

The Wayback Machine - <https://web.archive.org/web/20011122181832/http://www-ra.informatik.uni-tuebinge...>



What is SNNS ?

SNNS (Stuttgart Neural Network Simulator) is a software simulator for neural networks on Unix workstations developed at the Institute for Parallel and Distributed High Performance Systems (IPVR) at the University of Stuttgart. The goal of the SNNS project is to create an efficient and flexible simulation environment for research on and application of neural nets.

The SNNS simulator consists of two main components:

- 1) simulator kernel written in C
- 2) graphical user interface under X11R4 or X11R5

The simulator kernel operates on the internal network data structures of the neural nets and performs all operations of learning and recall. It can also be used without the other parts as a C program embedded in custom applications. It supports arbitrary network topologies and, like RCS, supports the concept of sites. SNNS can be extended by the user with user defined activation functions, output functions, site functions and learning procedures, which are written as simple C programs and linked to the simulator kernel.

Currently the following network architectures and learning procedures are included:

- Backpropagation (BP) for feedforward networks
 - vanilla (online) BP
 - BP with momentum term and flat spot elimination
 - batch BP
- Counterpropagation
- Quickprop
- Backpercolation 1
- RProp
- Generalized radial basis functions (RBF)
- ART1
- ART2
- ARTMAP
- Cascade Correlation
- Recurrent Cascade Correlation
- Dynamic LVQ
- Backpropagation through time (for recurrent networks)
- Quickprop through time (for recurrent networks)
- Self-organizing maps (Kohonen maps)
- TDNN (time-delay networks) with Backpropagation
- Jordan networks
- Elman networks and extended hierarchical Elman networks
- Associative Memory

The graphical user interface XGUI (X Graphical User Interface), built on top of the kernel, gives a 2D and a 3D graphical representation of the neural networks and controls the kernel during the simulation run. In addition, the 2D user interface has an integrated network editor which can be used to directly create, manipulate and visualize neural nets in various ways.

