

Neuron

INHERITS FROM	Object
DECLARED IN	Neuron.h

CLASS DESCRIPTION

The Neuron class implements the fundamental unit of neural-network computation. The Neuron can use various activation functions. The following Neuron types are defined in Neuron.h: *Binary, Sigmoid, Sign, Tanh*. The Neuron class also provides for stochastic networks by storing the current temperature and modifying the output of the Neuron accordingly.

INSTANCE VARIABLES

<i>Inherited from Object</i>	Class	isa;
<i>Declared in Neuron</i>	id	inputs;
	double	lastOutput;
	id	random;
	struct {	
	id	source;
	double	weight;
	connection	*next;
	} connection;	

double	T;	int	nodeType;
inputs			a Storage object containing a list of connections
lastOutput			the output of this Neuron based on the inputs
random			a Random instance (random number generator)
connection			a struct that contains the source and weight of an input connection
nodeType			the type of activation function being used (Binary, Sigmoid, Sign, Tanh). the default is Sigmoid.
T			the current temperature

METHOD TYPES

Initializing a new instance	- init
Creating connections	± connect: ± connect:withWeight:
Setting parameters	± setWeightFor:to: ± setOutput: ± setType: ± setTemp: ± setRandom: ± setSymmetric:
Getting parameters	± getWeightFor: ± getType ± getTemp ± getSymmetric

	\pm lastOutput
	\pm inputs
Generating new output	\pm step
	\pm activation:

INSTANCE METHODS

activation:

- (double)**activation:(double)***net*

Returns the value for the activation function of the receiver with input *net*. Does not change the output of the neuron.

See also: - **step**, \pm **lastOutput**

connect:

- **connect:***anObject*

Creates a connection from the senders output to the receivers input with a random weight value between 0.0 and 0.01.

See also: - **connect:withWeight:**

connect:withWeight:

- **connect:***anObject* **withWeight:(double)***weight*

Creates a connection from the senders output to the receivers input with the weight value specified by *weight*.

See also: - **connect:**

changeWeightFor:by:

- **changeWeightFor:***anObject* **by:**(double)*delta*

Changes the weight of the connection from the senders output to the receivers input by *delta*.

See also: - **connect:withWeight:**, \pm **setWeightFor:to:**, \pm **getWeightFor:**

getTemp

- (double)**getTemp**

Returns the current temperature of the neuron. The default temperature is 0.0.

See also: - **setTemp:**

getType

- (int)**getType**

Returns the type of activation function applied to this neurons inputs. Valid types are Binary, Sigmoid, Sign and Tanh. The default type is Sigmoid.

See also: - **setType:**

getSymmetric

- (BOOL)**getWeightFor**

Returns the symmetric status of the receiver.

See also: - **setSymmetric:**

getWeightFor:

- (double)**getWeightFor:***anObject*

Returns the weight value for the receivers input connection from *anObject*. Returns **NAN** if *anObject* isn't in the input list of the receiver.

See also: - **setWeightFor:to:, ± connect:withWeight:**

init

- **init**

Initializes the receiver, a new Neuron object.

inputs

- **inputs**

Returns a pointer to the Storage object that contains the receivers input connections and associated weights.

See also: - **connect:, - connect:withWeight:**

lastOutput

- (double)**lastOutput**

Returns the value of the **lastOutput** instance variable.

See also: - **setOutput:, ± step**

setOutput:

- **setOutput:**(double)*weight*

Sets the instance variable **lastOutput** of the receiver to *weight*. This is used to set the output values of the input Neurons, for instance.

See also: - **lastOutput**

setRandom:

- **setRandom:***theRandom*

Sets the the instance variable *random* to point to *theRandom*. This should be used for large nets in order to have only one instance of the Random class. This also assures the same seed for all Neuron instances. If *random* == nil when **connect:** or **connect:withWeight:** are called, a new instance is created.

See also:

setSymmetric:

- **setRandom:**(BOOL)*isSymmetric*

Controls whether a Neuron tries to update the reverse connection when it's **changeWeightFor:by:** method is called. Default is NO.

See also: \pm **changeWeightFor:by:**

setTemp:

- **setType:**(double)*newTemp*

Sets the current temperature to *newTemp*. The default temperature is 0.0.

See also: - (double)**getTemp**

setType:

- **setType:**(int)*type*

Sets the activation function to type *type*. Valid types are: Binary, Sigmoid, Sign and Tanh. The default type is Sigmoid.

See also: - **getType**

setWeightFor:to:

- **setWeightFor:***anObject* **to:**(double)*weight*

Sets the weight for the connection from the output of *anObject* to the receivers input to *weight*.

See also: - **getWeightFor:**, - **connect:withWeight:**

step

- **step**

This method causes the receiver to generate a new value for **lastOutput** by applying it's activation function to the weighted sum of it's inputs.