

Numenta Platform For Intelligent Computing

Release Notes For Version 1.7

June 25, 2009

What's New In NuPIC 1.7?

NuPIC 1.7 is a new release of NuPIC. The primary focus of the release is a number of enhancements to the vision capabilities within NuPIC. These improvements complement the accompanying Vision Toolkit Application release.

Most of the changes in NuPIC 1.7 are incremental in nature, but together add up to much better performance in many vision tasks. There have been very few changes to the NuPIC API itself or in the underlying algorithms.

The full list of changes is summarized here, and described in more detail below:

- In general vision networks will run much faster and better than in NuPIC 1.6
- Speed optimizations to greatly improve the speed of training vision networks
- An optimized and enhanced `py.GaborNode`
- A new classifier, `py.PMClassifierNode` applicable to some vision tasks
- Speed optimizations for the `py.KNNClassifierNode` and `py.SVMClassifierNode`
- Enhancements to inspectors
- A new vision example demonstrating the use of NuPIC on the MNIST dataset
- A new example demonstrating the use of the Numenta Web Services API
- Improved documentation for the learning algorithm, plus some bug fixes for the temporal learning algorithms introduced in NuPIC 1.6

We would like to acknowledge our forum users who took the time to document and report bugs - thank you! All NuPIC users should upgrade to 1.7. Please note the compatibility issues below.

Installing The Update

The installation process is similar to previous versions. Please see installation instructions at <http://www.numenta.com/for-developers/education/installing-nupic.php> for instructions on how to install for the first time, or upgrade to 1.7. **Important:** NuPIC 1.7 uses Python 2.5.4. Please see the instructions for links to pre-built packages for Python and detailed installation instructions.

Description Of Changes

Speed Optimizations For Vision

Vision experiments in NuPIC 1.7 can run much faster than in NuPIC 1.6. This is due to a number of changes

- In NuPIC 1.6 `TemporalPoolerNodes` had to learn their invariances based on temporal sequences. In general, for most vision problems, the same invariances are learned each time you train the network. NuPIC 1.7 includes an option to use hardcoded invariances that we have found to work well on a variety of tasks. To use this option, set the `temporalPoolerAlgorithm` option to `hardcoded`.
- Loading and saving vision networks is much faster.
- The `py.GaborNode` has been optimized and is significantly faster.
- The `py.SVMClassifierNode` and `py.KNNClassifierNode` includes an option to reduce the input dimensionality before running the classifier. Dimensionality reduction is done by extracting the first N principal components of the training samples using Singular Value Decomposition.
- There are two new sample experiments that model the networks used in the Vision Toolkit. These experiments demonstrate the use of all of the above optimizations. To use them, copy the `params.py` file to a new experiment, and set the data paths at the top of the file to point to your training and testing data. The experiments are located in `share/vision/experiments/toolkitNetworks`.

Vision Framework Changes

The Vision Framework (located in `share/vision`) is software for running HTM networks on vision problems. The framework, released with NuPIC 1.6, can be used for creating and training networks on images, testing and debugging their recognition performance, and visualizing the results.

There have been numerous changes to the Vision Framework since NuPIC 1.6 to improve its overall behavior and performance. Please see the document “Vision Framework Guide” in `share/vision` for details on using the framework.

New Node type: `py.PMClassifierNode`

There is a new node `py.PMClassifierNode` that is specific to vision tasks. The classifier has special treatment for blank areas in an object, as well as an option to tune all its parameters automatically. We have found it to work quite well in some vision problems. Please see the experiment `share/vision/experiments/toolkitNetworks` for an example experiment that uses it.

New Vision Experiment: MNIST

NuPIC 1.7 includes a new vision example demonstrating how to use NuPIC on MNIST, a handwritten digit recognition problem that is a widely used benchmark within the machine learning community. The MNIST data set is publicly available database of handwritten digits containing 60,000 training samples and 10,000 testing samples. The MNIST data has been packaged and published by Dr. Yann LeCun of the Courant Institute.

Please see the README file in the directory `share/vision/experiments/mnist` for further instructions.

New Web Services Example

This new example demonstrates the use of the Numenta Web Services API. This API allows you to build web-based applications for networks created with the Numenta Vision Toolkit Application. The API documentation is online at <http://www.numenta.com/vision/webservices-api.php> Please see the README.TXT file in `share/projects/webservices` for details about the example itself.

Other Improvements

Pre-built python packages: some of the pre-built Python packages have been updated in the release: `wxPython` has been updated to version 2.8.9.2, `matplotlib` has been updated to version 0.98.5.2, and `numpy` has ben updated to version 1.2.1.

Source distribution changes: the source distribution has been reorganized into three separate downloads. The basic plugin source distribution includes the API, example source code, and libraries for building your own plugins. The learning plugin source includes source to the the learning algorithms and their node implementations. The tools source includes C++ source code for NuPIC tools. The latter two distributions are available only to users who agree to a separate license.

Backward Compatibility Notes

Network files saved under NuPIC 1.6.x will not load in NuPIC 1.7. The scripts for training and testing should generally still work.

Known Issues

- The `KNNClassifierNode` and `SVMClassifierNode` inputs cannot be inspected when PCA is turned on