

The Core Engine

1. Features

The core engine is the fulcrum around which are built all the other modules and applications based on Joone. It exposes a Java API providing the developer with the following features:

- Components to create any neural net architecture (feed forward or recurrent)
- Several supervised training algorithms:
 - On-line backprop
 - Batch backprop
 - Resilient Backprop (RPROP)
- Components to build unsupervised neural networks (Kohonen SOMs and Principal Component Analysis)
- Components to build modular neural networks
- A serialization mechanism to save and restore a nnet to/from a file system or to send/receive a nnet on remote locations (via HTTP, RMI, etc.)
- I/O components to read patterns from:
 - Ascii comma delimited files from file system or HTTP
 - Excel files
 - Image (gif & jpeg) files
 - RDBMS using JDBC
 - Stock price time series from Yahoo Finance
- Components to implement both the supervised and unsupervised learning control
- Components to control the behavior of the neural net (start/stop, recall/learn) and its parameters (learning rate, momentum, etc.)
- Plug-ins to preprocess the input data (Normalization, Moving Average, etc.) and to control dynamically the training parameters (Annealing)
- A complete event notification mechanism, along with a scripting capability to control the behaviour of a neural network at the happening of some events

2. Packages

The engine is composed by a set of components subdivided into the following packages:

- **org.joone.engine**

All the classes of the core engine which represent the bricks to build the neural nets

- **org.joone.engine.learning**
All the components to train the neural net
- **org.joone.io**
All the I/O components to read and write patterns from/to external sources
- **org.joone.net**
The shell components of the neural net. They are useful to manage a neural net as a indivisible entity
- **org.joone.util**
Some utility classes to perform several tasks (input normalization, input scaling, scripting, etc.)
- **org.joone.log**
The classes to write the output messages to a log file (it can use either Log4J or an internal logger)
- **org.joone.script**
The classes to define and execute BeanShell scripts (called also Macro)
- **org.joone.helpers**
Helper classes to make it very simple to build, train and test a neural network

3. Resources

- [The Joone Complete Guide \(pdf\)](#)
- [The Java API cookbook](#)
- The Netbeans Homepage (We use it to develop Joone)
- [The JOONE GUI Editor](#)
- [A sample GUI, that learns the XOR problem](#)
- [A selection of the best Neural Networks books](#)