Contribution ID: **46** Type: **not specified**

MDP 2.0 - A data processing framework for scientific development and education

Monday 3 July 2006 14:00 (30 minutes)

We present release 2.0 of the Modular toolkit for Data Processing (MDP), a data processing framework written in Python and based on numpy (the most popular numerical extensions to Python).

From the user's perspective, MDP consists of a collection of trainable algorithms or other data processing units (nodes) that can be combined into data processing flows. Given a sequence of input data, MDP takes care of successively training or executing all nodes in the flow. This structure allows to specify complex algorithms as a sequence of simpler data processing steps in a natural way. Training can be performed using small chunks of input data, so that the use of very large data sets becomes possible while reducing the memory requirements. Memory usage can also be minimized by defining the internals of the nodes to be single precision.

From the developer's perspective, MDP is a framework to make the implementation of algorithms easier. The basic class 'Node' takes care of tedious tasks like type and dimension checking, leaving the developer free to concentrate on the implementation of the training and execution phases. The node then automatically integrates with the rest of the library and can be used in a flow together with other nodes.

MDP 2.0 introduces some important structural changes. It is now possible to implement nodes with multiple training phases and even nodes with an undetermined number of phases. This allows for example the implementation of algorithms that need to collect some statistics on the whole input before proceeding with the actual training, or others that need to iterate over a training phase until a convergence criterion is satisfied. The ability to train each phase using chunks of input data is maintained if the chunks are generated with iterators. Moreover, it is now possible to define nodes that require supervised training in a very straightforward way by passing additional arguments (e.g., labels or a target output) to the 'train' method.

Moreover, new algorithms have been added, expanding the base of readily available basic data processing elements. Currently implemented algorithms include Principal Component Analysis, two flavors of Independent Component Analysis, Slow Feature Analysis, Gaussian Classifiers, Growing Neural Gas, Fisher Discriminant Analysis, and Factor Analysis.

MDP has been written in the context of theoretical research in neuroscience, but it has been designed to be helpful in any context where trainable data processing algorithms are used. Its simplicity on

the user side together with the reusability of the implemented nodes make it also a valid educational tool.

As its user base is steadily increasing, MDP appears as a good candidate for becoming a common repository of user-supplied, freely available, Python implemented data processing algorithms.

http://mdp-toolkit.sourceforge.net

Authors: BERKES, Pietro (Gatsby Computational Neuroscience Unit, London); ZITO, Tiziano (Institute for Theoretical Biology, Berlin)

Presenter: ZITO, Tiziano (Institute for Theoretical Biology, Berlin)

Session Classification: Python in Science

Track Classification: Python in Science