

This class library is intended to use for research with different neural networks and training rules.

The class library is designed so that all types of network (i.e. single layer feed forward, multilayer feed forward, recurrent network with self feedback) can be implemented without major coding. Also these networks can be trained with almost any training technique.

The base classes are:

* Neural Network
* Layer
* Neuron
* Synapsis
* ActivationFunction
* LearningAlgorithm

For training purpose by different training rules one have to inherit the **LearningAlgorithm** class to implement the trainer. Add or override necessary methods of the base class and define the procedure of the training method.

**Activation functions:**

Activation functions are implemented on layer level, each layer can use different activation functions, and even input layer can have an activation function. There are eight types of activation functions defined in this class library: **UnipolarSignFunction**, **BipolarSignFunction**, **UnipolarLinearFunction**, **BipolarLinearFunction**, **SigmoidalUnipolarFunction**, **SigmoidalBipolarFunction**, **TangentHyperbolicFunction and GaussianActivationFunction** (Radial Basis Function), each of them inherited from the **ActivationFunction** base class. The users can define new activation function by inheriting from **ActivationFunction** class and the **GetOutput()** and **GetOutputDerivative()** functions must be overridden.

**Input combination functions:**

The users can also specify how the inputs of the neurons of a layer are combined to produce the net input. Three types are defined in this class library as **LinearProduct**, **EucledianDistance** and **ManhattanDistance.**

**Iftekhar Mahmud Towhid**

**Programmer, TLABS**