

# ASNN - Associative Neural Networks

---

The ASNN is an extension of the committee of machines that goes beyond a simple/weighted average of different models. ASNN represents a combination of an ensemble of feedforward neural networks and the k-nearest neighbor technique (kNN). It uses the correlation between ensemble responses as a measure of distance amid the analyzed cases for the kNN (or Parzen window regression). This corrects the bias of the neural network ensemble. An associative neural network has a memory that can coincide with the training set. If new data become available, the network instantly improves its predictive ability and provides data approximation (self-learn the data) without a need to retrain the ensemble. Another important feature of ASNN is the possibility to interpret neural network results by analysis of correlations between data cases in the space of models.

Several parameters can be specified by a user.

## Parameters overview

### Training method

This keyword indicates method that will be used to optimize neural network weights. All these methods, with an exception of stiff and Levenberg-Marquardt algorithms, are heuristic algorithms that have a number of adjustable parameters. These parameters were selected by respective authors to provide the fast convergence of the algorithms. Below of implementation of each algorithm are presented.

- **Momentum** is the simplest algorithm that triggered an interest in artificial neural networks after article of Rumelhart et al., 1986. The momentum rate is 0.9 and eps is 0.2.
- **SuperSAB** uses momentum is 0.9, eps is 0.2, multiplication(+) is 1.05, multiplication(-) is 0.5, max eps is 10.
- **RPROP** uses initial eps is 0.0001, minimal eps is 0.000001, multiplication(+) is 1.2, multiplication(-) is 0.5.
- **QuickProp** -- Falhman implementation updated from his original code. Eps is 0.55, decay is 0.0001, minimal derive is 0.1, maximal momentum is 1.75.
- **Differential equations** -- Second order optimisation algorithm that converts the weights optimisation to the problem of solution of the second order differential equations.
- **QuickProp ||** -- Another implementation inspired by the algorithm of Falhman. The same parameters as in QuickProp are used.
- **Levenberg-Marquardt** -- Second -order optimization algorithm (requires  $O(n^2)$  time, where n is the number of weights).

The default value is SuperSAB algorithm. It is not recommended to use **Differential equations** and **Levenberg-Marquardt** for any tasks with large number of descriptors or molecules (typically more than 100).

### Number of neurons

Only single layer neural networks are available. The number of neurons in input and output layers correspond to number of descriptors and number of properties, respectively.

### Learning iterations

The number of iterations that will be used in neural network training. The neural network training is stopped if there is no improvement of RMSE error for the validation set after ITERATIONS iterations (this corresponds to detection of the early stopping point). The training is also stopped if total number of iterations in program equals to 5\*ITERATIONS.

## Ensemble

Indicates the number of network in ensemble that will be analyzed. To use ASNN algorithm typically 64 - 100 neural networks are used. For fast preliminary calculations 10 networks can be also used to explore data.

## Disable ASNN

Disables ASNN correction and uses standard ensemble average.

## References

- Tetko, I.V. Associative Neural Network, [Neural Processing Letters](#), 2002, 16, 187-199.
- Tetko, I.V. Neural Network Studies. 4. Introduction to Associative Neural Networks, [J. Chem. Inf. Comput. Sci.](#), 2002, 42, 717 -728.
- Tetko, I.V. Associative Neural Network, In: [Neural Networks: Methods and Applications](#), Livingstone, D.J., Ed., The Humana Press Inc., vol. 458, 2008, pp. 185-202.