

Neural Network Toolbox 4

Design and simulate neural networks

The Neural Network Toolbox extends MATLAB® with tools for designing, implementing, visualizing, and simulating neural networks. Neural networks are invaluable for applications where formal analysis would be difficult or impossible, such as pattern recognition and nonlinear system identification and control. The Neural Network Toolbox provides comprehensive support for many proven network paradigms, as well as a graphical user interface (GUI) that enables you to design and manage your networks. The modular, open, and extensible design of the toolbox simplifies the creation of customized functions and networks.

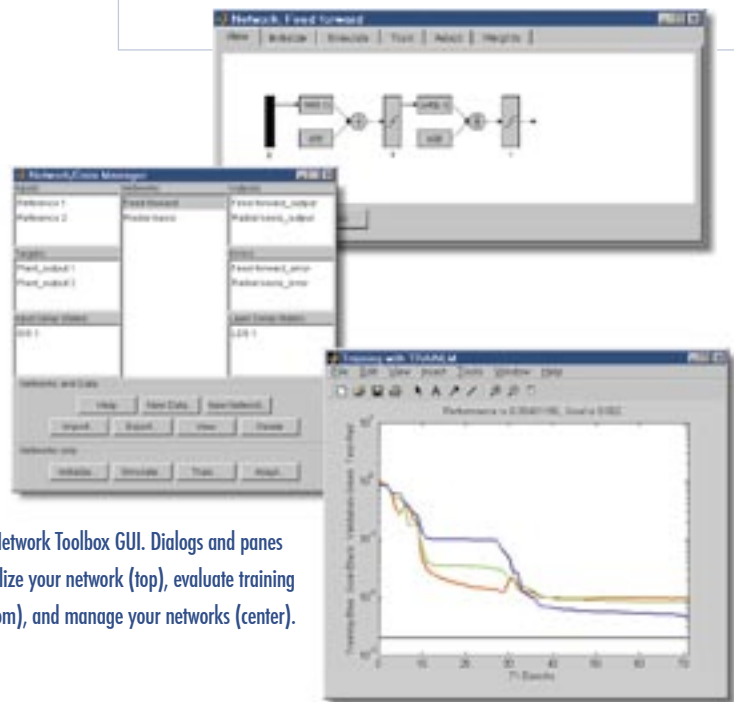
Working with the Neural Network Toolbox

Like its counterpart in the biological nervous system, a neural network can learn, and therefore can be trained to find solutions, recognize patterns, classify data, and forecast future events. The behavior of a neural network is defined by the way its individual computing elements are connected and by the strength of those connections, or weights. The weights are automatically adjusted by training the network according to a specified learning rule until it performs the desired task correctly.

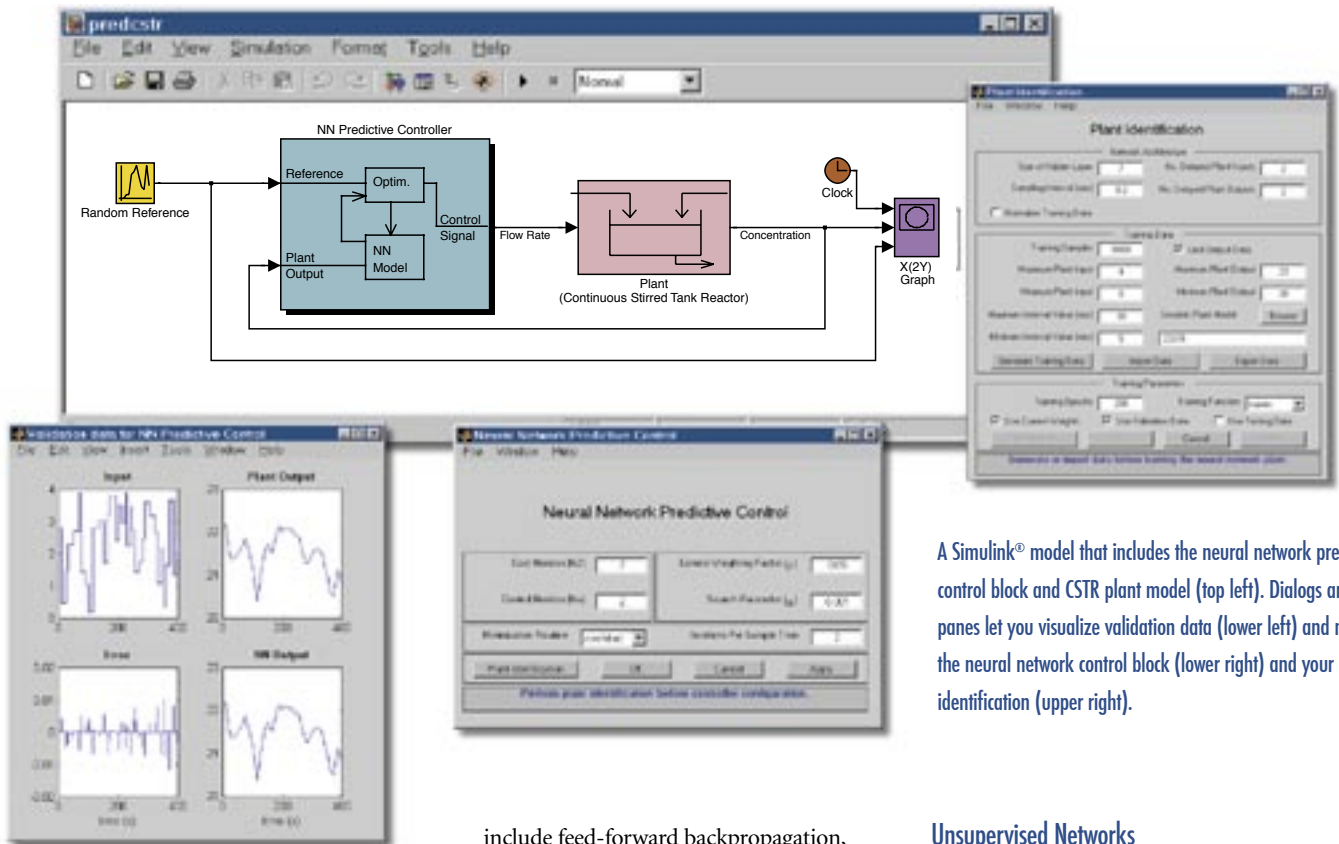
The Neural Network Toolbox GUI makes it easy to work with neural networks. It enables you to import large and complex data sets and quickly create, initialize, train, simulate, and manage your networks. Simple graphical representations enable you to visualize and understand network architecture.

KEY FEATURES

- GUI for creating, training, and simulating neural networks
- Support for the most commonly used supervised and unsupervised network architectures
- Comprehensive set of training and learning functions
- Simulink® blocks for building neural networks and advanced blocks for control systems applications
- Support for automatically generating Simulink blocks from neural network objects
- Modular network representation, enabling an unlimited number of input-setting layers and network interconnections
- Pre- and post-processing functions for improving network training and assessing network performance
- Routines for improving generalization
- Visualization functions for viewing network performance



The Neural Network Toolbox GUI. Dialogs and panes let you visualize your network (top), evaluate training results (bottom), and manage your networks (center).



A Simulink® model that includes the neural network predictive control block and CSTR plant model (top left). Dialogs and panes let you visualize validation data (lower left) and manage the neural network control block (lower right) and your plant identification (upper right).

Network Architectures

The Neural Network Toolbox supports both supervised and unsupervised networks.

Supervised Networks

Supervised neural networks are trained to produce desired outputs in response to sample inputs, making them particularly well suited to modeling and controlling dynamic systems, classifying noisy data, and predicting future events.

The Neural Network Toolbox supports four supervised networks:

- **Feed-forward networks** have one-way connections from input to output layers. They are most commonly used for prediction, pattern recognition, and nonlinear function fitting. Supported feed-forward networks

include feed-forward backpropagation, cascade-forward backpropagation, feed-forward input-delay backpropagation, linear, and perceptron networks.

- **Radial basis networks** provide an alternative fast method for designing nonlinear feed-forward networks. Supported variations include generalized regression and probabilistic neural networks.
- **Recurrent networks** use feedback to recognize spatial and temporal patterns. Supported recurrent networks include Elman and Hopfield.
- **Learning vector quantization (LVQ)** is a powerful method for classifying patterns that are not linearly separable. LVQ lets you specify class boundaries and the granularity of classification.

Unsupervised Networks

Unsupervised neural networks are trained by letting the network continually adjust itself to new inputs. They find relationships within data and can automatically define classification schemes.

The Neural Network Toolbox supports two types of self-organizing, unsupervised networks:

- **Competitive layers** recognize and group similar input vectors. By using these groups, the network automatically sorts the inputs into categories.
- **Self-organizing maps** learn to classify input vectors according to similarity. Unlike competitive layers, they also preserve the topology of the input vectors, assigning nearby inputs to nearby categories.

Training and Learning Functions

Training and learning functions are mathematical procedures used to automatically adjust the network's weights and biases. The training function dictates a global algorithm that affects all the weights and biases of a given network. The learning function can be applied to individual weights and biases within a network.

The Neural Network Toolbox supports a variety of training algorithms, including several gradient descent methods, conjugate gradient methods, the Levenberg-Marquardt algorithm (LM), and the resilient backpropagation algorithm (Rprop).

A suite of learning functions, including gradient descent, hebbian learning, LVQ, Widrow-Hoff, and Kohonen, is also provided.

Simulink Support

The Neural Network Toolbox provides a set of blocks for building neural networks in Simulink. These blocks are divided into three libraries:

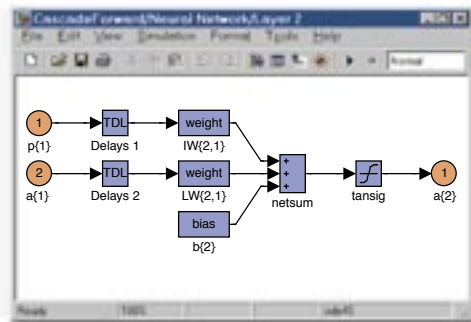
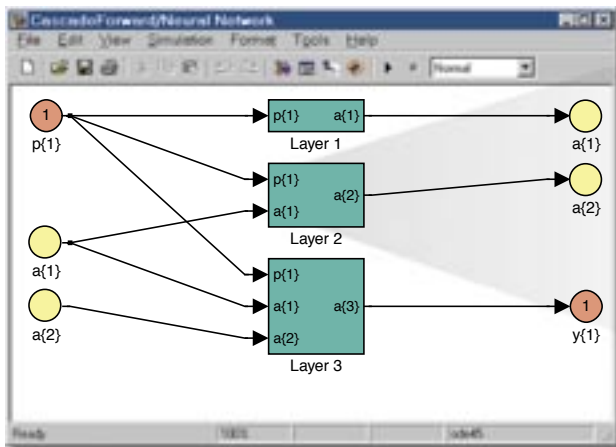
- **Transfer function** blocks take a net input vector and generate a corresponding output vector.
- **Net input function** blocks take any number of weighted input vectors, weight layer output vectors, and bias vectors, and return a net-input vector.
- **Weight function** blocks take a neuron's weight vector and apply it to an input vector (or a layer output vector) to get a weighted input value for a neuron.

Alternatively, you can create and train your networks in the MATLAB environment and automatically generate network simulation blocks for use with Simulink. This approach also enables you to view your networks graphically.

Control System Applications

The Neural Network Toolbox lets you apply neural networks to the identification and control of nonlinear systems. The toolbox includes descriptions, demonstrations, and Simulink blocks for three popular control applications: model predictive control, feedback linearization, and model reference adaptive control.

You can incorporate neural network predictive control blocks included in the toolbox into your Simulink models. By changing the parameters of these blocks, you can tailor the network's performance to your application.



A three-layer neural network converted into Simulink blocks. Neural network simulation blocks for use in Simulink can be automatically generated with the `gensim` command.

Pre- and Post-Processing Functions

Pre-processing the network inputs and targets improves the efficiency of neural network training. Post-processing enables detailed analysis of network performance.

The Neural Network Toolbox provides pre- and post-processing functions that enable you to:

- Reduce the dimensions of the input vectors using principal component analysis
- Perform regression analysis between the network response and the corresponding targets
- Scale inputs and targets so that they fall in the range [-1,1]
- Normalize the mean and standard deviation of the training set

Improving Generalization

Improving the network's ability to generalize helps prevent overfitting, a common problem in neural network design. Overfitting occurs when a network has memorized the training set but has not learned to generalize to new inputs. Overfitting produces a relatively small error on the training set but a much larger error when new data is presented to the network.

The Neural Network Toolbox provides two solutions to improve generalization:

Regularization modifies the network's performance function (the measure of error that the training process minimizes). By including the sizes of the weights and biases, training produces a network that performs well with the training data and exhibits smoother behavior when presented with new data.

Early stopping uses two different data sets: the training set, to update the weights and biases, and the validation set, to stop training when the network begins to overfit the data.

Required Products

MATLAB

Related Products

Control System Toolbox. Design and analyze controllers for closed-loop dynamic systems

Fuzzy Logic Toolbox. Design and simulate fuzzy logic systems

Optimization Toolbox. Solve standard and large-scale optimization problems

Statistics Toolbox. Apply statistical algorithms and probability models

System Identification Toolbox. Create linear dynamic models from measured input-output data

Platform and System Requirements

For platform and system requirements, visit www.mathworks.com/products/neuralnet/

For demos, application examples, tutorials, user stories, and pricing:

- Visit www.mathworks.com
- Contact The MathWorks directly
 - US & Canada 508-647-7000
 - Benelux +31 (0)182 696 700
 - France +33 (0)1 41 14 67 14
 - Germany +49 (0)241 470 750
 - Italy +39 (011) 2274 700
 - Korea +82 (0)2 6006 5114
 - Spain +34 91 799 1880
 - Sweden +46 (8)505 317 00
 - Switzerland +41 (0)31 950 60 20
 - UK +44 (0)1223 423 200

Visit www.mathworks.com to obtain contact information for authorized MathWorks representatives in countries throughout Asia Pacific, Latin America, the Middle East, Africa, and the rest of Europe.