KARMAA — The Concept

KARMAA (Knowledge Acquisition Retention Management Assimilation and Application) was conceived to provide knowledge solutions to industry by using cutting edge technology available in engineering and science. KARMAA is a suite of algorithms available for a number of applications. The suite includes proven algorithms from Computer Science, Artificial Intelligence, Machine Learning, Neural Networks, Fuzzy Systems and Electrical Engineering. A consequence of more than a decade of continuous research and development at one of India's premier institute, KARMAA Technologies brings together the bunch of techniques that you would need.

KARMAA is a Knowledge Solutions provider that mines information from your databases using high-end techniques. Most of the present areas have a plenty of data and the important issue is to use them so that the best and most useful information can be arrived on. We provide the industry with useful methods and a powerful modeling environment for catering to your needs without having to understand the back-end technology.

We have been associated with Indian Space Research Organization, Bhabha Atomic Research Centre, Engineers India Limited, Indian Oil Corporation Limited, Steel Authority of India Limited, Aptech, CMC, UPERC, CRISIL, Deloitte Touche Tohmatsu, Seedata Technologies and IDFC.

Committed to stronger relations between industry and academia, KARMAA Technologies is presently interested in providing services to industry and R&D Institutions in the following areas:

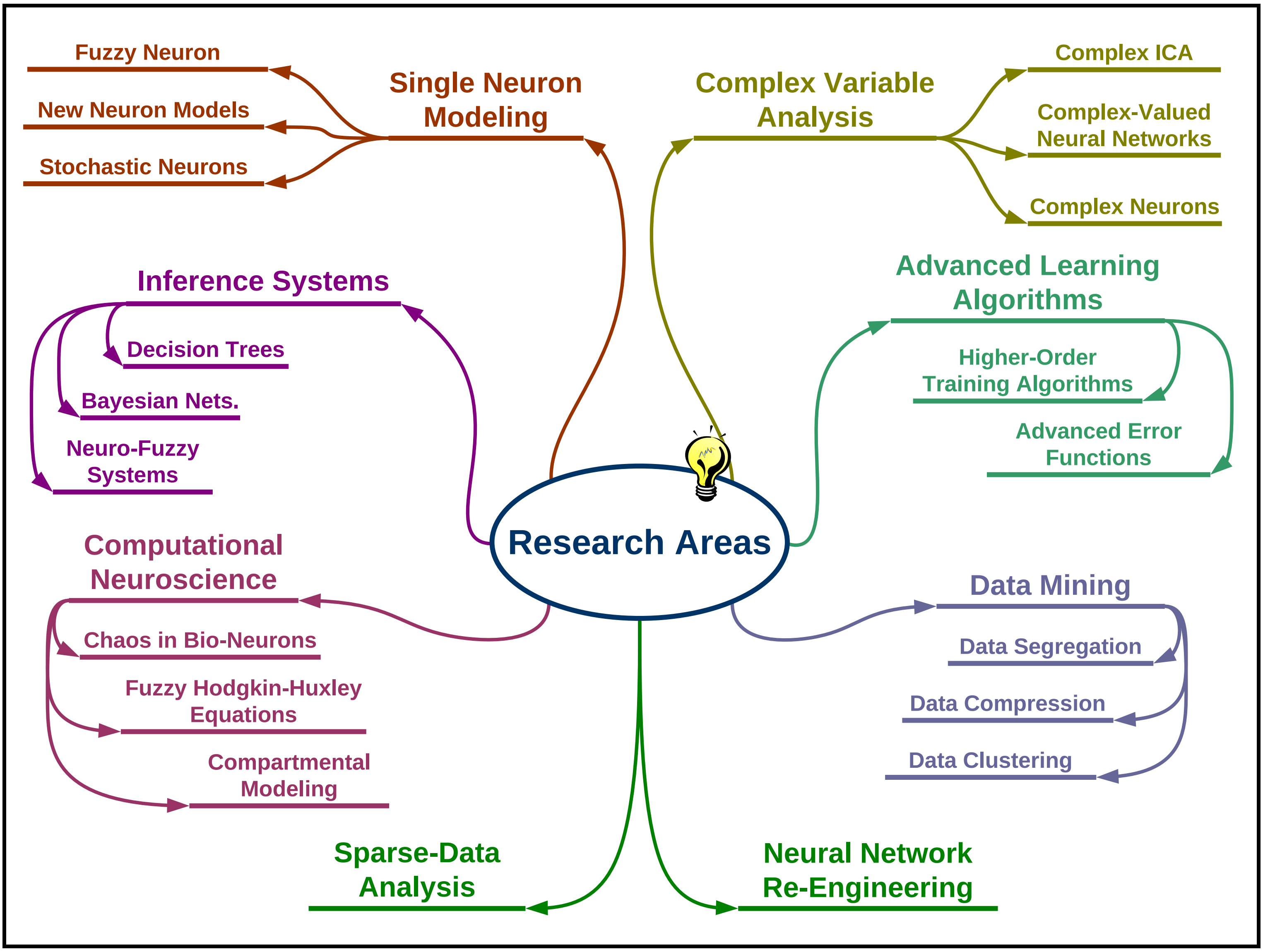
- Biometrics & Security Systems
- Finance & Stock Trading
- Drug Research & Bioinformatics
- Space & Defence Research
- Power Systems
- Critical Fault Detection Technology
- Data Centers
- Manufacturing and Industry Automation

Technology & Competence

KARMAA provides a general purpose as well as tailored application-specific software for Data Mining and Knowledge Engineering. Our core competence lies in

- Artificial Neural Networks
- Data Clustering
- Fuzzy Systems
- Support Vector Machines
- Bayesian Nets & Decision Trees
- Non-parametric modeling methods
- Global Optimization tools
- Statistical parametric measures

KARMAA provides an easy to use interface that makes it possible for those unaware of technical details to successfully use the package for their problems. We believe in building powerful solutions through continuous research and technology enhancements.



Unique Features

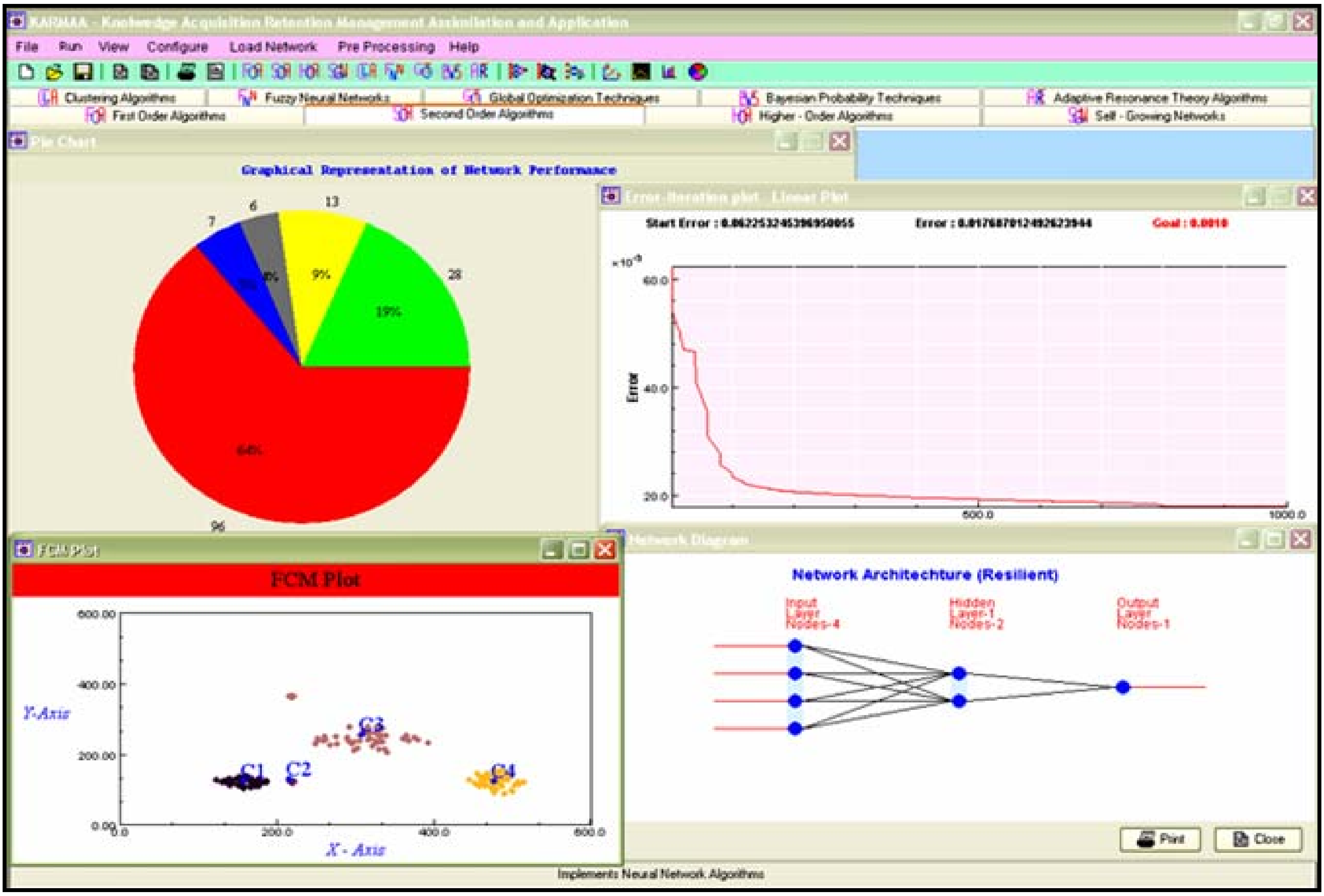
KARMAA brings together the latest in technology for knowledge solutions after more than a decade of research. The user is provided flexibility to use advanced neuron models and architectures that can be trained to fit even with fewer data examples. The algorithms are a result of thorough testing and validation following several industry implementations. For several case studies refer to the later part of the document. The uniqueness of the product lies in system design, architecture, user-interface, flexibility and algorithmic capabilities. We plan to tailor this general purpose simulator and analyzer for specific users by selecting the most relevant algorithms.

User Friendly Interface

We believe in a user-friendly technology that makes it possible for a novice to learn and use sophisticated algorithms to the best of his needs. The software suite has an expansive structure with all necessary options with proper icons. The overall system setup involves few guided clicks. Consider Artificial Neural Networks based algorithm, which are defined by the network architecture (*layers*) and the training algorithm (*learning rules*). The software guides the user through a set of algorithms and performance measure functions to choose from.

Powerful Visualization Modules

Data representation and understanding demands well-structured visualization tools that enables the user to understand the problem explicitly. All Knowledge Solutions begin with the choice of an appropriate modeling technique that demands prior analysis and accuracy investigation. The model should have better generalization abilities to identify the implicit data relationships. The software provides some of the most efficient techniques that are further tailored depending on the needs of the client. To understand the results obtained from these algorithms in a better way, proper visualization and demonstrable techniques are provided. From visualization of the performance and accuracy of the chosen algorithm and technique to explicit learning curves, the software brings them all together through a set of icons arranged over the main menu.



Support for Popular Platforms

All software modules have been implemented in Sun Microsystems' JAVA™ and runs on JAVA™ Virtual Machine for Microsoft Windows™ 98, 2000 and XP™ Operating Systems. In support to the Open Source Community, we also provide Knowledge Solutions for popular Linux Operating Systems (that include Red Hat, Mandrake and Gentoo). In addition to support for popular platforms, the software takes a minimal storage space and is highly computationally efficient.

Fast & Reliable

- Runs on Java Virtual machine (J2SDK 1.3.1 and above).
- Optimization of expressions before network evaluation to minimize the number of operations and also reduce the computational speed.
- Intelligent Organization of algorithms to begin the training with good performance.
- Well-organized palettes with command templates, options and links to online documentation.
- Extensive documentation including an introduction to the algorithms as well as specific applications.
- Special performance evaluation functions to validate and illustrate the quality of results.
- Powerful visualization tools, plots and charts with both online and offline performance measures.

Detailed Documentation

The software is equipped with a detailed software documentation that houses the following manuals along with a general help manual:

- **User Manual** - Provides a first-hand instruction for the user to begin using the software right away without any hassles.
- **Technical Manual** - Includes the algorithmic details of the methods provided with the software and can be used by developers who would like to improve upon existing techniques.
- **Verification Manual** - Contains useful case-studies of problems that have been successfully solved using the software.

KARMAA Technologies

- Modeling the World

ACES-107,
Department of Electrical Engineering,
Indian Institute of Technology, Kanpur,
Kanpur 208016, INDIA.
Phone: 91-512-2597810
Fax: 91-512-2590063
E-mail: kalra@iitk.ac.in
http://www.iitk.ac.in/karmaa/ktech

K Tech

KARMAA Technologies

Technical Details

Supported Network Architecture/ Neuron Models/ Algorithms

KARMAA includes various kind of Network Architecture such as Supervised Networks, Unsupervised Networks and Data Mining Techniques etc. In Supervised Networks, the Multi-Layered Feed-forward Networks are widely being used. Typically the network consists of a set of neurons that constitute the **input layer**, one or more **hidden layers** of computation nodes, and an **output layer** of computation nodes. The input signal propagates through the network in Feed-forward direction, on a layer-by-layer basis i.e. from input layer towards the output layer. The supervised Networks are used for-

- **Pattern recognition and classification**
- **Speech recognition and synthesis**
- **Forecasting**
- **Nonlinear system modeling**
- **Control etc.**

In the Unsupervised Networks, generally, the Clustering Methods are applied. The Unsupervised Networks are used for –

- **Classification**
- **Topology Mapping**
- **Decision Making**
- **Dimensionality Reduction etc.**

Supervised Networks

First Order Algorithms provides the family of Back-propagation algorithms. It consists of the following Learning Rules-

- **The Resilient Propagation (Batch Mode)**
- **Standard Back-Propagation (Pattern Mode)**
- **Back-Propagation with Momentum (Pattern and Batch Mode)**
- **Delta-Bar-Delta (Pattern Mode)**
- **Quick-Propagation (Batch Mode)**
- **Manhattan (Batch Mode)**
- **VOGL (Batch Mode)**
- **Gradient Based Heuristic (Pattern Mode)**
- **Global Descent (Batch Mode)**
- **Decoupled Momentum (Batch Mode)**

Second Order Algorithms the following two Learning Rules are in this case-

- **Scale Conjugate Gradient Algorithm**
- **Self-Scaling Scaled Conjugate Gradient Algorithm**

Higher Order Neurons The Standard Neuron has combination of aggregation and activation functions. The standard form of aggregation function can be linear weighted sum (linear basis function). The most common activation functions are sigmoid or tangent hyperbolic functions. Such neural models when used to solve real life problems may require a large number of neurons in standard neural network (STD). It is also well known that the number of unknowns to be determined to fix the architecture of the artificial neural network (ANN) grows with the number of neurons and the hidden layers. Therefore, the working of ANN becomes computation and memory intensive. The computational burden can be reduced either by reducing the number of neurons or by improving the learning techniques. The number of neurons in the ANN in turn depends on the neuron model itself. A higher neuron model may produce better ANN with fewer neurons. It provides the facility of choosing the following three types of Aggregation Functions and all the twenty-five (25) Activation (Transfer) Functions at ***neuron level***-

- **Summation**
- **Product of Two Functions**
- **Radial Basis Function**

It includes the following Learning Rules-

- **Back-Propagation with Momentum**
- **Quick-Propagation**
- **Resilient Propagation**
- **Simulated Annealing Resilient Propagation**

Self-Growing Networks The Back-propagation algorithm offers a way to learn arbitrarily complex boundaries using multilayer feed-forward networks. Such networks consist of one or more layers of hidden nodes between the input and output layers. The choice of the number of layers and the hidden nodes in each layer has a significant impact on the performance of networks. Thus one of the major criticisms of fully supervised feed-forward neural networks is their failure to cope with situations, which may require a novel topology. This is not so much due to the limitations of particular learning algorithms but is rather due to the limits of the networks structure and how these are set. The choice of network architecture and the type of neuron is usually a matter of trial and error.

Self-Growing Networks start with a small network and then add additional units and weights until a satisfactory solution is found. Pruning algorithms involve using a larger than needed network and training it until acceptable solution is found. After this, some hidden units or weights are removed which are no longer actively used.

There are three major classes in this case:

- **Cascade Correlation**
- **Cascade Error**
- **Cascade All**

All of these networks include the following learning rules-

- **Back-Propagation with Momentum**
- **Quick-Propagation**
- **Resilient Propagation**
- **Simulated Annealing Resilient Propagation**

Global Optimization Techniques Probabilistic global-optimization methods rely on probability to make the decisions. The simplest probabilistic algorithm uses restarts to bring a search out of a local minimum when little improvement can be made locally. More advanced methods such as simulated annealing rely on probability to indicate whether a search should descend from a local minimum.

The advantages of such techniques are that they don’t require any gradient information for updating weights, when there is no a priori information available to choose a search direction; these algorithms perform minimization of Error function and when Error is a complex function with a large number of weight variables and local minima, these will converge eventually to the global minimum. The Global Optimization techniques contains the following probabilistic methods-

- **Random Search (On-line and Batch Mode)**
- **Simulated Annealing**
- **ALOPEX [Algorithm for pattern extraction]**

Fuzzy Neural Networks Every intelligent technique has particular computational properties (e.g. ability to learn, explanation of decisions) that make them suited for particular problems and not for others. For example, while neural networks are good at recognizing the patterns, they are not good at explaining how they reach their decisions. Fuzzy logic systems, which can reason with imprecise information, are good at explaining their decisions but they can't automatically acquire the rules they use to make those decisions. These limitations have been a central driving force behind the creation of intelligent hybrid systems where two or more techniques are combined in a manner that overcomes the limitation of individual techniques. The computational power of Neuro -Fuzzy systems, which are fuzzy rule based systems, are implemented in the framework of neural networks. The Fuzzy Neural Networks consists of following Algorithms-

- **OR/AND Neuron based Neural Network (FNN)**
- **Adaptive Neuro Fuzzy Inference Systems**
- **Compensatory Neuro Fuzzy Systems**

Unsupervised Networks

Self Organizing Feature Maps The term self organization means the ability to learn and organize information by itself. The learning in this network is based on WTA (Winner Takes All) principle. The neuron with largest activation value is declared as winner. Only this neuron gives an output, all other neurons are suppressed to zero activation level. The main feature of Self-Organizing Maps is that it is **TOPOLOGY Preserving Networks**.

Adaptive Resonance Theory (ART) In unsupervised learning, the number of classes may be unknown previously. Therefore, the number of output nodes can not be accurately determined in advance. For solving this difficulty, it is useful to introduce a mechanism allowing adaptive expansion of the output layer until an adequate size is reached. It includes the following methods-

- **Binary Valued ART (ART1)**
- **Continuous Valued ART (ART2)**

Fuzzy C-Means (FCM) and FCM Robust Statistics (FCMRS) FCM/FCMRS is exclusively meant for partitioning the data set into cluster. It finds the cluster centers and simultaneously assigns the degree of membership of each point to a particular cluster. The algorithm works by an explicit minimization of the objective function and since the cluster centers are points of minimum in a distribution the algorithm has a natural proclivity to find them. The FCM algorithm is best suited to find spherical clusters.

Rule Based Algorithm Rules can be used to represent clusters. Rule premises can be assumed as cluster centers, since the rule-firing strength in it represents inverse of the distance of the data point from the rule premise part.

Gustafson – Kessel Algorithm (GKA) For elliptical and flat clusters the GKA algorithm is used that makes use of the fuzzy covariance matrix while computing distance between a feature vector and the center.

Other Networks

Bayesian Probabilistic Techniques The essence of the Bayesian Approach is to provide a mathematical rule explaining how you should change your existing beliefs in the light of new evidence. In other words, it allows us to combine new data with our existing knowledge or expertise.

Bayesian Networks are directed models. They have a more complicated notion of independence than undirected models, they do have several advantages. The most important is that one can record an arc from A to B as indicating that **A "causes" B**. This can be used as a guide to construct the graph structure. In addition, these models can encode deterministic relationships, and are easier to learn (fit to data). In addition to the graph structure, it is necessary to specify the parameters of the model. Specification of the Conditional Probability Distribution (CPD) at each node is prerequisite. If the variables are discrete, this can be represented as a conditional probability table (CPT), which lists the probability that the child node takes on each of its different values for each combination of values of its parents.

Principal Component Analysis In available data, many input variables are considered due to unavailability of proper knowledge regarding the percentage of contribution of each variable in system's behavior. The large number of input variables results in a large sized network, which in turn indicates the increase in cost and complexity in the hardware implementation. Therefore it is very important to reduce the number of input variables by a method which determines the percentage of contribution of each input variables in system's input-output behavior. Principal Component Analysis is one of the major techniques for dimensionality reduction.

Key Technical Features

A very wide choice of customizable parameters was incorporated in the design. The choice to pick from an array of 18 error functions, 3 aggregation functions and 25 activation or squashing functions, and using them in any combinations, creates a powerful utility.

Twenty-Five Activation Functions

Unipolar Sigmoid Transfer Function
Bipolar Sigmoid Transfer Function
Hyperbolic Tangent Transfer Function
Unipolar Arc Tangent Transfer Function
Bipolar Arc Tangent Transfer Function
Unipolar Sine Transfer Function
Bipolar Sine Transfer Function
Unipolar Cosine Transfer Function
Bipolar Cosine Transfer Function
Gaussian Transfer Function
Cauchy Transfer Function
Linear Transfer Function
Decaying Exponential Transfer Function
Unipolar-Threshold-Linear Transfer Function
Bipolar Threshold Linear Transfer Function
Unipolar-Threshold Transfer Function
Bipolar-Threshold Transfer Function
Signum Transfer Function
Net to the Power 'n' Transfer Function
Min Transfer Function
Max Transfer Function
Non-saturating Activation Function
Slow Sigmoid Transfer Function
Bipolar Logarithmic Transfer Function
[MIN-MAX] Transfer Function

Error Functions

Absolute Error Function
Andrew Error Function
Bipolar Hyperbolic Squared Error Function
Cauchy Error Function
Fair Error Function
Fourth Power Error Function
Geman-McClure Error Function
Huber Error Function
Hyperbolic Squared Error Function
Log-Cosh Error Function
Logarithmic Error Function
Mean-Median Error Function
Minkowski Error Function
Quadratic Error Function
Sinh Error Function
Tukey Biweight Error Function
Welsch Error Function
Root Mean Square Error Function

Normalization option is given for the scaling of the data between user-specific ranges.

Weights Initialization and Weight's Pruning facility is given for choosing of initial weights range and the pruning of the weights during the training process to get the global-minima.

Additive Iterations option is provided for continuous training with additional number of iterations without affecting the environment.

Separate Prediction Option for quickly predicting the output values, once the network is trained.

Loading and Configure an Existing Network facility is provided for loading of already trained neural network and reconfiguration of an existing network.

File-Formatting option is provided to make the proper training file format just from the row industrial/sensor data.

Graphical Help Attractive technical help guide with the Graphical User Interface.