

# Neural Network Toolbox™ 6

## Design and simulate neural networks

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. Neural Network Toolbox software provides comprehensive support for many proven network paradigms, as well as graphical user interfaces (GUIs) that enable 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 Neural Network Toolbox™

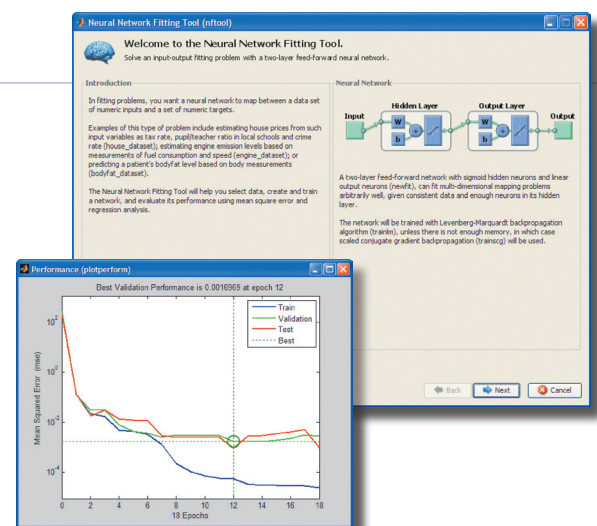
Like its counterpart in the biological nervous system, a neural network can learn and therefore can be trained to find solutions, recognize patterns, classify data, and forecast future events. The behavior of a neural network is defined by the way its individual computing elements are connected and by 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.

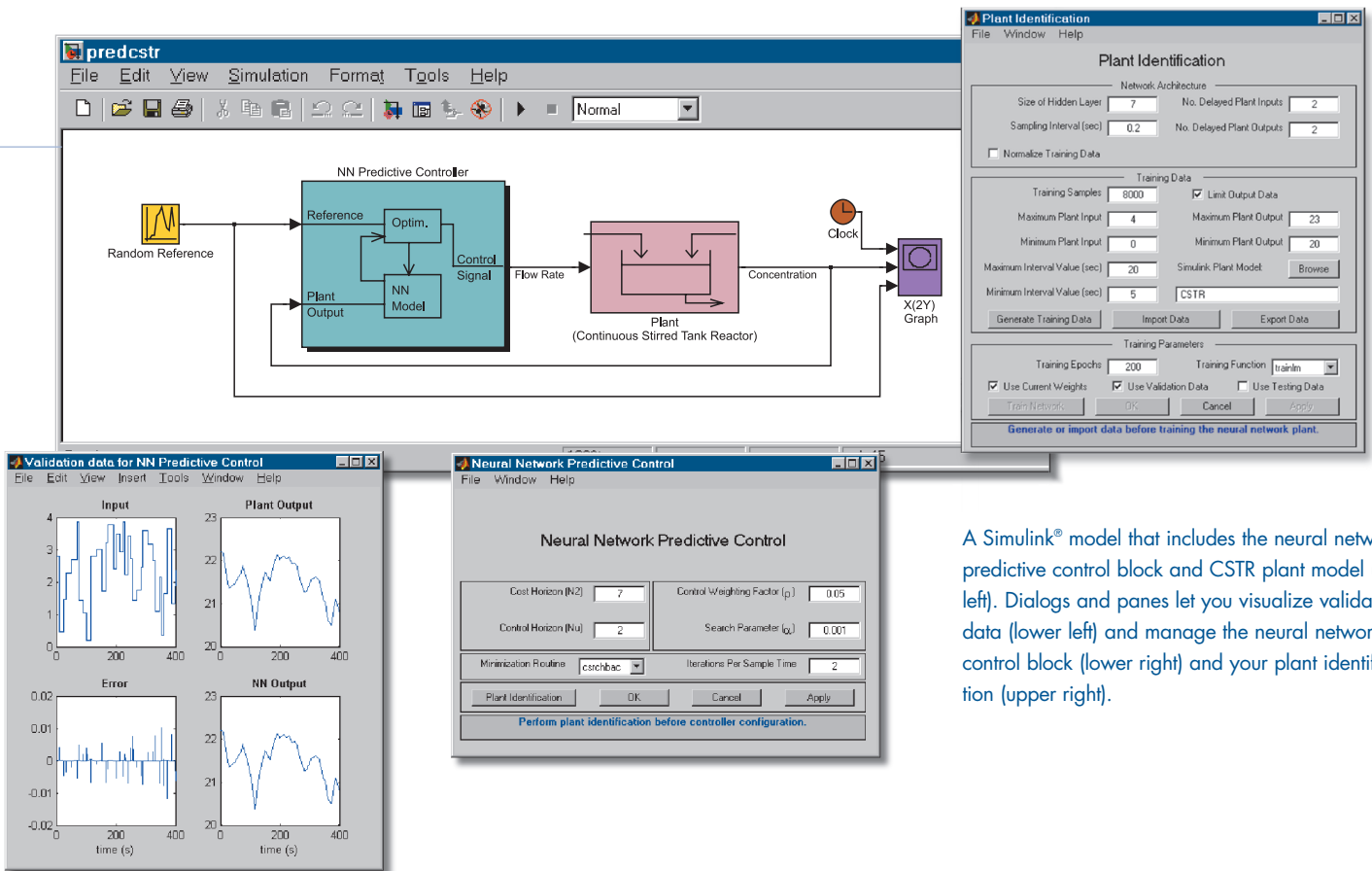
Neural Network Toolbox GUIs make it easy to work with neural networks. The Neural Network Fitting Tool is a wizard that leads you through the process of fitting data using neural networks. You can use the tool to import large and complex data sets, quickly create and train networks, and evaluate network performance.

The Neural Network Fitting Tool (top) and a performance plot (bottom). The Neural Network Fitting Tool guides you through the process of fitting data using neural networks, while additional GUIs are available for other common tasks such as pattern recognition and clustering.

### KEY FEATURES

- GUI for creating, training, and simulating neural networks
- Quick start wizards for fitting, pattern recognition, and clustering
- Support for the most commonly used supervised and unsupervised network architectures
- Comprehensive set of training and learning functions
- Dynamic learning networks, including time delay, nonlinear autoregressive (NARX), layer-recurrent, and custom dynamic
- 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 that enables an unlimited number of input-setting layers and network interconnections and a graphical view of network architecture
- Preprocessing and postprocessing functions and Simulink blocks for improving network training and assessing network performance
- Visualization functions and GUI for viewing network performance and monitoring the training process





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).

A second GUI gives you greater ability to customize the network architecture and learning algorithms. Simple graphical representations enable you to visualize and understand network architecture.

Additional GUIs are available for other common tasks including pattern recognition, clustering, and network training.

## Network Architectures

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.

Neural Network Toolbox supports four supervised networks: feedforward, radial basis, dynamic, and learning vector quantization (LVQ).

**Feedforward** networks have one-way connections from input to output layers. They are most commonly used for prediction, pattern recognition, and nonlinear function fitting. Supported feedforward networks include feedforward backpropagation, cascade-forward backpropagation, feedforward input-delay backpropagation, linear, and perceptron networks.

**Radial basis** networks provide an alternative, fast method for designing nonlinear feedforward networks. Supported variations include generalized regression and probabilistic neural networks.

**Dynamic** networks use memory and recurrent feedback connections to recognize spatial and temporal patterns in data. They are commonly used for time-series prediction, nonlinear dynamic system modeling, and control system applications. Prebuilt dynamic networks in the toolbox include focused and distributed time-delay, nonlin-

ear autoregressive (NARX), layer-recurrent, Elman, and Hopfield networks. The toolbox also supports dynamic training of custom networks with arbitrary connections.

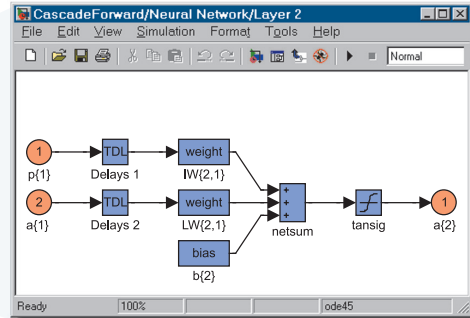
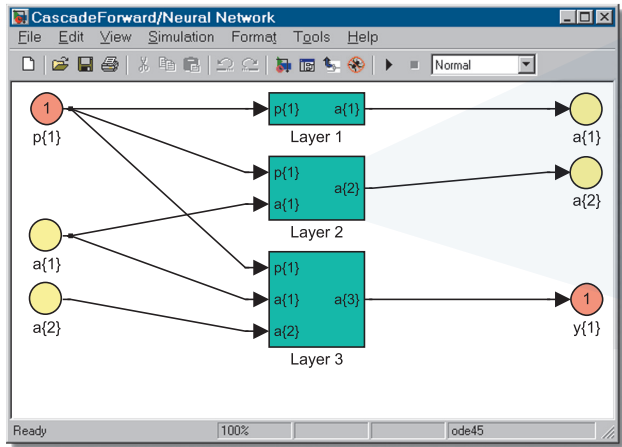
**LVQ** is a powerful method for classifying patterns that are not linearly separable. LVQ lets you specify class boundaries and the granularity of classification.

### 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.

Neural Network Toolbox supports two types of self-organizing, unsupervised networks: competitive layers and self-organizing maps.

Competitive layers recognize and group similar input vectors. By using these groups, the network automatically sorts the inputs into categories.



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.

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.

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). Algorithms can be accessed from the command line or via a training GUI, which shows a diagram of the network being trained, training algorithm choices, and stopping criteria values as the training progresses.

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

### Simulink® Support

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

- Transfer function blocks, which take a net-input vector and generate a corresponding output vector
- Net input function blocks, which take any number of weighted input vectors, weight layer output vectors, and bias vectors, and return a net-input vector
- Weight function blocks, which apply a neuron's weight vector to an input vector (or a layer output vector) to get a weighted input value for a neuron
- Data preprocessing blocks, which map input and output data into ranges best suited for the neural network to handle directly.

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

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.

### Preprocessing and Postprocessing Functions

Preprocessing the network inputs and targets improves the efficiency of neural network training. Postprocessing enables detailed

analysis of network performance. Neural Network Toolbox provides preprocessing and postprocessing functions and Simulink blocks 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

Automated data preprocessing and data division are built into the network creation process.

## 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.

Neural Network Toolbox provides two solutions to improve generalization: regularization and early stopping.

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 control systems

**Fuzzy Logic Toolbox™.** Design and simulate fuzzy logic systems

**Optimization Toolbox™.** Solve standard and large-scale optimization problems

**Statistics Toolbox™.** Perform statistical analysis, modeling, and algorithm development

**System Identification Toolbox™.** Create linear and nonlinear dynamic models from measured input-output data

## Platform and System Requirements

For platform and system requirements, visit [www.mathworks.com/products/neuralnet](http://www.mathworks.com/products/neuralnet) ■

## Resources

### VISIT

[www.mathworks.com](http://www.mathworks.com)

### TECHNICAL SUPPORT

[www.mathworks.com/support](http://www.mathworks.com/support)

### ONLINE USER COMMUNITY

[www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)

### DEMOS

[www.mathworks.com/demos](http://www.mathworks.com/demos)

### TRAINING SERVICES

[www.mathworks.com/training](http://www.mathworks.com/training)

### THIRD-PARTY PRODUCTS AND SERVICES

[www.mathworks.com/connections](http://www.mathworks.com/connections)

### WORLDWIDE CONTACTS

[www.mathworks.com/contact](http://www.mathworks.com/contact)

### E-MAIL

[info@mathworks.com](mailto:info@mathworks.com)