

# **Neural Network Toolbox Release Notes**

---

# Contents

---

- Summary by Version ..... 3**
- Version 5.0 (R2006a) Neural Network Toolbox ..... 5**
- Version 4.0.6 (R14SP3) Neural Network Toolbox ..... 9**
- Compatibility Summary for Neural Network Toolbox ..... 10**

## Summary by Version

This table provides quick access to what's new in each version. For clarification, see About Release Notes.

<b>Version (Release)</b>	<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>	<b>Related Documentation at Web Site</b>
<b>Latest Version V5.0 (R2006a)</b>	Yes Details	Yes Summary	No bug fixes	Printable Release Notes: PDF  V5.0 product documentation
V4.0.6 (R14SP3)	No	No	No bug fixes	No

### About Release Notes

Use release notes when upgrading to a newer version to learn about new features and changes, and the potential impact on your existing files and practices. Release notes are also beneficial if you use or support multiple versions.

If you are not upgrading from the most recent previous version, review release notes for all interim versions, not just for the version you are installing. For example, when upgrading from V1.0 to V1.2, review the New Features and Changes, Version Compatibility Considerations, and Bug Reports for V1.1 and V1.2.

### New Features and Changes

These include

- New functionality
- Changes to existing functionality
- Changes to system requirements (complete system requirements for the current version are at the MathWorks Web site)
- Any version compatibility considerations associated with each new feature or change

## **Version Compatibility Considerations**

When a new feature or change introduces a known incompatibility with the previous version, its description includes a **Compatibility Considerations** subsection that details the impact. For a list of all new features and changes that have compatibility impact, see the Compatibility Summary for Neural Network Toolbox.

Compatibility issues that become known after the product has been released are added to Bug Reports at the MathWorks Web site. Because bug fixes can sometimes result in incompatibilities, also review fixed bugs in Bug Reports for any compatibility impact.

## **Fixed Bugs and Known Problems**

MathWorks Bug Reports is a user-searchable database of known problems, workarounds, and fixes. The MathWorks updates the Bug Reports database as new problems and resolutions become known, so check it as needed for the latest information.

Access Bug Reports at the MathWorks Web site using your MathWorks Account. If you are not logged in to your MathWorks Account when you link to Bug Reports, you are prompted to log in or create an account. You then can view bug fixes and known problems for R14SP2 and more recent releases.

## **Related Documentation at Web Site**

**Printable Release Notes (PDF).** You can print release notes from the PDF version, located at the MathWorks Web site. The PDF version does not support links to other documents or to the Web site, such as to Bug Reports. Use the browser-based version of release notes for access to all information.

**Product Documentation.** At the MathWorks Web site, you can access complete product documentation for the current version and some previous versions, as noted in the summary table.

## Version 5.0 (R2006a) Neural Network Toolbox

This table summarizes what's new in V5.0 (R2006a):

<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>	<b>Related Documentation at Web Site</b>
Yes	Yes—Details labeled as <b>Compatibility Considerations</b> , below. See also Summary	No bug fixes	Printable Release Notes: PDF  V5.0 product documentation

New features and changes introduced in this version are organized by these topics:

- Dynamic Neural Networks
- Wizard for Fitting Data
- Data Preprocessing and Postprocessing
- Derivative Functions Are Obsolete

### Dynamic Neural Networks

Version 5.0 now supports these types of dynamic neural networks:

#### Time-Delay Neural Network

Both focused and distributed time-delay neural networks are now supported. Continue to use the `newfftd` function to create focused time-delay neural networks. To create distributed time-delay neural networks, use the `newtdnn` function.

#### Nonlinear Autoregressive Network (NARX)

To create parallel NARX configurations, use the `newnarx` function. To create series-parallel NARX networks, use the `newnarxsp` function. The `sp2narx` function lets you convert NARX networks from series-parallel to parallel configuration, which is useful for training.

## **Layer Recurrent Network (LRN)**

Use the `newlrn` function to create LRN networks. LRN networks are useful for solving some of the more difficult problems in filtering and modeling applications.

## **Custom Networks**

The training functions in the Neural Network Toolbox are enhanced to let you train arbitrary custom dynamic networks that model complex dynamic systems. For more information about working with these networks, see the Neural Network Toolbox documentation.

## **Wizard for Fitting Data**

The new Neural Network Fitting Tool is now available to fit your data using a neural network. The Neural Network Fitting Tool is designed as a wizard and walks you through the data-fitting process step by step.

To open the Neural Network Fitting Tool, type the following at the MATLAB prompt:

```
nftool
```

## **Data Preprocessing and Postprocessing**

Version 5.0 provides the following new data preprocessing and postprocessing functionality:

### **`dividevec` Automatically Splits Data**

The `dividevec` function facilitates dividing your data into three distinct sets to be used for training, cross validation, and testing, respectively. Previously, you had to split the data manually.

### **`fixunknowns` Encodes Missing Data**

The `fixunknowns` function encodes missing values in your data so that they can be processed in a meaningful and consistent way during network training. To reverse this preprocessing operation and return the data to its original state, call `fixunknowns` again with 'reverse' as the first argument.

### **removeconstantrows Handles Constant Values**

`removeconstantrows` is a new helper function that processes matrices by removing rows with constant values.

### **mapminmax, mapstd, and processpca Are New**

The `mapminmax`, `mapstd`, and `processpca` functions are new and perform data preprocessing and postprocessing operations.

**Compatibility Considerations.** Several functions are now obsolete, as described in the following table. Use the new functions instead.

<b>New Function</b>	<b>Obsolete Functions</b>
<code>mapminmax</code>	<code>premnmx</code> <code>postmnmx</code> <code>tramnmx</code>
<code>mapstd</code>	<code>prestd</code> <code>poststd</code> <code>trastd</code>
<code>processpca</code>	<code>prepca</code> <code>trapca</code>

Each new function is more efficient than its obsolete predecessors because it accomplishes both preprocessing and postprocessing of the data. For example, previously you used `premnmx` to process a matrix, and then `postmnmx` to return the data to its original state. In this release, you accomplish both operations using `mapminmax`; to return the data to its original state, you call `mapminmax` again with 'reverse' as the first argument:

```
mapminmax('reverse',Y,PS)
```

### **Derivative Functions Are Obsolete**

The following derivative functions are now obsolete:

```
ddotprod  
dhardlim  
dhardlms  
dlogsig
```

dmae  
dmse  
dmsereg  
dnetprod  
dnetsum  
dposlin  
dpurelin  
dradbas  
dsatlin  
dsatlins  
dsse  
dtansig  
dtribas

Each derivative function is named by prefixing a d to the corresponding function name. For example, sse calculates the network performance function and dsse calculated the derivative of the network performance function.

### **Compatibility Considerations**

To calculate a derivative in this version, you must pass a derivative argument to the function. For example, to calculate the derivative of a hyperbolic tangent sigmoid transfer function A with respect to N, use this syntax:

```
A = tansig(N,FP)
dA_dN = tansig('dn',N,A,FP)
```

Here, the argument 'dn' requests the derivative to be calculated.



## Version 4.0.6 (R14SP3) Neural Network Toolbox

This table summarizes what's new in V4.0.6 (R14SP3):

<b>New Features and Changes</b>	<b>Version Compatibility Considerations</b>	<b>Fixed Bugs and Known Problems</b>	<b>Related Documentation at Web Site</b>
No	No	No bug fixes	No

## Compatibility Summary for Neural Network Toolbox

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided with the description of the new feature or change.

<b>Version (Release)</b>	<b>New Features and Changes with Version Compatibility Impact</b>
<b>Latest Version V5.0 (R2006a)</b>	See the <b>Compatibility Considerations</b> subheading for this new feature or change: <ul style="list-style-type: none"><li>• mapminmax, mapstd, and processpca Are New</li><li>• Derivative Functions Are Obsolete</li></ul>
V4.0.6 (R14SP3)	None