Self-organizing Maps vs. Backpropagation: An Experimental Study *

Josef GÖPPERT and Wolfgang ROSENSTIEL
Lehrstuhl für Technische Informatik, Sand 13, 72076 Tübingen, Germany
goeppert@peanuts.informatik.uni-tuebingen.de

Abstract. Counter-propagation architecture with a self-organizing map in the competition layer is a powerful tool in various domains of signal processing and function approximation. Adopted with interpolation techniques in the output layer, the performance can be raised once more. Though the number of neurons is minimised the precision of the output is increased. Good and fast convergence of the self-organizing map, topology preservation combined with interpolated real output values may be an alternative to backpropagation for several applications. In this paper we compare the evaluation results of counter-propagation architecture with backpropagation trained feed-forward nets. The performance is tested with a real evaluation problem.

1 Introduction

Neural Networks with feed-forward structure are commonly used in various application domains. Initially used for pattern recognition tasks they have turned out to be suitable for several other tasks like function approximation or general data evaluation. A feed-forward net can solve any unambiguous evaluation problem in arbitrary accurate precision, if its architecture is appropriate and the correct interconnection weights are found. The training of the weight values can be realized with the backpropagation algorithm [6]. This algorithm realizes a gradient descent in an error landscape. This type of optimization method is mathematically well known and is able to find a minimal weight configuration.

But nevertheless feed-forward nets with backpropagation have turned out to be difficult to handle:

- Estimation of the number of layers of this net is very difficult.
- The ideal number of neurons in the hidden layers is not known.
- The convergence of the backpropagation algorithm depends strongly on its training parameters.
- The algorithm may converge to local minima. This depends on the training parameters and especially on the initial weight values.
- Large time can be needed for the convergence of the algorithm.

*In Proc. of *Workshop Design Methodologies for Microelectronics and Signal Processing,* P. 153-162, Institute of Electronics, Silesian Technical University, Giwice, Poland, Oct. 1993.

- Interpretation of trained weight values is very difficult because of the distributed character of information storage.

The development of a feed-forward net, either needs a lot of experience, exact knowledge of data properties or comparison of evaluation results of a large number of training cycles. These properties are a big hindrance to application of feed-forward nets in industrial environment.

The counter-propagation network [3] tries to overcome these problems by using another type of neurons in the competition layer. Information is stored locally at one neuron. Only the neuron with the biggest similarity to the input vector (the winner of competition) defines the output of the network, all other neurons are ignored. This type of network is commonly used for direct and inverse function approximation tasks.

A very common idea in the competition layer is the use of a self-organizing map (SOM) [4]. This unsupervised training algorithm converges to a solution which approximates input data by adapting prototype vectors. As an additional feature of this algorithm the neighbourhood relation of neurons is considered, which leads to a topology preserving mapping of the training data. This algorithm shows high ability in adaption to complex nonlinear data and has fast and stable convergence. Another algorithm which is inspired by the SOM algorithm is called "neural gas" [5] which realizes a mapping without knowledge of optimal data topology, but without topology preservation.

The use of interpolation techniques in the output layer may increase the performance of the system once more. Interpolation produces real output values and allow to work with SOM's of reduced size. Two different interpolation methods [1] using geometrical and topological interpolation are used.

2 Used Architecture

In this work we are using the normal counter-propagation architecture with a self-organizing map in the competition layer. A schematic representation is shown in figure 1.

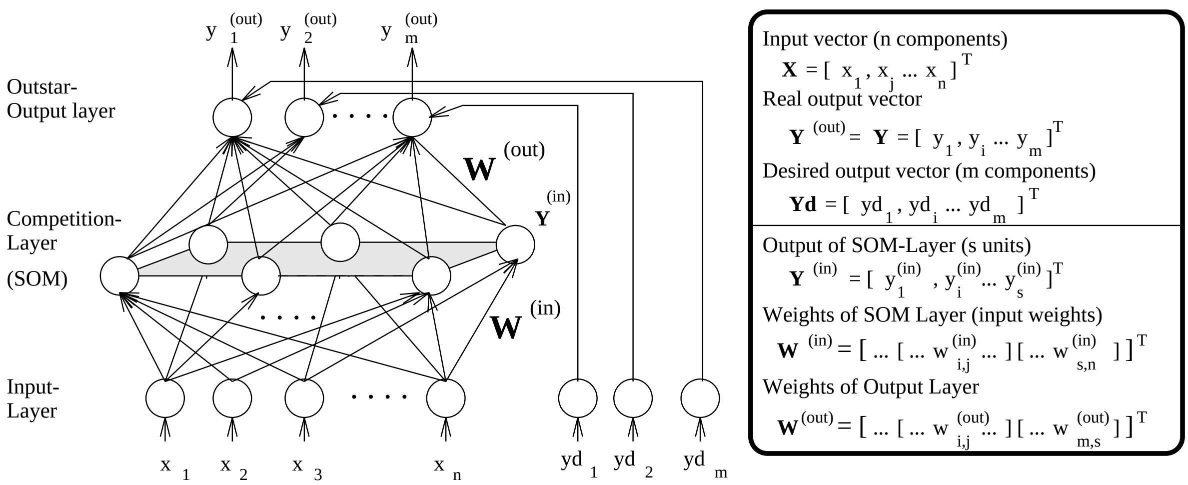


Figure 1: Architecture of the counter-propagation network.

The input layer sends the input values to the competition layer. Simple mathematical operation like normalization can be applied before. In the competition layer one input

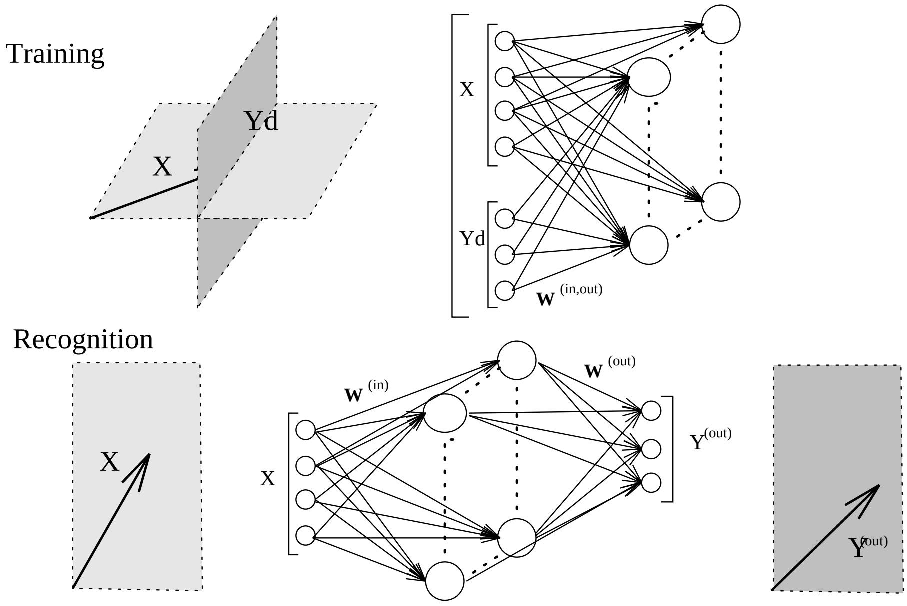


Figure 2: Training of the output layer

vector is compared to all prototype vectors of the neurons. The best matching unit is searched for. It also calculates the interpolation parameters. The output layer associates the output vector with the winning neuron with respect to the interpolation parameters.

Each neuron in the competition layer represents an association of an input vector to its corresponding output configuration. The difference between the input vector and the configuration stored at a neuron i is defined by a distance function d, in this case by the euclidean distance:

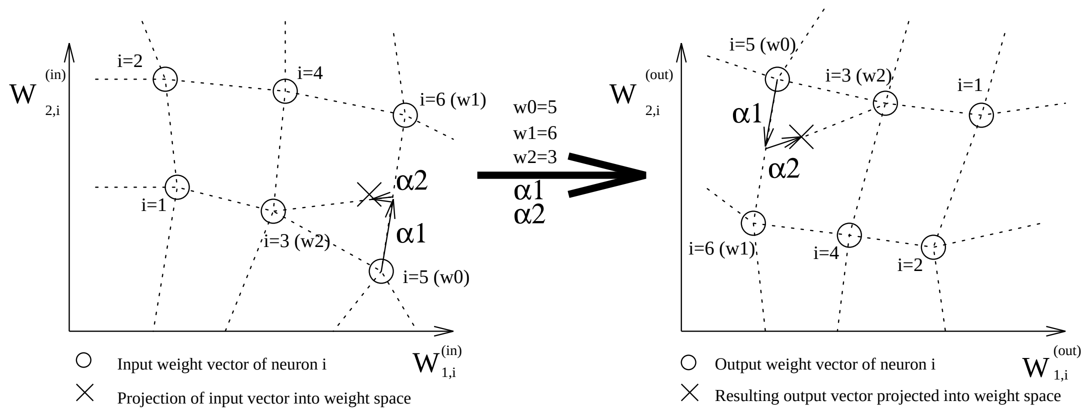
$$
\begin{aligned}
D_i = &\ d(\mathbf{X}, \mathbf{W}_i^{(in)}) && \mathbf{W}_i^{(in)} : Weight\ vector\ (Prototype) \\
D_i = &\ \sqrt{\sum_{j=1}^{n} \left(x_j - w_{i,j}^{(in)}\right)^2} && euclidean\ distance
\end{aligned}
\tag{1}
$$

2.1 The training of the Network

Training of the network is divided into two parts. First unsupervised training of the self-organizing map approximates the input vectors. The prototype vectors of the neurons are found as a configuration with minimum distance between input vector and corresponding winning neuron. The second step is a supervised adaption of the output weights (outstar training).

The desired output of the net is predetermined. Thus this information can be used to train the competition layer. The idea is to combine the n-dimensional input vector and the m-dimensional output vector to a $n+m$-dimensional training vector and to do training in $n+m$-dimensional space. This type of training realizes implicitly an association of the input-output vectors. It represents some kind of supervised SOM training, helps to come to a faster convergence and gives preference to organize the map according to input-output relationship (See figure 2).

After the training, this matrix is splitted into its input part $\mathbf{W}^{(in)}$ and its output part $\mathbf{W}^{(out)}$. Some cycles of weight adaption according to the outstar training procedure may



$$
\begin{aligned}
\tilde{\mathbf{X}}_0 &= \mathbf{W}_{w0}^{(in)} \\
\alpha_j &= \frac{\left(\mathbf{X} - \tilde{\mathbf{X}}_{j-1}\right)^T \left(\mathbf{W}_{wj}^{(in)} - \tilde{\mathbf{X}}_{j-1}\right)}{\left|\mathbf{W}_{wj}^{(in)} - \tilde{\mathbf{X}}_{j-1}\right|^2} \\
\tilde{\mathbf{X}}_j &= \tilde{\mathbf{X}}_{j-1} + \alpha_j \left(\mathbf{W}_{wj}^{(in)} - \tilde{\mathbf{X}}_{j-1}\right)
\end{aligned}
$$

$$
\begin{aligned}
\tilde{\mathbf{Y}}_0 &= \mathbf{W}_{w0}^{(out)} \\
\tilde{\mathbf{Y}}_j &= \tilde{\mathbf{Y}}_{j-1} + \alpha_j \left(\mathbf{W}_{wj}^{(out)} - \tilde{\mathbf{Y}}_{j-1}\right)
\end{aligned}
$$

Figure 3: Interpolation in two dimensional output space using geometrical information

follow to improve the output values [2]. During the recognition phase an input vector activates the best matching input weight vector $\mathbf{W}_i^{(in)}$ at neuron i, which applies the corresponding output vector $\mathbf{W}_i^{(out)}$ to the network output.

2.2 Winner takes all

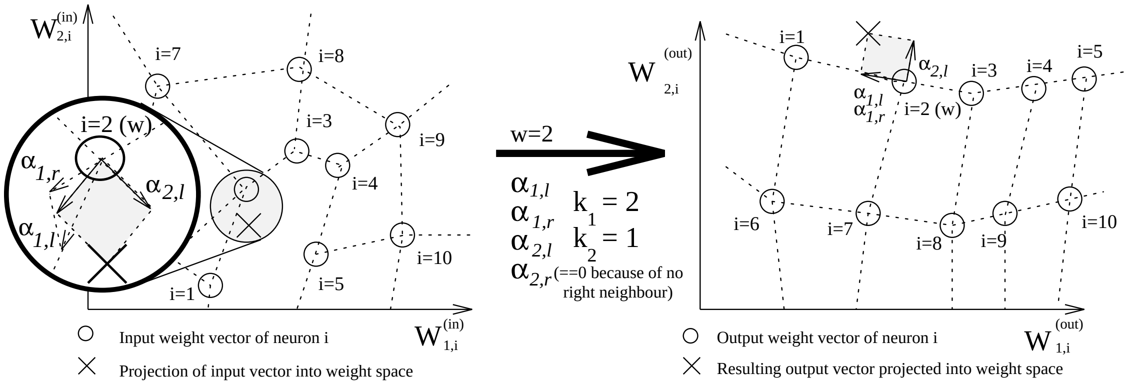
WTA and two different interpolation methods [1] are used in this paper. Geometrical interpolation tries to transfer the geometry in the input space into the space of output values. In the case of topology preserving map (SOM) ambiguous configurations can be avoided using the topology, which leads to a topological interpolation [1].

The simplest case is the WTA training. Here, the most similar neuron (smallest Distance: $D_w \leq D_i \quad \forall i$) is called "winner" (Index w) of the competition. This winning neuron (competition layer) has an output value of $y_w^{(in)} = 1$. The other neurons have output values of $y_i^{(in)} = 0; \quad i \neq w$ which lead to the following relation:

$$
y_i^{(out)} = \sum_j w_{i,j}^{(out)} y_j^{(in)} = w_{i,w}^{(out)} \tag{2}
$$

2.3 Geometrical Interpolation

A good idea is the use of geometrical information in the input space for the interpolation in the output space. It is tried to project the *geometry* from the input into the output space, by help of some global parameters. Interpolation is done between several winners (Index $w0$ for the first winner, wj for the jth winner) in order to find the best approximation $\tilde{\mathbf{X}}$ of the input vector $\mathbf{X}$. The interpolation parameters in the input space are used in the same way in the output space in order to get an the estimate of the output vector. The principle is visualised in figure 3.



$$\alpha_{d,l} = \frac{(\mathbf{X} - \mathbf{W}_w^{(in)})^T \left(\mathbf{W}_{d,l}^{(in)} - \mathbf{W}_w^{(in)}\right)}{\left|\mathbf{W}_{d,l}^{(in)} - \mathbf{W}_w^{(in)}\right|^2}$$

$$\alpha_{d,r} = \frac{\left(\mathbf{X} - \mathbf{W}_w^{(in)}\right)^T \left(\mathbf{W}_{d,r}^{(in)} - \mathbf{W}_w^{(in)}\right)}{\left|\mathbf{W}_{d,r}^{(in)} - \mathbf{W}_w^{(in)}\right|^2}$$

$$\begin{aligned} \mathbf{Y}^{(out)} &= \mathbf{W}_w^{(out)} + \sum_{d=1}^{Dim} \Delta\mathbf{W}_d^{(out)} \\ \Delta\mathbf{W}_d^{(out)} &= \frac{\alpha_{d,l}}{k_d}(\mathbf{W}_{d,l}^{(out)} - \mathbf{W}_w^{(out)}) \\ &+ \frac{\alpha_{d,r}}{k_d}(\mathbf{W}_{d,r}^{(out)} - \mathbf{W}_w^{(out)}) \end{aligned}$$

Figure 4: Interpolation in two dimensional output space using topological information According to distance in input space the neuron $i = 5$ will be the second winner, even if it represents in the output space a completely different configuration. By the use of topological interpolation this type of error is avoided.

An advantage of this method is, that α_j can also become negative if the next winner is situated in the opposite direction. This feature enables linear extrapolation into regions outside of the area covered by the self-organizing map.

This method can be coded in a neural structure with a linear output neuron. Output values of winning competition neurons are calculated by iterative adaption according to the number of winners.

2.4 Use of topological information

The topological interpolation takes into account the most important feature of the SOM, its topology preservation. With highly nonlinear relationship in input data, the second, or third winner can represent a configuration, which is similar in input space but quite different in the output space. One example of such a configuration is shown in figure 4.

Topological information takes only care of the winning neuron and its topological neighbours, to avoid averaging of completely different configurations. It is supposed, that all direct neighbours of the winning neuron interact in interpolation. The parameter k_d indicates the number of neighbours of dimension d — in border regions: $k_d = 1$, otherwise: $k_d = 2$.

Like topological interpolation, this method can be coded in a neural structure. It must be noticed, that this kind of interpolation strongly needs data adapted topology of the map. If the topology doesn't fit to the internal data organization, big topological defects will produce big errors, and geometric interpolation combined with neural gas training will be a better choice.

3 Results

The methods described in the previous section are used to solve a real evaluation problem in the domain of gas recognition [2]. It was shown that the self-organizing map is able to solve this type of problem, if enough neurons are used. Here the configuration of standard counter-propagation and interpolation methods is compared to a backpropagation net.

3.1 Evaluation problem

New sensor systems, developed at the University of Tübingen, are able to do continuous control of chemical processes, exhaust gas and waste water. Chemical and biochemical membranes are used in order to achieve multi-component systems, which are evaluated by interference spectra in the range of visible light. The variation of thickness of polymerical membranes is a result of absorption of organic solvents from its environment.

The optical part consists of a fiber, placed perpendicularly to the polymer film which provides light in the range of 300 to 720 nm by a XENON light source. The reflected light is analysed by a diode array spectrometer with 32 spectral diodes. The exact concentration of used gases for the training spectra is known by the settings of mixing unit. Training is done with a subset of 120 input-output vectors. The evaluation of the error is done with 1080 different input vectors. The evaluation error is calculated as the root mean square (RMS) error of 10 independent tests. The standard deviation of this tests is calculated.

3.2 Counter-propagation and interpolated SOM

In a first step, data is evaluated according to the original counter-propagation algorithm. Each winning neuron in the competition layer is associated to an output value. So the number of possible output values cannot exceed the number of neurons in this layer. This is the reason why a small number of neurons produces a big error (figure 5). The error is considerably decreasing for a bigger number of neurons. The algorithm has a certain convergence to configurations with the similar evaluation error. This property can be confirmed by the small range of variance of the RMS error on the validation data (see standard deviation in figure 5). The standard deviation is smaller than the scale of resolution and therefore not visible. This also holds for the following figures 6 and 7.

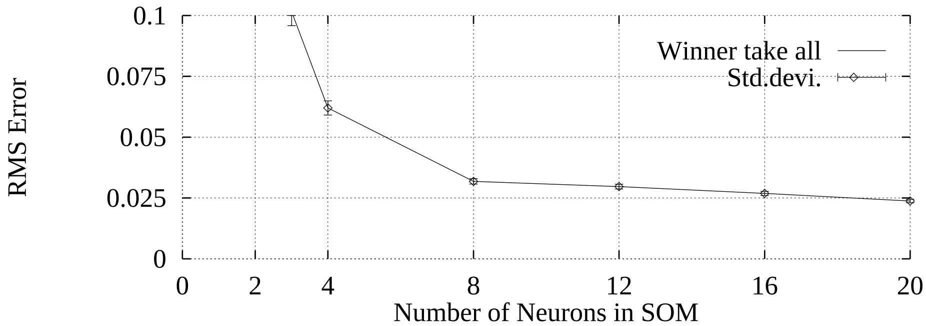


Figure 5: The relative error and its standard deviation as a function of the number of neurons for winner take all.

Geometrical interpolation is supposed to have better interpolation and especially better extrapolation properties (figure 6). In this test the maps of the same SOM training cycles as in the previous subsection were used, but with the application of geometrical interpolation of three winners. All configurations show better results. Even with a small number of neurons, the results are very good. In fact, four neurons being geometrically interpolated, lead to a better output than 16 WTA output neurons, The good result with small map size confirms the extrapolation capacities. The variance of the mean error increases, but remains small. So different training cycles lead to similar configuration and stable convergence.

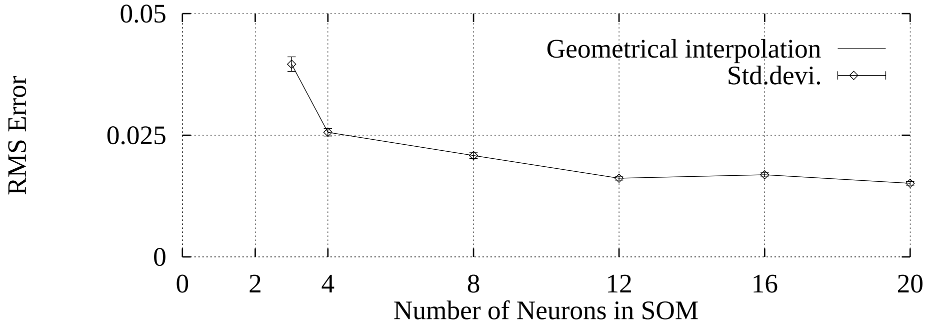


Figure 6: Relative RMS error and standard deviation for geometrical interpolation.

Topological interpolation takes advantage of the topology preserving feature of the self-organizing algorithm. The result is nearly equivalent to the result of the geometrical interpolation. In the given case both methods lead to nearly the same RMS error and variance.

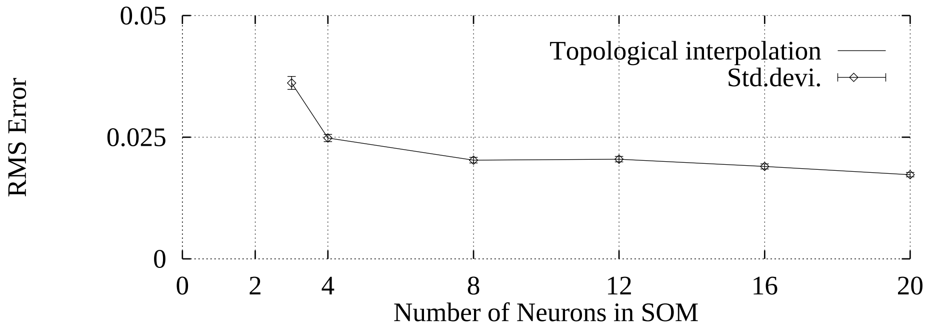


Figure 7: Relative RMS error and variance for topological interpolation.

It must be noticed, that in the given example the non-linearities are not very strong, so the use of geometrical interpolation may be sufficient. Comparable results for both methods show similar capacities, even if the set of used interpolation neurons (k-winners or neighbours) and the type of interpolation (iterative approximation or averaging of left and right neighbour) is quite different.

3.3 Feed-forward nets

Counter-propagation and interpolated SOM is capable to solve this type of evaluation problem. In this chapter the performance is compared to standard backpropagation. The problem was solved for different net configurations. A one layered linear neuron (in fact a special case of backpropagation), and two layered feed-forward nets with variable number of neurons in the hidden layer (2,4,8,12,16,20) and a linear output neuron are used.

First it can be noticed, that backpropagation is able to solve the given evaluation problem. A linear neuron results in a RMS error of 4.2%. With increasing number of neurons, the error decreases and seems to reach about 2% (SOM: 1.5%). In fact, backpropagation with 4 neurons in the hidden layer has a better result as interpolated SOM with 4 neurons, but unlike SOM, increasing the number of neurons didn't lead to better results. Considering figure 8 a big standard deviation of the remaining error after 2000 training cycles can be seen, which reflects an uncertain convergence. There are only one or two, out of ten tests responsible for this big variance (training get stuck in local minima, see Min/Max error in figure 8). To know if the training was successful, different tests are necessary.

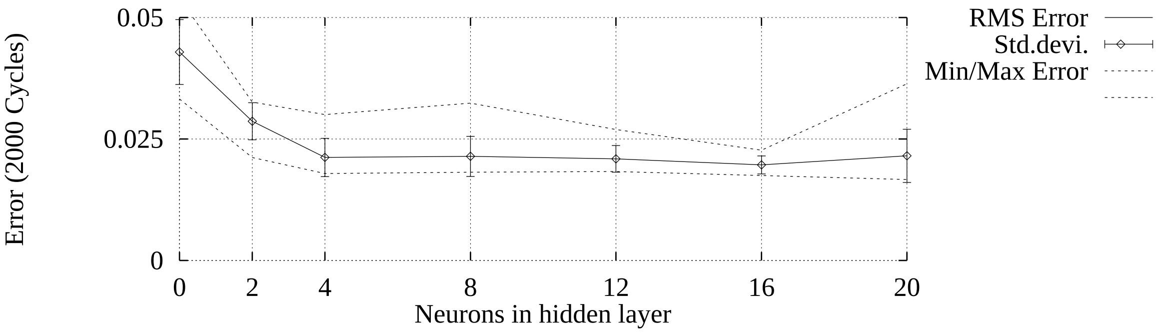


Figure 8: Relative RMS error on validation data after 2000 training cycles with standard deviation and Min/Max error.

This variance is not reduced by longer training. In figure 9 even bigger standard deviation can be observed (20000 training cycles).

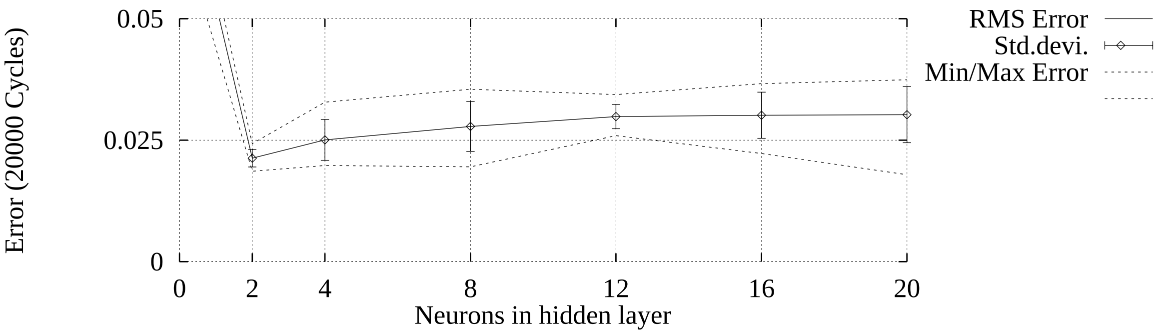


Figure 9: Relative RMS error on validation data after 20000 training cycles with standard deviation and Min/Max error.

Longer training did not guarantee better convergence, even worse: The RMS error on the validation set increases. This effect is well known for backpropagation and is called "over-learning". The backpropagation algorithm realizes a gradient minimization on the training data. In the beginning the output error on training data and the output error on validation data decreases up to an optimal configuration. Training should stop at this point. Longer training results in a better adaption of training data. In the case of noisy data, this noise would be learned, but error on validation data — with different noise values — increases. The effect of over-learning is shown in figure 10.

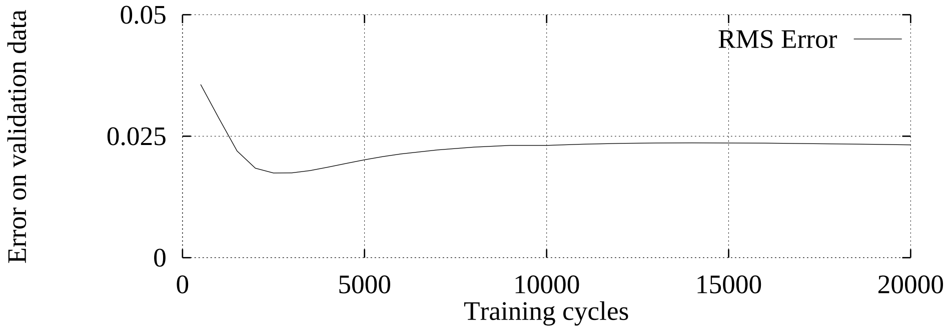


Figure 10: Relative RMS error of validation data as a function of training cycles.

3.4 Comparison

In the presented evaluation problem, good and stable convergence can be observed for the counter-propagation net and the self-organizing map with interpolated output. Convergence of backpropagation turned out to be difficult. Several times it get stuck in local minima, and the over-learning effect made it difficult to find optimal training parameters.

It can be noticed that for a small (but the same) number of neurons in the hidden layer, the backpropagation net is still better, but incrementation of the number of hidden neurons, doesn't lead to better results. This is due to the fact, that incrementing the number of neurons increments also the risk to get stuck in a local minimum, one of the major disadvantage of the backpropagation algorithm. Contrary for counter-propagation, incrementation of the number of neurons doesn't raise the risk of local minimum, but results in a bigger number of used prototypes, which leads to a better approximation. For this reason, more neurons can (and will) be used in counter-propagation than in backpropagation. Notice also, that the training of 20 neuron SOM is faster ($\approx$ 10 sec.) than training of 4 neuron backpropagation ($\approx$ 2 min). The best overall result was obtained by the interpolated SOM (see table 1).

Table 1: Comparison of the best configurations of all methods.

Method	Error	Neurons in hidden layer
Geometrical interpolation	0.015	20
Topological interpolation	0.017	20
Backpropagation	0.020	$4 \ldots 16$
Winner take all	0.023	20

The problems, discussed in the introduction and the problems, seen in the given evaluation problem make it difficult for inexperienced users to develop backpropagation nets. Especially in industrial environment the time and know-how of backpropagation training is not available and the uncertain convergence properties prevent the use in security relevant domains. This is completely different for counter-propagation and self-organizing maps. User friendly design of the network architecture and certain convergence allow a fast development of applications and the definition of statistically significant statements of its properties. The use of output interpolation allows the reduction of neurons in the competition layer and increases the precision of output values.

In fact there are some application domains, where self-organizing maps cannot be used, but in large domains of signal processing and function approximation the counter-propagation architecture is able to solve the given problem and to take advantage of its properties, described in this paper.

4 Conclusion

Using counter-propagation architecture and interpolation of output aims to application domains which are traditionally dominated by feed-forward nets, trained with backpropagation algorithm. If output interpolating results deliver as good values as backpropagation — shown in this paper — it might be interesting to replace backpropagation by counter-propagation and self-organizing maps. This is advantageous because of stable convergence of the SOM, less risk of local minima, better interpretation tools for training result (local information storage) and fast training.

In this paper two completely different evaluation methods are used. The counter-propagation architecture, using a self-organizing map in the hidden layer and a feed-forward net, using backpropagation training. In the given type of problem, the counter-propagation net leads to better results, faster training and stable convergence. Comparison of both methods showed that convergence properties of the backpropagation algorithm make it difficult to use this type of properties in an industrial environment.

References

[1] J. Göppert and W. Rosenstiel. Topology-preserving interpolation in Self-Organizing maps. In *Proceedings of NeuroNimes 93*, pages 425–434, Nanterre, France, 10 1993, EC2.

[2] J. Göppert, H. Speckmann, W. Rosenstiel, W. Kessler, G. Kraus, and G. Gauglitz. Evaluation of spectra in chemistry and physics with kohonen's selforganizing feature map. In *Proceedings of NEURO NIMES 92*, pages 405–416, Nanterre France, 11 1992.

[3] Robert Hecht-Nielsen. Counterpropagation networks. In Maureen Caudill and Charles Butler, editors, *Proceedings of the IEEE First International Conference of Neural Networks*, pages II:19–II:32, Piscataway, NJ, 1987. IEEE.

[4] T. Kohonen. Self-Organized Formation of Topology Correct Feature Maps. *Biological Cybernetics*, pages 59–69, 1982.

[5] T. Martinez and Schulten K. A "neural gas" network learns topologies. In Kohonen et al., editor, *Proc. Int. Conf. Artificial Neural Networks*, pages 397 – 407, Espoo Finland, 6 1991.

[6] D.E. Rumelhard and J.L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognitron, I, & II.* MIT Press, Cambridge MA, 1986.