Identifiable

Methods

- *connect*
  `public Node` **connect**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
- *getIncoming*
  `public Collection` **getIncoming**`( )`
- *getMetadata*
  `public String` **getMetadata**`( java.lang.String` **key** `)`
- *getOutgoing*
  `public Collection` **getOutgoing**`( )`
- *getX*
  `public int` **getX**`( )`
- *getY*
  `public int` **getY**`( )`
- *getZ*
  `public int` **getZ**`( )`
- *setMetadata*
  `public Node` **setMetadata**`( java.lang.String` **key**`, java.lang.String` **item** `)`
- *setPos*
  `public void` **setPos**`( int` **x**`, int` **y**`, int` **z** `)`
- *tick*
  `public Node` **tick**`( )`
  - **Usage**
    - States that this node has advanced one ”tick” in time

18.1.5 Interface **Saveable**

Declaration

public interface Saveable
implements java.io.Serializable

18.2 Classes

18.2.1 Class **Graph**

Declaration

public class Graph
extends java.lang.Object
implements java.io.Serializable, Identifiable

Serializable Fields

- private int id
  –

Constructors

- *Graph*
  `public` **Graph**`( )`

Methods

- *addAllNodes*
  `public Graph` **addAllNodes**`( java.util.Collection` **ns** `)`
  - **Usage**
    * Adds a collection of nodes to the graph, only if that collection doesn't contain itself.
  - **Parameters**
    * `ns` - Collection of nodes to add.
  - **Returns** - Itself with the nodes added or not added.

- *addEdge*
  `public Graph` **addEdge**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`

  - **Usage**
    - Adds an edge to the graph and adds its start and end nodes to the graph.
  - **Parameters**
    - `e` - Edge to add.
  - **Returns** - Itself

- *addNode*
  `public Graph` **addNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**
    - Adds input node to the graph as long as input node is not itself, returns itself.
  - **Parameters**
    - `n` - Node to add.
  - **Returns** - Itself with the node added or not added.

- *forEachEdge*
  `public Graph` **forEachEdge**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c** `)`
  - **Usage**
    - Conducts a command on each edge within the graph.
  - **Parameters**
    - `c` - Command to execute.
  - **Returns** - Itself.

- *forEachNode*
  `public Graph` **forEachNode**`( uk.ac.ic.doc.neuralnets.graph.Graph.Command` **c** `)`
  - **Usage**
    - Conducts a command on each node within the graph.
  - **Parameters**
    - `c` - Command to execute.
  - **Returns** - Itself.

- *getEdges*
  `public Collection` **getEdges**`( )`
  - **Usage**
    - Gets the edges from within.
  - **Returns** - The edges.

- *getFreshID*
  `public void` **getFreshID**`( )`
  - **Usage**
    - Sets the id of the object to a new fresh id.

- *getID*
  `public int` **getID**`( )`
  - **Usage**

  - ∗ Gets the id of the object.
  - **Returns** - The id.

- *getNodes*
  `public Collection` **getNodes**`( )`
  - **Usage**
    - ∗ Gets the nodes from within.
  - **Returns** - The nodes.

- *merge*
  `public Graph` **merge**`( uk.ac.ic.doc.neuralnets.graph.Graph` **o** `)`
  - **Usage**
    - ∗ Merges one graph with its self, as all the edges and nodes.
  - **Parameters**
    - ∗ `o` - Graph to merge with.
  - **Returns** - Itself

- *setID*
  `public void` **setID**`( int` **id** `)`
  - **Usage**
    - ∗ Sets the id of the object to parameter.
  - **Parameters**
    - ∗ `int` - New id.

- *toString*
  `public String` **toString**`( )`

- *type*
  `protected String` **type**`( )`
  - **Usage**
    - ∗ Returns the object type.
  - **Returns** - Object type.

18.2.2 Class **GraphStreamer**

Declaration

---

public class GraphStreamer
extends java.lang.Object

Constructors

- *GraphStreamer*
  `public` **GraphStreamer**`( uk.ac.ic.doc.neuralnets.graph.Graph` **g**`,`
  `uk.ac.ic.doc.neuralnets.util.Transformer` **edgeMaker**`,`
  `uk.ac.ic.doc.neuralnets.util.Transformer` **nodeMaker** `)`

METHODS

- *getEdgeIterator*
  `public Iterator` **`getEdgeIterator`**`( )`
  - **Usage**
    * Returns an iterator for the edges that are contained in the GraphStreamer
  - **Returns** - Iterator of edges.

- *getNodeIterator*
  `public Iterator` **`getNodeIterator`**`( )`
  - **Usage**
    * Returns an iterator for the nodes that are contained in the GraphStreamer
  - **Returns** - Iterator of nodes.

18.2.3 CLASS **Metadata**

Constants for use in setting and getting metadata Useful to keep all in one place, should be inlined by compiler too.

DECLARATION

public class Metadata
extends java.lang.Object

FIELDS

- public static final String X_POS
  –
- public static final String Y_POS
  –

CONSTRUCTORS

- *Metadata*
  `public` **`Metadata`**`( )`

Chapter 19

Package uk.ac.ic.doc.neuralnets.coreui

Package Contents *Page*

Classes

19.1 Classes

19.1.1 Class **InterfaceManager**

Declaration

public abstract class InterfaceManager
extends java.lang.Object

Constructors

- *InterfaceManager*
  `public` **InterfaceManager**`( )`

Methods

- *addConnection*
  `public void` **addConnection**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    * Adds the given edge to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `e` -

- *addNetwork*
  `public void` **addNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    * Adds the given neural network to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `n` - the neural network to add to the current section of the neural network

- *addNeurone*
  `public void` **addNeurone**`( uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **n** `)`
  - **Usage**
    * Adds the given neurone to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `n` - the neurone to add to the current section of the neural network

- *addNode*
  `public void` **addNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**

  - ∗ Adds the given node to the current view, and redraws the screen as necessary.
- **Parameters**
  - ∗ `n` - the node to add to the current section of the neural network

---

- *addNode*
  `public void` **addNode**`( uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification` **spec** `)`
  - **Usage**
    - ∗ Creates a node from the give specification, adds to the current view, and redraws the screen as necessary.
  - **Parameters**
    - ∗ `spec` - the specification of the node to add to the current section of the neural network

---

- *getCommandControl*
  `public CommandControl` **getCommandControl**`( )`
  - **Usage**
    - ∗ Gets the command control used by the GUIManager. This object handles the undo and redo stacks as commands are executed and undone.
  - **Returns** - the CommandControl object used by the GUIManager

---

- *getCurrentNetwork*
  `public abstract NeuralNetwork` **getCurrentNetwork**`( )`
  - **Usage**
    - ∗ Returns the neural network layer currently being viewed in the GUIManager.
  - **Returns** - the current neural network layer

- *getGraph*
  `public abstract Object` **getGraph**`( )`
  - **Usage**
    - ∗ Returns the Graph representation used by this UI Manager.
  - **Returns** - the Graph that the Manager draws onto

- *getNode*
  `public abstract Object` **getNode**`(`
  `uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **n** `)`
  - **Usage**
    - ∗ Finds the GUINode in the GUI corresponding to the given Neurone and returns it. Returns null if the given Neurone is not loaded in the GUI.
  - **Parameters**
    - ∗ `n` - the Neurone to look up in the GUI
  - **Returns** - the GUINode in the GUI corresponding to the given Neurone

---

- *getRootNetwork*
  `public NeuralNetwork` **getRootNetwork**`( )`

  - **Usage**
    - Gets the root of the layered neural network stored in the GUIManager.
  - **Returns** - the root of the main neural network

- *getSaveLocation*
  `public FileSpecification` **getSaveLocation**`( )`
  - **Usage**
    - Gets the location to save the network to, or null if no such location exists.
  - **Returns** - the network's save location, or null if none exists

---

- *getUtils*
  `public InteractionUtils` **getUtils**`( )`
  - **Usage**
    - Returns the GUIManager's interaction utilities.
  - **Returns** - the InteractionUtils object used by the GUIManager

---

- *persistLocations*
  `public abstract void` **persistLocations**`( )`
  - **Usage**
    - Pushes down the locations of all Nodes to the model. Allows positions to be persisted to storage and reloaded.

- *redrawCurrentView*
  `public abstract void` **redrawCurrentView**`( )`
  - **Usage**
    - Draws the current view of the graph. Imports the current network layer from the internal model and applies the current layout.

- *remove*
  `public abstract void` **remove**`( java.lang.Object` **i** `)`
  - **Usage**
    - Removes the given GraphItem from the view.
  - **Parameters**
    - `i` - the graphitem to be removed from the view

- *removeNetwork*
  `public void` **removeNetwork**`(`
  `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    - Removes the given neural network from the current view, and redraws the screen as necessary.
  - **Parameters**
    - `n` - the neural network to remove from the current section of the neural network

---

- *reset*
  `protected abstract void` **reset**`( )`
  - **Usage**
    - Reset the current manager, e.g. when a new network is loaded

---

- *setNetwork*
  `public void` **setNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **network**`, uk.ac.ic.doc.neuralnets.persistence.FileSpecification` **location** `)`
  - **Usage**
    - Loads the given neural network into the GUIManager, from the given location.
  - **Parameters**
    - `network` - the network to be loaded into the GUIManager
    - `location` - the location to load the network from

- *setSaveLocation*
  `public void` **setSaveLocation**`( uk.ac.ic.doc.neuralnets.persistence.FileSpecification` **saveLoc** `)`
  - **Usage**
    - Sets the network's save location.
  - **Parameters**
    - `saveLoc` -

- *updateInterfaceHints*
  `public abstract void` **updateInterfaceHints**`( )`
  - **Usage**
    - Updates the tooltips or other UI hints of all graph elements in the current view.

19.1.2 Class **ZoomingInterfaceManager**

Declaration

public abstract class ZoomingInterfaceManager
extends uk.ac.ic.doc.neuralnets.coreui.InterfaceManager

Constructors

- *ZoomingInterfaceManager*
  `public` **ZoomingInterfaceManager**`( )`

METHODS

- *canZoomIn*
  `public abstract boolean` **canZoomIn**`( )`
  - **Usage**
    - Checks whether or not it is possible to zoom in. It is only possible to zoom in if exactly one internal network layer is selected.
  - **Returns** - whether or not it is possible to zoom in

- *canZoomOut*
  `public abstract boolean` **canZoomOut**`( )`
  - **Usage**
    - Checks whether or not it is possible to zoom out. It is always possible to zoom out unless the current view is the root network.
  - **Returns** - whether or not it is possible to zoom out

- *getZoomIDs*
  `public abstract Stack` **getZoomIDs**`( )`
  - **Usage**
    - Returns a stack containing the IDs of each network layer that has currently been zoomed into. This can be used to trace the current zoom path from the root of the neural network.
  - **Returns** - a stack of IDs of each network layer that is currently zoomed into

- *getZoomLevels*
  `public abstract Stack` **getZoomLevels**`( )`
  - **Usage**
    - Returns a stack containing each network layer that has currently been zoomed into, starting with the root network.
  - **Returns** - a stack containing each network layer that has currently been zoomed into.

- *zoomIn*
  `public abstract void` **zoomIn**`(`
  `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    - Zooms into the selected network layer. Clears the current view, and instead shows the contents of the selected network layer.
  - **Parameters**
    - `n` - the network to zoom into.

- *zoomOut*
  `public abstract void` **zoomOut**`( )`
  - **Usage**
    - Zooms out one layer. Clears the current view, and instead shows the contents of the current layer's parent. If the current view is the root network, then nothing happens as it is not possible to zoom out further.

Methods inherited from class `uk.ac.ic.doc.neuralnets.coreui.InterfaceManager`

(in 19.1.1, page 189)

- *addConnection*
  `public void` **addConnection**`( uk.ac.ic.doc.neuralnets.graph.Edge` **e** `)`
  - **Usage**
    * Adds the given edge to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `e` -

- *addNetwork*
  `public void` **addNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    * Adds the given neural network to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `n` - the neural network to add to the current section of the neural network

---

- *addNeurone*
  `public void` **addNeurone**`( uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **n** `)`
  - **Usage**
    * Adds the given neurone to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `n` - the neurone to add to the current section of the neural network

---

- *addNode*
  `public void` **addNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**
    * Adds the given node to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `n` - the node to add to the current section of the neural network

---

- *addNode*
  `public void` **addNode**`( uk.ac.ic.doc.neuralnets.graph.neural.NodeSpecification` **spec** `)`
  - **Usage**
    * Creates a node from the give specification, adds to the current view, and redraws the screen as necessary.
  - **Parameters**
    * `spec` - the specification of the node to add to the current section of the neural network

---

- *getCommandControl*
  `public CommandControl` **getCommandControl**`( )`
  - **Usage**
    * Gets the command control used by the GUIManager. This object handles the undo and redo stacks as commands are executed and undone.
  - **Returns** - the CommandControl object used by the GUIManager

---

- *getCurrentNetwork*
  `public abstract NeuralNetwork` **getCurrentNetwork**`( )`
  - **Usage**
    * Returns the neural network layer currently being viewed in the GUIManager.

- **Returns** - the current neural network layer

- *getGraph*
  `public abstract Object` **`getGraph`**`( )`
  - **Usage**
    - Returns the Graph representation used by this UI Manager.
  - **Returns** - the Graph that the Manager draws onto

- *getNode*
  `public abstract Object` **`getNode`**`( uk.ac.ic.doc.neuralnets.graph.neural.Neurone` **`n`** `)`
  - **Usage**
    - Finds the GUINode in the GUI corresponding to the given Neurone and returns it. Returns null if the given Neurone is not loaded in the GUI.
  - **Parameters**
    - `n` - the Neurone to look up in the GUI
  - **Returns** - the GUINode in the GUI corresponding to the given Neurone

---

- *getRootNetwork*
  `public NeuralNetwork` **`getRootNetwork`**`( )`
  - **Usage**
    - Gets the root of the layered neural network stored in the GUIManager.
  - **Returns** - the root of the main neural network

- *getSaveLocation*
  `public FileSpecification` **`getSaveLocation`**`( )`
  - **Usage**
    - Gets the location to save the network to, or null if no such location exists.
  - **Returns** - the network's save location, or null if none exists

---

- *getUtils*
  `public InteractionUtils` **`getUtils`**`( )`
  - **Usage**
    - Returns the GUIManager's interaction utilities.
  - **Returns** - the InteractionUtils object used by the GUIManager

---

- *persistLocations*
  `public abstract void` **`persistLocations`**`( )`
  - **Usage**
    - Pushes down the locations of all Nodes to the model. Allows positions to be persisted to storage and reloaded.

- *redrawCurrentView*
  `public abstract void` **`redrawCurrentView`**`( )`
  - **Usage**
    - Draws the current view of the graph. Imports the current network layer from the internal model and applies the current layout.

- *remove*
  `public abstract void` **`remove`**`( java.lang.Object` **`i`** `)`
  - **Usage**
    - Removes the given GraphItem from the view.
  - **Parameters**

    - `i` - the graphitem to be removed from the view

- *removeNetwork*
  `public void` **removeNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **n** `)`
  - **Usage**
    - Removes the given neural network from the current view, and redraws the screen as necessary.
  - **Parameters**
    - `n` - the neural network to remove from the current section of the neural network

---

- *reset*
  `protected abstract void` **reset**`( )`
  - **Usage**
    - Reset the current manager, e.g. when a new network is loaded

---

- *setNetwork*
  `public void` **setNetwork**`( uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **network**`, uk.ac.ic.doc.neuralnets.persistence.FileSpecification` **location** `)`
  - **Usage**
    - Loads the given neural network into the GUIManager, from the given location.
  - **Parameters**
    - `network` - the network to be loaded into the GUIManager
    - `location` - the location to load the network from

- *setSaveLocation*
  `public void` **setSaveLocation**`( uk.ac.ic.doc.neuralnets.persistence.FileSpecification` **saveLoc** `)`
  - **Usage**
    - Sets the network's save location.
  - **Parameters**
    - `saveLoc` -

- *updateInterfaceHints*
  `public abstract void` **updateInterfaceHints**`( )`
  - **Usage**
    - Updates the tooltips or other UI hints of all graph elements in the current view.

Chapter 20

Package uk.ac.ic.doc.neuralnets.events

Package Contents *Page*

20.1 Interfaces

20.1.1 Interface **EventHandler**

Declaration

public interface EventHandler
implements uk.ac.ic.doc.neuralnets.util.plugins.Plugin

Methods

- *flush*
  `public void` **flush**`( )`
  - **Usage**
    - Instructs this Statistician to flush its buffers of data (usually indicating that execution has completed)

- *handle*
  `public void` **handle**`( uk.ac.ic.doc.neuralnets.events.Event e )`
  - **Usage**
    - Fires an event at this Statistician
  - **Parameters**
    - `e` - The event which has occurred

- *isValid*
  `public boolean` **isValid**`( )`
  - **Usage**
    - Answers whether or not this Statistician is valid for execution. If not, when a new Neural Network run begins the Statistician may be re-created by the StatisticsManager.
  - **Returns** - True iff this Statistician may process new input

20.2 Classes

20.2.1 Class **Event**

Declaration

public abstract class Event
extends java.lang.Object

Constructors

- *Event*
  `public` **Event**`( )`

Methods

- *toString*
  `public abstract String` **toString**`( )`

20.2.2 Class **EventManager**

Declaration

public class EventManager
extends java.lang.Object

Methods

- *deregisterAsync*
  `public void` **deregisterAsync**`( java.lang.Class` **c**`, uk.ac.ic.doc.neuralnets.events.EventHandler` **s** `)`
- *deregisterSynchro*
  `public void` **deregisterSynchro**`( java.lang.Class` **c**`, uk.ac.ic.doc.neuralnets.events.EventHandler` **s** `)`
- *fire*
  `public void` **fire**`( uk.ac.ic.doc.neuralnets.events.Event` **e** `)`
- *flush*
  `public boolean` **flush**`( java.lang.Class` **e** `)`
- *flushAll*
  `public void` **flushAll**`( )`
- *get*
  `public static EventManager` **get**`( )`
- *getUniqueID*
  `public synchronized int` **getUniqueID**`( )`
- *handle*
  `protected void` **handle**`( java.lang.Class` **c**`, uk.ac.ic.doc.neuralnets.events.Event` **e**`, java.util.Map` **handlers** `)`
- *registerAsync*
  `public void` **registerAsync**`( java.lang.Class` **c**`, uk.ac.ic.doc.neuralnets.events.EventHandler` **s** `)`

- *registerSynchro*
  `public void` **registerSynchro**`( java.lang.Class  c, uk.ac.ic.doc.neuralnets.events.EventHandler  s )`

20.2.3 Class GraphUpdateEvent

Declaration

public class GraphUpdateEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *GraphUpdateEvent*
  `public` **GraphUpdateEvent**`( )`

Methods

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

20.2.4 Class NumericalEvent

Declaration

public abstract class NumericalEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *NumericalEvent*
  `public` **NumericalEvent**`( )`

METHODS

- *get*
  `public abstract double` **get**`( int` **idx** `)`
- *numPoints*
  `public abstract double` **numPoints**`( )`
- *push*
  `public abstract void` **push**`(`
  `uk.ac.ic.doc.neuralnets.events.NumericalStatistician` **s** `)`

METHODS INHERITED FROM CLASS `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

20.2.5 CLASS **NumericalStatistician**

DECLARATION

public abstract class NumericalStatistician
extends java.lang.Object
implements EventHandler

CONSTRUCTORS

- *NumericalStatistician*
  `public` **NumericalStatistician**`( )`

METHODS

- *handle*
  `public void` **handle**`( uk.ac.ic.doc.neuralnets.events.Event` **e** `)`
- *handle*
  `public void` **handle**`( java.lang.Integer []` **vs** `)`
- *handle*
  `public void` **handle**`( java.util.List` **vs** `)`
- *handle*
  `public void` **handle**`( uk.ac.ic.doc.neuralnets.events.NumericalEvent` **e** `)`
- *isValid*
  `public boolean` **isValid**`( )`
- *saveAs*
  `public void` **saveAs**`( java.lang.String` **file** `)`

20.2.6 Class **RevalidateStatisticiansEvent**

Declaration

public class RevalidateStatisticiansEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *RevalidateStatisticiansEvent*
  `public` **RevalidateStatisticiansEvent**`( )`

Methods

- *toString*
  `public String` **toString**`( )`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

20.2.7 Class **SingletonEvent**

Declaration

public abstract class SingletonEvent
extends uk.ac.ic.doc.neuralnets.events.Event

Constructors

- *SingletonEvent*
  `public` **SingletonEvent**`( )`

Methods

- *equals*
  `public abstract boolean` **equals**`( java.lang.Object` **o** `)`

Methods inherited from class `uk.ac.ic.doc.neuralnets.events.Event`

(in 20.2.1, page 198)

- *toString*
  `public abstract String` **toString**`( )`

Chapter 21

Package uk.ac.ic.doc.neuralnets.util.reflect

Package Contents *Page*

Classes

21.1 Classes

21.1.1 Class MethodPseudoAccessor

Declaration

public class MethodPseudoAccessor
extends java.lang.Object
implements sun.reflect.FieldAccessor

Constructors

- *MethodPseudoAccessor*
  `public` **MethodPseudoAccessor**`( java.lang.Class` **c**`, java.lang.String` **f** `)`
- *MethodPseudoAccessor*
  `public` **MethodPseudoAccessor**`( java.lang.reflect.Field` **f** `)`

Methods

- *get*
  `public Object` **get**`( java.lang.Object` **o** `)`
- *getBoolean*
  `public boolean` **getBoolean**`( java.lang.Object` **o** `)`
- *getByte*
  `public byte` **getByte**`( java.lang.Object` **o** `)`
- *getChar*
  `public char` **getChar**`( java.lang.Object` **o** `)`
- *getDouble*
  `public double` **getDouble**`( java.lang.Object` **o** `)`
- *getFloat*
  `public float` **getFloat**`( java.lang.Object` **o** `)`
- *getInt*
  `public int` **getInt**`( java.lang.Object` **o** `)`
- *getLong*
  `public long` **getLong**`( java.lang.Object` **o** `)`
- *getShort*
  `public short` **getShort**`( java.lang.Object` **o** `)`
- *set*
  `public void` **set**`( java.lang.Object` **o**`, java.lang.Object` **v** `)`

- *setBoolean*
  `public void` **setBoolean**`( java.lang.Object` **o**`, boolean` **b** `)`
- *setByte*
  `public void` **setByte**`( java.lang.Object` **o**`, byte` **b** `)`
- *setChar*
  `public void` **setChar**`( java.lang.Object` **o**`, char` **c** `)`
- *setDouble*
  `public void` **setDouble**`( java.lang.Object` **o**`, double` **d** `)`
- *setFloat*
  `public void` **setFloat**`( java.lang.Object` **o**`, float` **f** `)`
- *setInt*
  `public void` **setInt**`( java.lang.Object` **o**`, int` **i** `)`
- *setLong*
  `public void` **setLong**`( java.lang.Object` **o**`, long` **l** `)`
- *setShort*
  `public void` **setShort**`( java.lang.Object` **o**`, short` **s** `)`

21.1.2 Class **ReflectionHelper**

Used to perform potentially unsafe reflection - e.g. setting private fields, or getting Fields that backend to Methods.

Declaration

public class ReflectionHelper
extends java.lang.Object

Constructors

- *ReflectionHelper*
  `public` **ReflectionHelper**`( )`

Methods

- *getMethodField*
  `public static final Field` **getMethodField**`( java.lang.String` **m**`, java.lang.Class` **c** `)`
  - **Usage**
    - Get a Field object which backends data access to the given method name, from the supplied class
  - **Parameters**

  - ∗ `m` - The name of the method
  - ∗ `c` - The class to get the method from
- **Returns** - a Field with an accessor that backends to the requested Method
- **Exceptions**
  - ∗ `java.lang.NoSuchMethodException` -
  - ∗ `java.lang.IllegalArgumentException` -
  - ∗ `java.lang.IllegalAccessException` -

- *getReflectionFactory*
  `public static final ReflectionFactory` **getReflectionFactory**`( )`
  - **Usage**
    - ∗ Get the Sun-JVM-specific ReflectionFactory object (in an unsafe manner). This allows us to assign values to and read from private Fields
  - **Returns** - the ReflectionFactory

- *set*
  `public static final void` **set**`( java.lang.Class` **c**`, java.lang.String` **fi**`, java.lang.Object` **target**`, java.lang.Object` **v** `)`
  - **Usage**
    - ∗ Find the requested Field declared in the given class, and set its value (irrespective of the field's modifiers)
  - **Parameters**
    - ∗ `c` - The Class to look in
    - ∗ `fi` - The field name to seek
    - ∗ `target` - The target object
    - ∗ `v` - The value to set the field to
  - **Exceptions**
    - ∗ `java.lang.IllegalArgumentException` -
    - ∗ `java.lang.IllegalAccessException` -

- *set*
  `public static final void` **set**`( java.lang.reflect.Field` **f**`, java.lang.Object` **target**`, java.lang.Object` **v** `)`
  - **Usage**
    - ∗ Set the given field on target to value, irrespective of its modifiers
  - **Parameters**
    - ∗ `f` - The Field to set
    - ∗ `target` - The object to set it on
    - ∗ `v` - The value to set the field to
  - **Exceptions**
    - ∗ `java.lang.IllegalArgumentException` -
    - ∗ `java.lang.IllegalAccessException` -

- *set*
  `public static final void` **set**`( java.lang.String` **fi**`, java.lang.Object` **target**`, java.lang.Object` **v** `)`
  - **Usage**

  - Find the requested Field declared in the target object's class, and set its value (irrespective of the field's modifiers)
- **Parameters**
  - `fi` - The field name to seek
  - `target` - The target object
  - `v` - The value to set the field to
- **Exceptions**
  - `java.lang.IllegalArgumentException` -
  - `java.lang.IllegalAccessException` -

Chapter 22

Package uk.ac.ic.doc.neuralnets.gui.graph

Package Contents *Page*

Interfaces

Classes

22.1 Interfaces

22.1.1 Interface **NodeContainer**

Objects of this type contain a model Node.

Declaration

```
public interface NodeContainer
```

Methods

- *getNode*
  `public Node` **getNode**`( )`
  - **Usage**
    - Get the node contained in the container.
  - **Returns** - the contained node
- *setNode*
  `public void` **setNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **n** `)`
  - **Usage**
    - Set the node contained in the container.
  - **Parameters**
    - `n` -

22.2 Classes

22.2.1 Class **CachingLayout**

Declaration

public class CachingLayout
extends java.lang.Object
implements org.eclipse.zest.layouts.LayoutAlgorithm

Constructors

- *CachingLayout*
  `public` **CachingLayout**`( )`
- *CachingLayout*
  `public` **CachingLayout**`( org.eclipse.zest.layouts.LayoutAlgorithm` **child** `)`

- *CachingLayout*
  `public` **CachingLayout**`( org.eclipse.zest.layouts.LayoutAlgorithm` **child**`, boolean` **useCache** `)`

Methods

- *addEntity*
  `public void` **addEntity**`( org.eclipse.zest.layouts.LayoutEntity` **entity** `)`

- *addProgressListener*
  `public void` **addProgressListener**`( org.eclipse.zest.layouts.progress.ProgressListener` **listener** `)`

- *addRelationship*
  `public void` **addRelationship**`( org.eclipse.zest.layouts.LayoutRelationship` **relationship** `)`

- *applyLayout*
  `public void` **applyLayout**`( org.eclipse.zest.layouts.LayoutEntity []` **entitiesToLayout**`, org.eclipse.zest.layouts.LayoutRelationship []` **relationshipsToConsider**`, double` **x**`, double` **y**`, double` **width**`, double` **height**`, boolean` **asynchronous**`, boolean` **continuous** `)`

- *getEntityAspectRatio*
  `public double` **getEntityAspectRatio**`( )`

- *getStyle*
  `public int` **getStyle**`( )`

- *isRunning*
  `public boolean` **isRunning**`( )`

- *removeEntity*
  `public void` **removeEntity**`( org.eclipse.zest.layouts.LayoutEntity` **entity** `)`

- *removeProgressListener*
  `public void` **removeProgressListener**`( org.eclipse.zest.layouts.progress.ProgressListener` **listener** `)`

- *removeRelationship*
  `public void` **removeRelationship**`( org.eclipse.zest.layouts.LayoutRelationship` **relationship** `)`

- *removeRelationships*
  `public void` **removeRelationships**`( java.util.List` **relationships** `)`

- *setChildAlgorithm*
  `public void` **setChildAlgorithm**`( org.eclipse.zest.layouts.LayoutAlgorithm` **child** `)`

- *setComparator*
  `public void` **setComparator**`( java.util.Comparator` **comparator** `)`

- *setEntityAspectRatio*
  `public void` **setEntityAspectRatio**`( double` **ratio** `)`

- *setFilter*
  `public void` **setFilter**`( org.eclipse.zest.layouts.Filter` **filter** `)`
- *setStyle*
  `public void` **setStyle**`( int` **style** `)`
- *stop*
  `public void` **stop**`( )`

22.2.2 Class GUIAnchor

GUIAnchor acts as both a source and sink in a network to show what it connects to and what connects to it.

Declaration

public class GUIAnchor
extends org.eclipse.zest.core.widgets.GraphNode
implements NodeContainer

Constructors

- *GUIAnchor*
  `public` **GUIAnchor**`( boolean` **isSink**, `uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork` **network**, `org.eclipse.zest.core.widgets.IContainer` **graphModel**, `int` **style** `)`
  - **Usage**
    - Creates a GUI Anchor.
  - **Parameters**
    - `isSink` - It is a Sink Node if true, Source Node if false
    - `network` - Network to add Anchor to
    - `graphModel` - Graph to insert Anchor into
    - `style` - Style of Anchor

Methods

- *createFigureForModel*
  `protected IFigure` **createFigureForModel**`( )`
- *createToolTip*
  `public void` **createToolTip**`( )`
- *getNode*
  `public Node` **getNode**`( )`
- *highlight*
  `public void` **highlight**`( )`
  - **Usage**

  - ∗ Highlights the anchor node.

- *isSink*
  `public boolean` **isSink**`( )`

- *setNode*
  `public void` **setNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **network** `)`

- *unhighlight*
  `public void` **unhighlight**`( )`
  - **Usage**
    - ∗ Unhighlights the anchor node.

Methods inherited from class `org.eclipse.zest.core.widgets.GraphNode`

- *cacheLabel*
  `public boolean` **cacheLabel**`( )`
- *createFigureForModel*
  `protected IFigure` **createFigureForModel**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *fishEye*
  `protected IFigure` **fishEye**`( boolean` **arg0**`, boolean` **arg1** `)`
- *getBackgroundColor*
  `public Color` **getBackgroundColor**`( )`
- *getBorderColor*
  `public Color` **getBorderColor**`( )`
- *getBorderHighlightColor*
  `public Color` **getBorderHighlightColor**`( )`
- *getBorderWidth*
  `public int` **getBorderWidth**`( )`
- *getFont*
  `public Font` **getFont**`( )`
- *getForegroundColor*
  `public Color` **getForegroundColor**`( )`
- *getGraphModel*
  `public Graph` **getGraphModel**`( )`
- *getHighlightColor*
  `public Color` **getHighlightColor**`( )`
- *getItemType*
  `public int` **getItemType**`( )`
- *getLayoutEntity*
  `public LayoutEntity` **getLayoutEntity**`( )`
- *getLocation*
  `public Point` **getLocation**`( )`
- *getNodeFigure*
  `public IFigure` **getNodeFigure**`( )`
- *getNodeStyle*
  `public int` **getNodeStyle**`( )`

- *getSize*
  `public Dimension` **getSize**`( )`
- *getSourceConnections*
  `public List` **getSourceConnections**`( )`
- *getStyle*
  `public int` **getStyle**`( )`
- *getTargetConnections*
  `public List` **getTargetConnections**`( )`
- *getTooltip*
  `public IFigure` **getTooltip**`( )`
- *highlight*
  `public void` **highlight**`( )`
- *initFigure*
  `protected void` **initFigure**`( )`
- *initModel*
  `protected void` **initModel**`( org.eclipse.zest.core.widgets.IContainer` **arg0**`, java.lang.String` **arg1**`, org.eclipse.swt.graphics.Image` **arg2** `)`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isSelected*
  `public boolean` **isSelected**`( )`
- *isSizeFixed*
  `public boolean` **isSizeFixed**`( )`
- *isVisible*
  `public boolean` **isVisible**`( )`
- *refreshLocation*
  `protected void` **refreshLocation**`( )`
- *setBackgroundColor*
  `public void` **setBackgroundColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderColor*
  `public void` **setBorderColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderHighlightColor*
  `public void` **setBorderHighlightColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderWidth*
  `public void` **setBorderWidth**`( int` **arg0** `)`
- *setCacheLabel*
  `public void` **setCacheLabel**`( boolean` **arg0** `)`
- *setFont*
  `public void` **setFont**`( org.eclipse.swt.graphics.Font` **arg0** `)`
- *setForegroundColor*
  `public void` **setForegroundColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setHighlightColor*
  `public void` **setHighlightColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setImage*
  `public void` **setImage**`( org.eclipse.swt.graphics.Image` **arg0** `)`
- *setLocation*
  `public void` **setLocation**`( double` **arg0**`, double` **arg1** `)`
- *setNodeStyle*
  `public void` **setNodeStyle**`( int` **arg0** `)`
- *setSize*
  `public void` **setSize**`( double` **arg0**`, double` **arg1** `)`

- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`
- *setTooltip*
  `public void` **setTooltip**`( org.eclipse.draw2d.IFigure` **arg0** `)`
- *setVisible*
  `public void` **setVisible**`( boolean` **arg0** `)`
- *toString*
  `public String` **toString**`( )`
- *unhighlight*
  `public void` **unhighlight**`( )`
- *updateFigureForModel*
  `protected void` **updateFigureForModel**`( org.eclipse.draw2d.IFigure` **arg0** `)`

Methods inherited from class `org.eclipse.zest.core.widgets.GraphItem`

- *checkStyle*
  `protected boolean` **checkStyle**`( int` **arg0** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getGraphModel*
  `public abstract Graph` **getGraphModel**`( )`
- *getItemType*
  `public abstract int` **getItemType**`( )`
- *highlight*
  `public abstract void` **highlight**`( )`
- *isVisible*
  `public abstract boolean` **isVisible**`( )`
- *setVisible*
  `public abstract void` **setVisible**`( boolean` **arg0** `)`
- *unhighlight*
  `public abstract void` **unhighlight**`( )`

Methods inherited from class `org.eclipse.swt.widgets.Item`

- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *getImage*
  `public Image` **getImage**`( )`
- *getText*
  `public String` **getText**`( )`
- *setImage*
  `public void` **setImage**`( org.eclipse.swt.graphics.Image` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`

Methods inherited from class `org.eclipse.swt.widgets.Widget`

- *addDisposeListener*
  `public void` **addDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *addListener*
  `public void` **addListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *checkWidget*
  `protected void` **checkWidget**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *getData*
  `public Object` **getData**`( )`
- *getData*
  `public Object` **getData**`( java.lang.String` **arg0** `)`
- *getDisplay*
  `public Display` **getDisplay**`( )`
- *getListeners*
  `public Listener` **getListeners**`( int` **arg0** `)`
- *getStyle*
  `public int` **getStyle**`( )`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isListening*
  `public boolean` **isListening**`( int` **arg0** `)`
- *notifyListeners*
  `public void` **notifyListeners**`( int` **arg0**`, org.eclipse.swt.widgets.Event` **arg1** `)`
- *removeDisposeListener*
  `public void` **removeDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *removeListener*
  `public void` **removeListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *removeListener*
  `protected void` **removeListener**`( int` **arg0**`, org.eclipse.swt.internal.SWTEventListener` **arg1** `)`
- *setData*
  `public void` **setData**`( java.lang.Object` **arg0** `)`
- *setData*
  `public void` **setData**`( java.lang.String` **arg0**`, java.lang.Object` **arg1** `)`
- *toString*
  `public String` **toString**`( )`

22.2.3 Class **GUIBridge**

Connection between two GUI Networks containing links connecting nodes between each network

Declaration

public class GUIBridge
extends org.eclipse.zest.core.widgets.GraphConnection

Constructors

- *GUIBridge*
  `public` **GUIBridge**`( uk.ac.ic.doc.neuralnets.graph.neural.NetworkBridge` **bridge**, `org.eclipse.zest.core.widgets.Graph` **graphModel**, `int` **style**, `org.eclipse.zest.core.widgets.GraphNode` **source**, `org.eclipse.zest.core.widgets.GraphNode` **destination** `)`
  - **Usage**
    - Create GUI Bridge that connects two GUI Networks in the UI.
  - **Parameters**
    - `bridge` - Network Bridge between the neural networks
    - `graphModel` - Graph that the bridge is inserted into
    - `style` - Style of edge
    - `source` - Start point of bridge
    - `destination` - End point of bridge

Methods

- *createToolTip*
  `public void` **createToolTip**`( )`
- *getBridge*
  `public NetworkBridge` **getBridge**`( )`
- *setBridge*
  `public void` **setBridge**`( uk.ac.ic.doc.neuralnets.graph.neural.NetworkBridge` **bridge** `)`

Methods inherited from class `org.eclipse.zest.core.widgets.GraphConnection`

- *changeLineColor*
  `public void` **changeLineColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getConnectionFigure*
  `public Connection` **getConnectionFigure**`( )`
- *getConnectionStyle*
  `public int` **getConnectionStyle**`( )`
- *getDestination*
  `public GraphNode` **getDestination**`( )`
- *getExternalConnection*
  `public Object` **getExternalConnection**`( )`
- *getFont*
  `public Font` **getFont**`( )`
- *getGraphModel*
  `public Graph` **getGraphModel**`( )`
- *getHighlightColor*
  `public Color` **getHighlightColor**`( )`

- *getItemType*
  `public int` **getItemType**`( )`
- *getLayoutRelationship*
  `public LayoutRelationship` **getLayoutRelationship**`( )`
- *getLineColor*
  `public Color` **getLineColor**`( )`
- *getLineStyle*
  `public int` **getLineStyle**`( )`
- *getLineWidth*
  `public int` **getLineWidth**`( )`
- *getSource*
  `public GraphNode` **getSource**`( )`
- *getTooltip*
  `public IFigure` **getTooltip**`( )`
- *getWeightInLayout*
  `public double` **getWeightInLayout**`( )`
- *highlight*
  `public void` **highlight**`( )`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isHighlighted*
  `public boolean` **isHighlighted**`( )`
- *isVisible*
  `public boolean` **isVisible**`( )`
- *setConnectionStyle*
  `public void` **setConnectionStyle**`( int` **arg0** `)`
- *setFont*
  `public void` **setFont**`( org.eclipse.swt.graphics.Font` **arg0** `)`
- *setHighlightColor*
  `public void` **setHighlightColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setLineColor*
  `public void` **setLineColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setLineStyle*
  `public void` **setLineStyle**`( int` **arg0** `)`
- *setLineWidth*
  `public void` **setLineWidth**`( int` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`
- *setTooltip*
  `public void` **setTooltip**`( org.eclipse.draw2d.IFigure` **arg0** `)`
- *setVisible*
  `public void` **setVisible**`( boolean` **arg0** `)`
- *setWeight*
  `public void` **setWeight**`( double` **arg0** `)`
- *toString*
  `public String` **toString**`( )`
- *unhighlight*
  `public void` **unhighlight**`( )`

Methods inherited from class `org.eclipse.zest.core.widgets.GraphItem`

- *checkStyle*
  `protected boolean` **checkStyle**`( int` **arg0** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getGraphModel*
  `public abstract Graph` **getGraphModel**`( )`
- *getItemType*
  `public abstract int` **getItemType**`( )`
- *highlight*
  `public abstract void` **highlight**`( )`
- *isVisible*
  `public abstract boolean` **isVisible**`( )`
- *setVisible*
  `public abstract void` **setVisible**`( boolean` **arg0** `)`
- *unhighlight*
  `public abstract void` **unhighlight**`( )`

Methods inherited from class `org.eclipse.swt.widgets.Item`

- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *getImage*
  `public Image` **getImage**`( )`
- *getText*
  `public String` **getText**`( )`
- *setImage*
  `public void` **setImage**`( org.eclipse.swt.graphics.Image` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`

Methods inherited from class `org.eclipse.swt.widgets.Widget`

- *addDisposeListener*
  `public void` **addDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *addListener*
  `public void` **addListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *checkWidget*
  `protected void` **checkWidget**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *getData*
  `public Object` **getData**`( )`
- *getData*
  `public Object` **getData**`( java.lang.String` **arg0** `)`

- *getDisplay*
  `public Display` **getDisplay**`( )`
- *getListeners*
  `public Listener` **getListeners**`( int` **arg0** `)`
- *getStyle*
  `public int` **getStyle**`( )`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isListening*
  `public boolean` **isListening**`( int` **arg0** `)`
- *notifyListeners*
  `public void` **notifyListeners**`( int` **arg0**`, org.eclipse.swt.widgets.Event` **arg1** `)`
- *removeDisposeListener*
  `public void` **removeDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *removeListener*
  `public void` **removeListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *removeListener*
  `protected void` **removeListener**`( int` **arg0**`, org.eclipse.swt.internal.SWTEventListener` **arg1** `)`
- *setData*
  `public void` **setData**`( java.lang.Object` **arg0** `)`
- *setData*
  `public void` **setData**`( java.lang.String` **arg0**`, java.lang.Object` **arg1** `)`
- *toString*
  `public String` **toString**`( )`

22.2.4 Class **GUIEdge**

Represent a Synapse in the Zest graph.

Declaration

public class GUIEdge
extends org.eclipse.zest.core.widgets.GraphConnection

Constructors

- *GUIEdge*
  `public` **GUIEdge**`( uk.ac.ic.doc.neuralnets.graph.Edge` **edge**`, org.eclipse.zest.core.widgets.Graph` **graphModel**`, int` **style**`, org.eclipse.zest.core.widgets.GraphNode` **source**`, org.eclipse.zest.core.widgets.GraphNode` **destination** `)`
  - **Usage**
    - Creates a new edge in the specified graph for a Synapse. The edge decoration is set through the node specification, essentially ignoring the specified edge style.
  - **Parameters**
    - `edge` - - the synapse to represent.

- `graphModel` - - the graph into which to insert the edge
- `style` - - the style of the edge (see ZestStyles) - this is overridden
- `source` - - the start point of the edge.
- `destination` - - the end point of the edge.

Methods

- *createToolTip*
  `public void` **createToolTip**`( )`

- *getEdge*
  `public Edge` **getEdge**`( )`
  - **Usage**
    - Get the Synapse represented.
  - **Returns** - the synapse edge.

- *highlight*
  `public void` **highlight**`( )`
  - **Usage**
    - Unhighlight the edge

- *setEdge*
  `public void` **setEdge**`( uk.ac.ic.doc.neuralnets.graph.Edge` **edge** `)`
  - **Usage**
    - Set the Synapse represented.
  - **Parameters**
    - `edge` - - synapse to represent.

- *unhighlight*
  `public void` **unhighlight**`( )`
  - **Usage**
    - Highlight the edge.

Methods inherited from class `org.eclipse.zest.core.widgets.GraphConnection`

- *changeLineColor*
  `public void` **changeLineColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getConnectionFigure*
  `public Connection` **getConnectionFigure**`( )`
- *getConnectionStyle*
  `public int` **getConnectionStyle**`( )`
- *getDestination*
  `public GraphNode` **getDestination**`( )`

- *getExternalConnection*
  `public Object` **getExternalConnection**`( )`
- *getFont*
  `public Font` **getFont**`( )`
- *getGraphModel*
  `public Graph` **getGraphModel**`( )`
- *getHighlightColor*
  `public Color` **getHighlightColor**`( )`
- *getItemType*
  `public int` **getItemType**`( )`
- *getLayoutRelationship*
  `public LayoutRelationship` **getLayoutRelationship**`( )`
- *getLineColor*
  `public Color` **getLineColor**`( )`
- *getLineStyle*
  `public int` **getLineStyle**`( )`
- *getLineWidth*
  `public int` **getLineWidth**`( )`
- *getSource*
  `public GraphNode` **getSource**`( )`
- *getTooltip*
  `public IFigure` **getTooltip**`( )`
- *getWeightInLayout*
  `public double` **getWeightInLayout**`( )`
- *highlight*
  `public void` **highlight**`( )`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isHighlighted*
  `public boolean` **isHighlighted**`( )`
- *isVisible*
  `public boolean` **isVisible**`( )`
- *setConnectionStyle*
  `public void` **setConnectionStyle**`( int` **arg0** `)`
- *setFont*
  `public void` **setFont**`( org.eclipse.swt.graphics.Font` **arg0** `)`
- *setHighlightColor*
  `public void` **setHighlightColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setLineColor*
  `public void` **setLineColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setLineStyle*
  `public void` **setLineStyle**`( int` **arg0** `)`
- *setLineWidth*
  `public void` **setLineWidth**`( int` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`
- *setTooltip*
  `public void` **setTooltip**`( org.eclipse.draw2d.IFigure` **arg0** `)`
- *setVisible*
  `public void` **setVisible**`( boolean` **arg0** `)`
- *setWeight*
  `public void` **setWeight**`( double` **arg0** `)`
- *toString*
  `public String` **toString**`( )`
- *unhighlight*
  `public void` **unhighlight**`( )`

Methods inherited from class `org.eclipse.zest.core.widgets.GraphItem`

- *checkStyle*
  `protected boolean` **checkStyle**`( int` **arg0** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getGraphModel*
  `public abstract Graph` **getGraphModel**`( )`
- *getItemType*
  `public abstract int` **getItemType**`( )`
- *highlight*
  `public abstract void` **highlight**`( )`
- *isVisible*
  `public abstract boolean` **isVisible**`( )`
- *setVisible*
  `public abstract void` **setVisible**`( boolean` **arg0** `)`
- *unhighlight*
  `public abstract void` **unhighlight**`( )`

Methods inherited from class `org.eclipse.swt.widgets.Item`

- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *getImage*
  `public Image` **getImage**`( )`
- *getText*
  `public String` **getText**`( )`
- *setImage*
  `public void` **setImage**`( org.eclipse.swt.graphics.Image` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`

Methods inherited from class `org.eclipse.swt.widgets.Widget`

- *addDisposeListener*
  `public void` **addDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *addListener*
  `public void` **addListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *checkWidget*
  `protected void` **checkWidget**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *getData*
  `public Object` **getData**`( )`
- *getData*
  `public Object` **getData**`( java.lang.String` **arg0** `)`

- *getDisplay*
  public Display **getDisplay**( )
- *getListeners*
  public Listener **getListeners**(int **arg0**)
- *getStyle*
  public int **getStyle**( )
- *isDisposed*
  public boolean **isDisposed**( )
- *isListening*
  public boolean **isListening**(int **arg0**)
- *notifyListeners*
  public void **notifyListeners**(int **arg0**, org.eclipse.swt.widgets.Event **arg1**)
- *removeDisposeListener*
  public void **removeDisposeListener**(org.eclipse.swt.events.DisposeListener **arg0**)
- *removeListener*
  public void **removeListener**(int **arg0**, org.eclipse.swt.widgets.Listener **arg1**)
- *removeListener*
  protected void **removeListener**(int **arg0**, org.eclipse.swt.internal.SWTEventListener **arg1**)
- *setData*
  public void **setData**(java.lang.Object **arg0**)
- *setData*
  public void **setData**(java.lang.String **arg0**, java.lang.Object **arg1**)
- *toString*
  public String **toString**( )

22.2.5 Class **GUINetwork**

Declaration

public class GUINetwork
extends org.eclipse.zest.core.widgets.GraphContainer
implements NodeContainer

Constructors

- *GUINetwork*
  public **GUINetwork**(uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork **network**, org.eclipse.zest.core.widgets.IContainer **container**, org.eclipse.zest.core.widgets.Graph **g**, int **style**)
  - **Usage**
    - Creates a GUI Network which can contain more GUI Networks or GUI Nodes.
  - **Parameters**
    - `network` - Network to model in GUI
    - `container` - Graph to insert GUI Network into
    - `g` - Contents of network in a displayable format
    - `style` - Style of GUI Network

Methods

- *createToolTip*
  `public void` **createToolTip**`( )`
- *getNode*
  `public Node` **getNode**`( )`
- *persistLocation*
  `public void` **persistLocation**`( )`
  - **Usage**
    - Persists the location of this node in the GUI to the model node.
- *setNode*
  `public void` **setNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **network** `)`

Methods inherited from class `org.eclipse.zest.core.widgets.GraphContainer`

- *applyLayout*
  `public void` **applyLayout**`( )`
- *close*
  `public void` **close**`( boolean` **arg0** `)`
- *getGraph*
  `public Graph` **getGraph**`( )`
- *getItemType*
  `public int` **getItemType**`( )`
- *getNodeFigure*
  `public IFigure` **getNodeFigure**`( )`
- *getNodes*
  `public List` **getNodes**`( )`
- *getScale*
  `public double` **getScale**`( )`
- *initFigure*
  `protected void` **initFigure**`( )`
- *open*
  `public void` **open**`( boolean` **arg0** `)`
- *refreshLocation*
  `protected void` **refreshLocation**`( )`
- *setCustomFigure*
  `public void` **setCustomFigure**`( org.eclipse.draw2d.IFigure` **arg0** `)`
- *setLayoutAlgorithm*
  `public void` **setLayoutAlgorithm**`( org.eclipse.zest.layouts.LayoutAlgorithm` **arg0**`, boolean` **arg1** `)`
- *setScale*
  `public void` **setScale**`( double` **arg0** `)`
- *updateFigureForModel*
  `protected void` **updateFigureForModel**`( org.eclipse.draw2d.IFigure` **arg0** `)`

Methods inherited from class `org.eclipse.zest.core.widgets.GraphNode`

- *cacheLabel*
  `public boolean` **cacheLabel**`( )`
- *createFigureForModel*
  `protected IFigure` **createFigureForModel**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *fishEye*
  `protected IFigure` **fishEye**`( boolean` **arg0**`, boolean` **arg1** `)`
- *getBackgroundColor*
  `public Color` **getBackgroundColor**`( )`
- *getBorderColor*
  `public Color` **getBorderColor**`( )`
- *getBorderHighlightColor*
  `public Color` **getBorderHighlightColor**`( )`
- *getBorderWidth*
  `public int` **getBorderWidth**`( )`
- *getFont*
  `public Font` **getFont**`( )`
- *getForegroundColor*
  `public Color` **getForegroundColor**`( )`
- *getGraphModel*
  `public Graph` **getGraphModel**`( )`
- *getHighlightColor*
  `public Color` **getHighlightColor**`( )`
- *getItemType*
  `public int` **getItemType**`( )`
- *getLayoutEntity*
  `public LayoutEntity` **getLayoutEntity**`( )`
- *getLocation*
  `public Point` **getLocation**`( )`
- *getNodeFigure*
  `public IFigure` **getNodeFigure**`( )`
- *getNodeStyle*
  `public int` **getNodeStyle**`( )`
- *getSize*
  `public Dimension` **getSize**`( )`
- *getSourceConnections*
  `public List` **getSourceConnections**`( )`
- *getStyle*
  `public int` **getStyle**`( )`
- *getTargetConnections*
  `public List` **getTargetConnections**`( )`
- *getTooltip*
  `public IFigure` **getTooltip**`( )`
- *highlight*
  `public void` **highlight**`( )`
- *initFigure*
  `protected void` **initFigure**`( )`

- *initModel*
  `protected void` **initModel**`( org.eclipse.zest.core.widgets.IContainer` **arg0**`, java.lang.String` **arg1**`, org.eclipse.swt.graphics.Image` **arg2** `)`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isSelected*
  `public boolean` **isSelected**`( )`
- *isSizeFixed*
  `public boolean` **isSizeFixed**`( )`
- *isVisible*
  `public boolean` **isVisible**`( )`
- *refreshLocation*
  `protected void` **refreshLocation**`( )`
- *setBackgroundColor*
  `public void` **setBackgroundColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderColor*
  `public void` **setBorderColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderHighlightColor*
  `public void` **setBorderHighlightColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderWidth*
  `public void` **setBorderWidth**`( int` **arg0** `)`
- *setCacheLabel*
  `public void` **setCacheLabel**`( boolean` **arg0** `)`
- *setFont*
  `public void` **setFont**`( org.eclipse.swt.graphics.Font` **arg0** `)`
- *setForegroundColor*
  `public void` **setForegroundColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setHighlightColor*
  `public void` **setHighlightColor**`( org.eclipse.swt.graphics.Color` **arg0** `)`
- *setImage*
  `public void` **setImage**`( org.eclipse.swt.graphics.Image` **arg0** `)`
- *setLocation*
  `public void` **setLocation**`( double` **arg0**`, double` **arg1** `)`
- *setNodeStyle*
  `public void` **setNodeStyle**`( int` **arg0** `)`
- *setSize*
  `public void` **setSize**`( double` **arg0**`, double` **arg1** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`
- *setTooltip*
  `public void` **setTooltip**`( org.eclipse.draw2d.IFigure` **arg0** `)`
- *setVisible*
  `public void` **setVisible**`( boolean` **arg0** `)`
- *toString*
  `public String` **toString**`( )`
- *unhighlight*
  `public void` **unhighlight**`( )`
- *updateFigureForModel*
  `protected void` **updateFigureForModel**`( org.eclipse.draw2d.IFigure` **arg0** `)`

Methods inherited from class `org.eclipse.zest.core.widgets.GraphItem`

- *checkStyle*
  `protected boolean` **checkStyle**`(` `int` **arg0** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getGraphModel*
  `public abstract Graph` **getGraphModel**`( )`
- *getItemType*
  `public abstract int` **getItemType**`( )`
- *highlight*
  `public abstract void` **highlight**`( )`
- *isVisible*
  `public abstract boolean` **isVisible**`( )`
- *setVisible*
  `public abstract void` **setVisible**`(` `boolean` **arg0** `)`
- *unhighlight*
  `public abstract void` **unhighlight**`( )`

Methods inherited from class `org.eclipse.swt.widgets.Item`

- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *getImage*
  `public Image` **getImage**`( )`
- *getText*
  `public String` **getText**`( )`
- *setImage*
  `public void` **setImage**`(` `org.eclipse.swt.graphics.Image` **arg0** `)`
- *setText*
  `public void` **setText**`(` `java.lang.String` **arg0** `)`

Methods inherited from class `org.eclipse.swt.widgets.Widget`

- *addDisposeListener*
  `public void` **addDisposeListener**`(` `org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *addListener*
  `public void` **addListener**`(` `int` **arg0**, `org.eclipse.swt.widgets.Listener` **arg1** `)`
- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *checkWidget*
  `protected void` **checkWidget**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *getData*
  `public Object` **getData**`( )`
- *getData*
  `public Object` **getData**`(` `java.lang.String` **arg0** `)`

- *getDisplay*
  `public Display` **getDisplay**`( )`
- *getListeners*
  `public Listener` **getListeners**`( int` **arg0** `)`
- *getStyle*
  `public int` **getStyle**`( )`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isListening*
  `public boolean` **isListening**`( int` **arg0** `)`
- *notifyListeners*
  `public void` **notifyListeners**`( int` **arg0**`, org.eclipse.swt.widgets.Event` **arg1** `)`
- *removeDisposeListener*
  `public void` **removeDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *removeListener*
  `public void` **removeListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *removeListener*
  `protected void` **removeListener**`( int` **arg0**`, org.eclipse.swt.internal.SWTEventListener` **arg1** `)`
- *setData*
  `public void` **setData**`( java.lang.Object` **arg0** `)`
- *setData*
  `public void` **setData**`( java.lang.String` **arg0**`, java.lang.Object` **arg1** `)`
- *toString*
  `public String` **toString**`( )`

22.2.6 Class **GUINode**

Represents a Neurone in the Zest graph.

Declaration

public class GUINode
extends org.eclipse.zest.core.widgets.GraphNode
implements NodeContainer

Constructors

- *GUINode*
  `public` **GUINode**`( org.eclipse.zest.core.widgets.IContainer` **graphModel**`, int` **style** `)`
- *GUINode*
  `public` **GUINode**`( uk.ac.ic.doc.neuralnets.graph.Node` **node**`, org.eclipse.zest.core.widgets.IContainer` **graphModel**`, int` **style** `)`

Methods

- *createFigureForModel*
  `protected IFigure` **createFigureForModel**`( )`

- *createToolTip*
  `public void` **createToolTip**`( )`

- *getNode*
  `public Node` **getNode**`( )`

- *highlight*
  `public void` **highlight**`( )`
  - **Usage**
    - Highlights the node.

- *persistLocation*
  `public void` **persistLocation**`( )`
  - **Usage**
    - Persists the location of this node in the GUI to the model node.

- *setNode*
  `public void` **setNode**`( uk.ac.ic.doc.neuralnets.graph.Node` **node** `)`

- *setOverlayColor*
  `public void` **setOverlayColor**`( org.eclipse.swt.graphics.Color` **c** `)`
  - **Usage**
    - Change the background color of the charge overlay to the specified color.
  - **Parameters**
    - `c` - - the new overlay color.

- *unhighlight*
  `public void` **unhighlight**`( )`
  - **Usage**
    - Unhightlights the node.

- *updateChargeOverlay*
  `public void` **updateChargeOverlay**`( )`
  - **Usage**
    - Update the size of the charge overlay. Should be called when the model node ticks.

Methods inherited from class `org.eclipse.zest.core.widgets.GraphNode`

- *cacheLabel*
  `public boolean` **cacheLabel**`( )`
- *createFigureForModel*
  `protected IFigure` **createFigureForModel**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *fishEye*
  `protected IFigure` **fishEye**`( boolean` **arg0**`, boolean` **arg1** `)`
- *getBackgroundColor*
  `public Color` **getBackgroundColor**`( )`
- *getBorderColor*
  `public Color` **getBorderColor**`( )`
- *getBorderHighlightColor*
  `public Color` **getBorderHighlightColor**`( )`
- *getBorderWidth*
  `public int` **getBorderWidth**`( )`
- *getFont*
  `public Font` **getFont**`( )`
- *getForegroundColor*
  `public Color` **getForegroundColor**`( )`
- *getGraphModel*
  `public Graph` **getGraphModel**`( )`
- *getHighlightColor*
  `public Color` **getHighlightColor**`( )`
- *getItemType*
  `public int` **getItemType**`( )`
- *getLayoutEntity*
  `public LayoutEntity` **getLayoutEntity**`( )`
- *getLocation*
  `public Point` **getLocation**`( )`
- *getNodeFigure*
  `public IFigure` **getNodeFigure**`( )`
- *getNodeStyle*
  `public int` **getNodeStyle**`( )`
- *getSize*
  `public Dimension` **getSize**`( )`
- *getSourceConnections*
  `public List` **getSourceConnections**`( )`
- *getStyle*
  `public int` **getStyle**`( )`
- *getTargetConnections*
  `public List` **getTargetConnections**`( )`
- *getTooltip*
  `public IFigure` **getTooltip**`( )`
- *highlight*
  `public void` **highlight**`( )`
- *initFigure*
  `protected void` **initFigure**`( )`

- *initModel*
  `protected void` **initModel**`(` `org.eclipse.zest.core.widgets.IContainer` **arg0**, `java.lang.String` **arg1**, `org.eclipse.swt.graphics.Image` **arg2** `)`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isSelected*
  `public boolean` **isSelected**`( )`
- *isSizeFixed*
  `public boolean` **isSizeFixed**`( )`
- *isVisible*
  `public boolean` **isVisible**`( )`
- *refreshLocation*
  `protected void` **refreshLocation**`( )`
- *setBackgroundColor*
  `public void` **setBackgroundColor**`(` `org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderColor*
  `public void` **setBorderColor**`(` `org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderHighlightColor*
  `public void` **setBorderHighlightColor**`(` `org.eclipse.swt.graphics.Color` **arg0** `)`
- *setBorderWidth*
  `public void` **setBorderWidth**`(` `int` **arg0** `)`
- *setCacheLabel*
  `public void` **setCacheLabel**`(` `boolean` **arg0** `)`
- *setFont*
  `public void` **setFont**`(` `org.eclipse.swt.graphics.Font` **arg0** `)`
- *setForegroundColor*
  `public void` **setForegroundColor**`(` `org.eclipse.swt.graphics.Color` **arg0** `)`
- *setHighlightColor*
  `public void` **setHighlightColor**`(` `org.eclipse.swt.graphics.Color` **arg0** `)`
- *setImage*
  `public void` **setImage**`(` `org.eclipse.swt.graphics.Image` **arg0** `)`
- *setLocation*
  `public void` **setLocation**`(` `double` **arg0**, `double` **arg1** `)`
- *setNodeStyle*
  `public void` **setNodeStyle**`(` `int` **arg0** `)`
- *setSize*
  `public void` **setSize**`(` `double` **arg0**, `double` **arg1** `)`
- *setText*
  `public void` **setText**`(` `java.lang.String` **arg0** `)`
- *setTooltip*
  `public void` **setTooltip**`(` `org.eclipse.draw2d.IFigure` **arg0** `)`
- *setVisible*
  `public void` **setVisible**`(` `boolean` **arg0** `)`
- *toString*
  `public String` **toString**`( )`
- *unhighlight*
  `public void` **unhighlight**`( )`
- *updateFigureForModel*
  `protected void` **updateFigureForModel**`(` `org.eclipse.draw2d.IFigure` **arg0** `)`

METHODS INHERITED FROM CLASS `org.eclipse.zest.core.widgets.GraphItem`

- *checkStyle*
  `protected boolean` **checkStyle**`( int` **arg0** `)`
- *dispose*
  `public void` **dispose**`( )`
- *getGraphModel*
  `public abstract Graph` **getGraphModel**`( )`
- *getItemType*
  `public abstract int` **getItemType**`( )`
- *highlight*
  `public abstract void` **highlight**`( )`
- *isVisible*
  `public abstract boolean` **isVisible**`( )`
- *setVisible*
  `public abstract void` **setVisible**`( boolean` **arg0** `)`
- *unhighlight*
  `public abstract void` **unhighlight**`( )`

METHODS INHERITED FROM CLASS `org.eclipse.swt.widgets.Item`

- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *getImage*
  `public Image` **getImage**`( )`
- *getText*
  `public String` **getText**`( )`
- *setImage*
  `public void` **setImage**`( org.eclipse.swt.graphics.Image` **arg0** `)`
- *setText*
  `public void` **setText**`( java.lang.String` **arg0** `)`

METHODS INHERITED FROM CLASS `org.eclipse.swt.widgets.Widget`

- *addDisposeListener*
  `public void` **addDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *addListener*
  `public void` **addListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *checkSubclass*
  `protected void` **checkSubclass**`( )`
- *checkWidget*
  `protected void` **checkWidget**`( )`
- *dispose*
  `public void` **dispose**`( )`
- *getData*
  `public Object` **getData**`( )`
- *getData*
  `public Object` **getData**`( java.lang.String` **arg0** `)`

- *getDisplay*
  `public Display` **getDisplay**`( )`
- *getListeners*
  `public Listener` **getListeners**`( int` **arg0** `)`
- *getStyle*
  `public int` **getStyle**`( )`
- *isDisposed*
  `public boolean` **isDisposed**`( )`
- *isListening*
  `public boolean` **isListening**`( int` **arg0** `)`
- *notifyListeners*
  `public void` **notifyListeners**`( int` **arg0**`, org.eclipse.swt.widgets.Event` **arg1** `)`
- *removeDisposeListener*
  `public void` **removeDisposeListener**`( org.eclipse.swt.events.DisposeListener` **arg0** `)`
- *removeListener*
  `public void` **removeListener**`( int` **arg0**`, org.eclipse.swt.widgets.Listener` **arg1** `)`
- *removeListener*
  `protected void` **removeListener**`( int` **arg0**`, org.eclipse.swt.internal.SWTEventListener` **arg1** `)`
- *setData*
  `public void` **setData**`( java.lang.Object` **arg0** `)`
- *setData*
  `public void` **setData**`( java.lang.String` **arg0**`, java.lang.Object` **arg1** `)`
- *toString*
  `public String` **toString**`( )`

Chapter 23

Package uk.ac.ic.doc.neuralnets.gui.listeners

Package Contents *Page*

Classes

23.1 Classes

23.1.1 Class **ContinueQuestion**

Prompts the user for to confirm continuing with an action

Declaration

public class ContinueQuestion
extends java.lang.Object

Constructors

- *ContinueQuestion*
  `public` **ContinueQuestion**`( )`

Methods

- *ask*
  `public static boolean` **ask**`( org.eclipse.swt.widgets.Shell` **parent** `)`
  - **Usage**
    - Ask a question with the standard description: ”All unsaved changes will be lost!”.
  - **Parameters**
    - `parent` - - root shell
  - **Returns** - true to continue, false otherwise

- *ask*
  `public static boolean` **ask**`( org.eclipse.swt.widgets.Shell` **parent**`,`
  `java.lang.String` **desc** `)`
  - **Usage**
    - Ask a continue question of the user.
  - **Parameters**
    - `parent` - - root shell
    - `desc` - - question description
  - **Returns** - true to continue, false otherwise

APPENDIX C: Persistence Examples

```xml
<?xml version="1.0" encoding="UTF-8"?>
<networkml xmlns="http://morphml.org/networkml/schema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:meta="http://morphml.org/metadata/schema"
  xsi:schemaLocation="http://morphml.org/networkml/schema/Schemata/v1.7.3/Level3/NetworkML_v1.7.3.xsd"
  lengthUnits="micron">
  <meta:notes>Produced by ANNE: the Artifical Neural Network Editor</meta:notes>
  <populations>
    <population name="14">
      <instances size="2">
        <instance id="16">
          <meta:properties>
            <meta:property tag="instance_type" value="uk.ac.ic.doc.neuralnets.graph.neural.SpikingNeurone" />
            <meta:property tag="postSpikeReset" value="-56.17718294810968" />
            <meta:property tag="z" value="0" />
            <meta:property tag="recoverySensitivity" value="0.2" />
            <meta:property tag="recoveryScale" value="0.02" />
            <meta:property tag="x" value="73" />
            <meta:property tag="trigger" value="30.0" />
            <meta:property tag="charge" value="-65.0" />
            <meta:property tag="y" value="26" />
            <meta:property tag="pSRRecovery" value="4.0027868923975625" />
          </meta:properties>
          <location x="73" y="26" z="0" />
        </instance>
        <instance id="15">
          <meta:properties>
            <meta:property tag="instance_type" value="uk.ac.ic.doc.neuralnets.graph.neural.SpikingNeurone" />
            <meta:property tag="postSpikeReset" value="-58.618716832163614" />
            <meta:property tag="z" value="0" />
            <meta:property tag="recoverySensitivity" value="0.2" />
            <meta:property tag="recoveryScale" value="0.02" />
            <meta:property tag="x" value="456" />
            <meta:property tag="trigger" value="30.0" />
            <meta:property tag="charge" value="-65.0" />
            <meta:property tag="y" value="292" />
            <meta:property tag="pSRRecovery" value="4.551832219630733" />
          </meta:properties>
          <location x="456" y="292" z="0" />
        </instance>
      </instances>
    </population>
  </populations>
  <projections units="Physiological Units">
    <projection name="Network-Synapses" source="14" target="14">
      <synapse_props synapse_type="uk.ac.ic.doc.neuralnets.graph.neural.Synapse" />
      <connections size="4">
        <connection id="17" pre_cell_id="16" post_cell_id="16">
          <properties weight="0.8604082982707334">
            <meta:properties>
              <meta:property tag="instance_type" value="uk.ac.ic.doc.neuralnets.graph.neural.Synapse" />
            </meta:properties>
          </properties>
        </connection>
        <connection id="20" pre_cell_id="15" post_cell_id="15">
          <properties weight="0.5024901465406223">
            <meta:properties>
              <meta:property tag="instance_type" value="uk.ac.ic.doc.neuralnets.graph.neural.Synapse" />
            </meta:properties>
          </properties>
        </connection>
        <connection id="18" pre_cell_id="16" post_cell_id="15">
          <properties weight="0.270062017417825">
            <meta:properties>
              <meta:property tag="instance_type" value="uk.ac.ic.doc.neuralnets.graph.neural.Synapse" />
            </meta:properties>
          </properties>
        </connection>
        <connection id="19" pre_cell_id="15" post_cell_id="16">
          <properties weight="0.13243622803794775">
            <meta:properties>
              <meta:property tag="instance_type" value="uk.ac.ic.doc.neuralnets.graph.neural.Synapse" />
            </meta:properties>
          </properties>
        </connection>
      </connections>
    </projection>
  </projections>
</networkml>
```

Code 1: Example network in NeuroML (XML) format.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Immersive.." version="2.0">
  <Scene>
    <Background skyColor="0.6 0.7 0.9"/>
    <Viewpoint description="Down z axis, 500 microns away" position="0 0 500"/>
    <Viewpoint description="Down z axis, 200 microns away" position="0 0 200"/>
    <Viewpoint description="Down z axis, 2mm away" position="0 0 2000"/>
    <Transform rotation="0 0 1 -1.570795">
      <Shape>
        <Appearance>
          <Material diffuseColor="0 1 0"/>
        </Appearance>
        <Cylinder height="200" radius="0.5"/>
      </Shape>
      <Transform translation="0 105 0">
        <Shape>
          <Appearance>
            <Material diffuseColor="0 1 0"/>
          </Appearance>
          <Cone height="10" bottomRadius="1"/>
        </Shape>
      </Transform>
    </Transform>
    <Transform>
      <Shape>
        <Appearance>
          <Material diffuseColor="1 1 0"/>
        </Appearance>
        <Cylinder height="200" radius="0.5"/>
      </Shape>
      <Transform translation="0 105 0">
        <Shape>
          <Appearance>
            <Material diffuseColor="1 1 0"/>
          </Appearance>
        <Cone height="10" bottomRadius="1"/>
        </Shape>
      </Transform>
    </Transform>
    <Transform rotation="1 0 0 1.570795">
      <Shape>
        <Appearance>
          <Material diffuseColor="1 0 0"/>
        </Appearance>
        <Cylinder height="200" radius="0.5"/>
      </Shape>
      <Transform translation="0 105 0">
        <Shape>
          <Appearance>
            <Material diffuseColor="1 0 0"/>
          </Appearance>
          <Cone height="10" bottomRadius="1"/>
        </Shape>
      </Transform>
    </Transform>
    <Transform translation="73  26  0">
      <Shape>
        <Appearance>
          <Material diffuseColor="0 1 0"/>
        </Appearance>
        <Sphere radius="5"/>
      </Shape>
    </Transform>
    <Transform translation="456  292  0">
      <Shape>
        <Appearance>
          <Material diffuseColor="0 1 0"/>
        </Appearance>
        <Sphere radius="5"/>
      </Shape>
    </Transform>
    <!--Projection Network-Synapses between 14 and 14-->
    <Transform>
      <Shape>
        <Appearance>
          <Material/>
        </Appearance>
        <LineSet vertexCount="2">
          <Coordinate point="73  26  0, 73  26  0"/>
          <Color color="0 1 0, 1 0 0"/>
        </LineSet>
      </Shape>
    </Transform>
    <Transform>
      <Shape>
        <Appearance>
          <Material/>
        </Appearance>
        <LineSet vertexCount="2">
          <Coordinate point="456  292  0, 456  292  0"/>
          <Color color="0 1 0, 1 0 0"/>
        </LineSet>
      </Shape>
    </Transform>
    <Transform>
      <Shape>
        <Appearance>
          <Material/>
        </Appearance>
        <LineSet vertexCount="2">
          <Coordinate point="73  26  0, 456  292  0"/>
          <Color color="0 1 0, 1 0 0"/>
        </LineSet>
      </Shape>
    </Transform>
    <Transform>
      <Shape>
        <Appearance>
          <Material/>
        </Appearance>
        <LineSet vertexCount="2">
          <Coordinate point="456  292  0, 73  26  0"/>
          <Color color="0 1 0, 1 0 0"/>
        </LineSet>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

Code 2: Example network in X3D (XML) format.

```
0,uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork,@@,
1,uk.ac.ic.doc.neuralnets.graph.neural.NeuralNetwork,@@,0
3,uk.ac.ic.doc.neuralnets.graph.neural.SpikingNeurone,@@postSpikeReset=-
   64.81116438247822@@z=0@@x=73@@recoverySensitivity=0.2@@recoveryScale=0.02@@trigger=30.0@@charge=-
   65.0@@pSRRecovery=7.850554546968316@@y=26@@,1
2,uk.ac.ic.doc.neuralnets.graph.neural.SpikingNeurone,@@postSpikeReset=-
   58.609280417269574@@z=0@@x=456@@recoverySensitivity=0.2@@recoveryScale=0.02@@trigger=30.0@@charge=-
   65.0@@pSRRecovery=5.045980335180888@@y=292@@,1
4,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,1
4,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,3:3:-1
6,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,1
6,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,2:3:-1
7,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,1
7,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,2:2:-1
5,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,1
5,uk.ac.ic.doc.neuralnets.graph.neural.Synapse,@@,3:2:-1
```

Code 3: Example network in Text Network Serializer format.

Appendix D: UML Diagrams

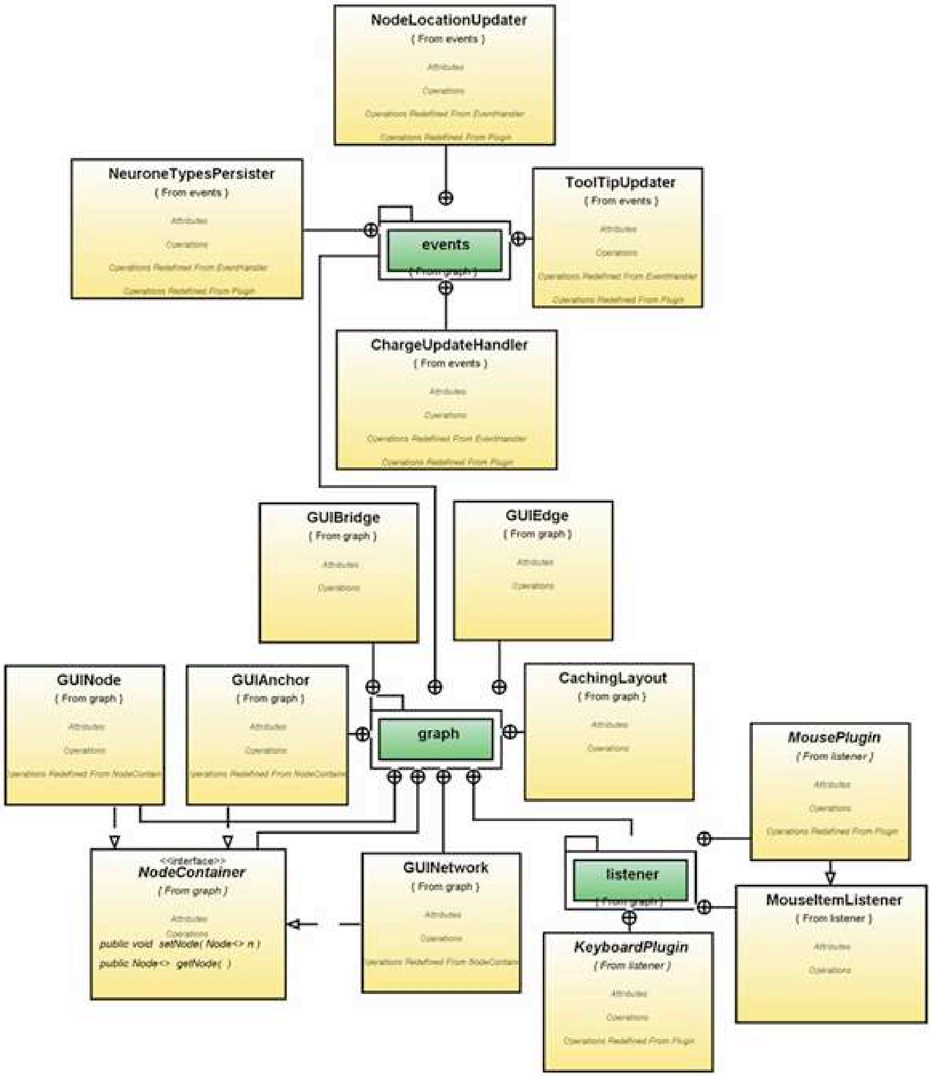


Figure 1: *GUI* Classes UML Diagram

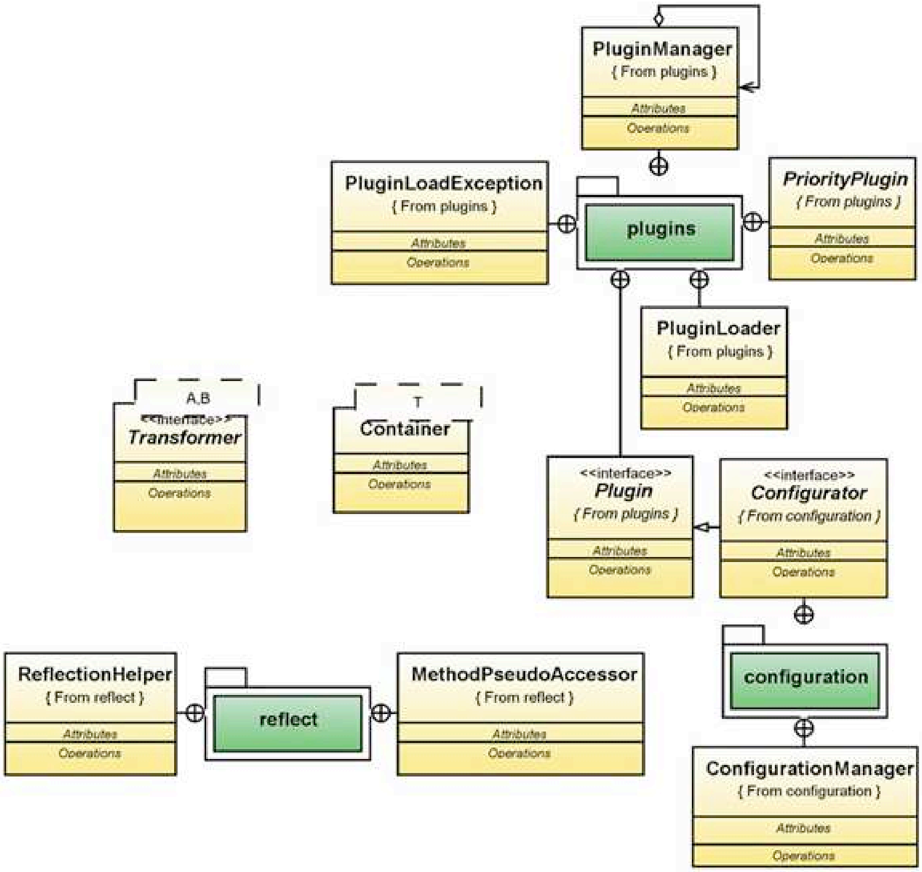


Figure 2: *Util* Classes UML Diagram

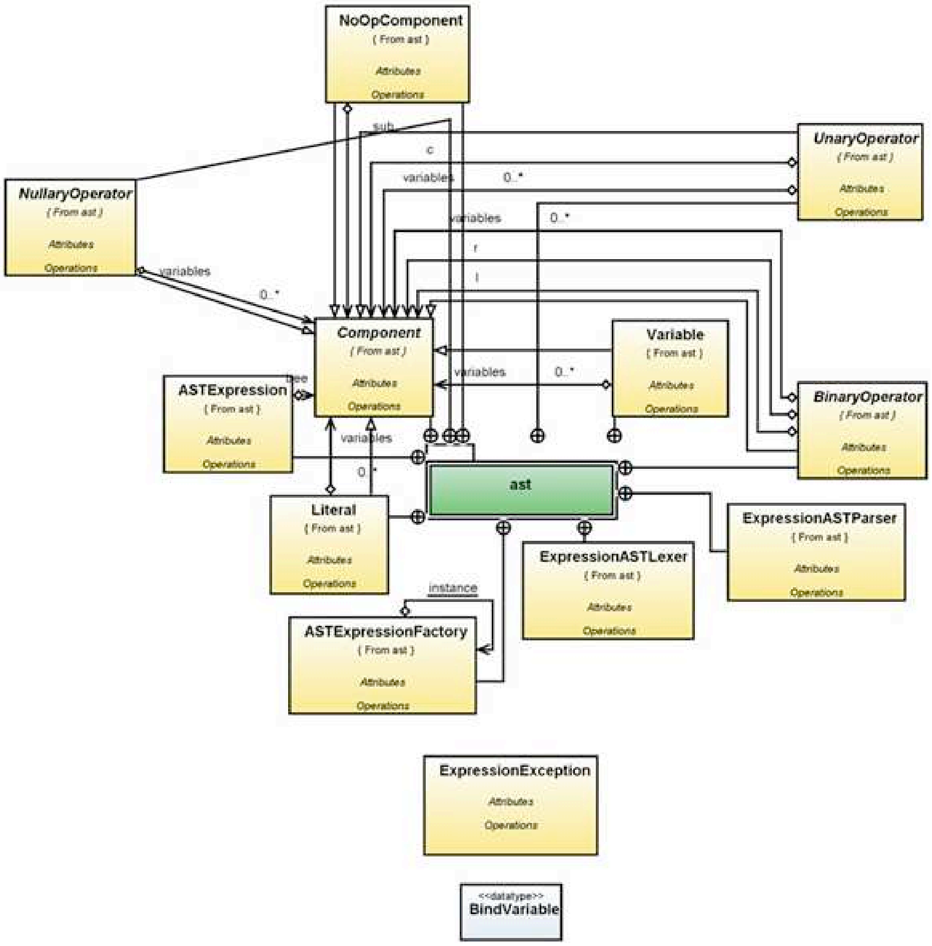


Figure 3: *ASTExpression* Classes UML Diagram

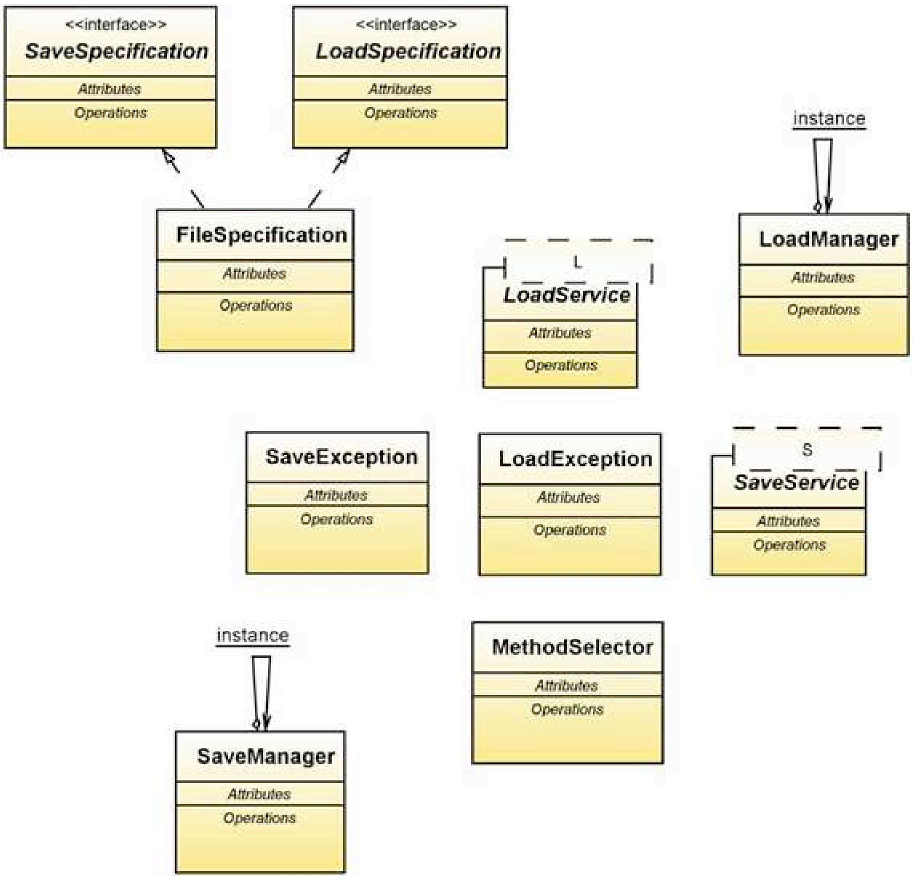


Figure 4: *Persistence* Classes UML Diagram