Second-Order Learning in Self-Organizing Maps

Ralf Der*[a]
Michael Herrmann†[b]

[a] University of Leipzig, Institute of Informatics
Postfach 920, 04109 Leipzig, Germany

[b] Max-Planck-Institute for Fluid Dynamics
Bunsenstr. 10, 37073 Göttingen, Germany

Kohonen's self-organizing map bears large potentials as a universal tool of nonlinear data analysis. From the practical point of view control parameters like the learning rate and the neighborhood width need special attention in order to exploit the possibilities of the approach. Our paper introduces second order learning methods which generalize the dynamics of Kohonen's learning algorithm in that control parameters are individually attributed to each neuron and adapted automatically. This is achieved by making use of the special properties of the map at phase transitions it undergoes when learning parameters cross critical values. We demonstrate by way of examples both the automatic control of the self-organization process itself, the extraction of principal manifolds, the mapping of hierarchically structured data, and provide also a version of the algorithm which proves feasible in the case of sparse data sets.

1. Introduction

The clever handling of control parameters plays an essential role in most learning algorithms. Manipulating the parameters in a convenient way not only may speed up the learning procedure itself but often is responsible for the success of the learning as such. In many learning procedures of practical interest finding the correct learning parameters or cooling strategies is done by trial and error. In a few cases, strong theorems are known which formulate the parameter strategy in an explicit way an example being the Robins-Monroe theorem for the case of stochastic gradient descent. Other examples may be found in [13].

Since these theorems refer to the asymptotic time behavior they are of limited value for practical applications. There an initial learning period is often decisive on whether a meaningful solution is approached during the convergence phase or not. For this reason a variety of empirical parameter-learning procedures has been invented. The self-learning of the learning parameters is called second-order learning and introduces another set of meta parameters which can however be assumed to be less sensitive to variations or are

*der@informatik.uni-leipzig.de
†michael@chaos.gwdg.de

to be controlled by higher-order learning. In this way the hope is that learning does not degenerate so much to a trial and error procedure.

Whereas the basic learning algorithms are usually guided by some error functional and perhaps further constraints the learning of learning parameters is intended to achieve additional qualities of the solution. This concern generalizability, plasticity, robustness etc. and allows to cope with problems such as over-fitting, convergence speed, non-stationary input distributions or outliers.

2. Second-order learning in SOMs

In the following we will focus on Kohonen's self-organizing feature map (SOM), cf. Ref. [11] for a comprehensive overview, where the control parameters are the learning rate ϵ and the neighborhood width σ. The learning rate controls the plasticity of the map since it defines the attention a neuron attributes to a single stimulus. By an individual tuning of the neural attention, the magnification factor of the map can be controlled locally, cf. Ref. [2] such as to achieve maps minimizing particular error criteria. In another approach [1] the plasticity of the map is improved by a heuristical scheme for choosing optmimal learning step lengths.

The neighborhood width σ governs both self-organization process and the topographic properties of the map. The self-organization of topological order is realized by a convenient cooling strategy which however is motivated only empirically. In contrast to the usual learning rates, σ must not be cooled down to zero. In fact, below a critical value the topographic structure of the map will break down. Interestingly this happens by a phase transition as shown in Ref. [7]. Even more so, the asymptotic value of σ is of crucial importance in one of the main applications of the SOM which is the dimension reduction of noisy data distributions. We may consider this case as the mapping of the neuron lattice into the input space to form there a curvilinear coordinate system representing the main features of the the higher dimensional data distribution.

The quality of this fit hinges on the value of σ in a sensitive way. With an appropriate value of σ the curvilinear coordinate system follows the data distribution as close as possible without fitting the noise. Finding this value may be done again by trial and error in slowly shrinking σ while permanently registering the preservation of topology by a convenient measure, cf. Ref. [14]. However, this approach will break down if the strength of the noise varies over the data distribution in an essential way. Then, one needs a local value of σ. In the present paper we discuss a new algorithm which handles the neighborhood width in a completely self-regulating way for both the process of topological ordering and the fine tuning of the local asymptotic values of σ in order to find the best fit to the data distribution. We will do this by exploiting the specific properties of Kohonen's learning algorithm at the phase transition between different regimes of the map. This way of reasoning can be carried over also to other learning procedures so that we understand our paper also as a contribution to the more general problem of second-order learning.

2.1. The algorithm

For the sake of simplicity we study the case of a chain, i.e. of a one-dimensional lattice of neurons to be mapped into a data distribution embedded into a higher dimensional input space. Apart from noise the data cloud is proposed to be one-dimensional so that the image of the chain may serve as a nonlinear principal component of the data distribution. Kohonen's update rule for the image $\mathbf{w}_r \in R^n$ of the neuron position (lattice site $r \in \{1, 2, \ldots, N\}$) is

$$\Delta \mathbf{w}_r = \epsilon h(r, r^*)(\mathbf{v} - \mathbf{w}_r), \tag{1}$$

where $\mathbf{v} \in R^n$ is the input vector and $h(r, r^*)$ denotes the neighborhood function as usual.

2.2. Individual neighborhood width

Usually the neighborhood width is defined globally for all neurons to be the same. In the present paper we consider the case that the noise is varying strongly over the data distribution so that the neighborhood width of the neurons is to be chosen individually in order to get a good fit of the principal curve to the data distribution. We define the neighborhood function as

$$h(r, r^*) = \frac{1}{\sqrt{2\pi}\sigma_r} \exp\left(-\frac{(r - r^*)^2}{2\pi\sigma_r^2}\right), \tag{2}$$

where σ_r is the local value of the neighborhood width to be defined below. The prefactor containing σ_r normalizes the influence of the current winner neuron r^* onto its neighbors. Without this factor the algorithm turns out to be less stable.

2.3. Phase transition dynamics

In order to find convenient values for the σ_r we exploit the dynamics of the phase transition from the topographic to the over-fitting situation, cf. [12,6]. For a discussion consider our case of mapping a chain of neurons into a data manifold of dimension higher than one. The algorithm tries to adapt the (image of the) chain to the data points as close as possible under the smoothness constraint defined by the value of σ.[3] While gradually shrinking σ, after the self-organization process the chain adapts closer and closer to the data points ending up with a complete match to all the data points which is the over-fitting catastrophe.

The point now is that there is a sharp phase transition to the over-fitting regime which occurs at a critical value σ^{crit} of the neighborhood width, σ^{crit} depending on the scattering of the data points about the principal curve [12,6]. At the phase transition point the quality of the map changes in that characteristic oscillations (folding) are formed. These are signaled by topology violations.[4] Although the question of measuring the topographic properties of the map is not trivial, cf. Refs. [4,14], we have found a simple criterion [3] which proved reliable in practical applications. We consider the distance $\alpha = \|r' - r''\|$ which is the distance between the first and second closest neuron to the current data point where $\alpha \geq 1$. If for any data point the first and second winner are not neighbors ($\alpha > 1$) than there is a violation of topology in the region of the data point which means that σ has fallen below its critical value. In other words $\alpha > 1$ signals a local onset of the phase transition to the over-fitting (topology violating) regime.

[3] We consider the case of a global value of σ for the moment.

[4] Actually, there is a coexistence between a topology violating and a topology preserving phase. The latter however is much less stable, cf. [6], and can be ignored for the present discussion.

2.4. Parameter learning dynamics

Consequently, our approach consists in keeping σ_r fluctuating around its (unknown) critical value σ_r^{crit}. This is done in two steps. On the one hand we decrement $\sigma_{\mathbf{r}}$

$$\Delta\sigma_r = -\frac{1}{NT_\sigma}\sigma_r \qquad \forall r \tag{3}$$

which is carried out at each step (presentation of data point) for all i and on the other hand we increase the σ_r locally if ever a violation of topology is registered, i.e. whenever $\alpha > 1$ we reset the local values of σ as

$$\sigma_r := \max\left(\sigma_r, \alpha\, K \exp\left(-\frac{2\,(r-R)^2}{\alpha^2}\right)\right), \quad \text{where } R = \frac{1}{2}\left(r' + r''\right)\ . \tag{4}$$

where K is an empirical factor. In our simulations we always choose $K = 2.4$.[5] As a result, the map fluctuates around the principal curve due to the phase transition taking place each time the phase barrier corresponding to the local critical value σ^{crit} is crossed.[6]

2.5. Fluctuation smoothing

In order to average over the fluctuations each neuron keeps a second pointer $\overline{\mathbf{w}}_r$ obtained by the moving average

$$\Delta\overline{\mathbf{w}}_r = \frac{1}{KNT_\sigma}\left(\mathbf{w}_r - \overline{\mathbf{w}}_r\right) \tag{5}$$

over the fluctuations, where K is of the order of 10. The $\overline{\mathbf{w}}_r$ provide in most cases a very good first order data model. Further improvements depend on the task. In the case of modeling a functional relationship one may use the $\overline{\mathbf{w}}_r$ to investigate the properties of the noise η in order to improve the model. For the principal curve case, an essential improvement consists in using the $\overline{\mathbf{w}}_r$ as starting positions for a final step in the sense of the iterative Hastie-Stuetzle algorithm [8]. This can be implemented more easily by monitoring directly the averages over the data in each domain. Hence, instead of $\overline{\mathbf{w}}_r$ each neuron gets a second pointer $\overline{\mathbf{v}}_r$ updated if the neuron is the winner as

$$\Delta\overline{\mathbf{v}}_s = \frac{1}{KT_\sigma}\left(\mathbf{v} - \overline{\mathbf{v}}_s\right). \tag{6}$$

The set $\{\overline{\mathbf{v}}_r \mid r = 1, \ldots, N\}$ is the final result of the algorithm, i.e. they represent the principal curve in input space. Several toy applications of the present algorithm may be found in the Figures.

3. Applications

We are now going to demonstrate the properties of the above algorithm by the results of numerical simulations. We have applied the above algorithms to map both one- and two-dimensional lattices into higher-dimensional input spaces with inhomogeneous data distributions of effective dimension $D = 1$ or $D = 2$, respectively. The local scattering of the data points around the central manifold varied by up to an order of magnitude.

[5] The value of K can be made plausible via the well know relation $\lambda = 2.02\tilde{\sigma}$ between the wave length of the critical folding and the width $\tilde{\sigma}$ of the neighborhood measured in input space.

[6] This is the main difference to related methods of self-learning the neighborhood width presented in [9,10].

3.1. The self-control of the self-organization process

In the usual scenario of the self-organization process one starts with a random initialization of the map and a high value of the (global) neighborhood width σ which then is subject to a cooling procedure. Usually this is done by an exponential decay like the one of Eq. (3) on a time scale T which is fixed empirically.

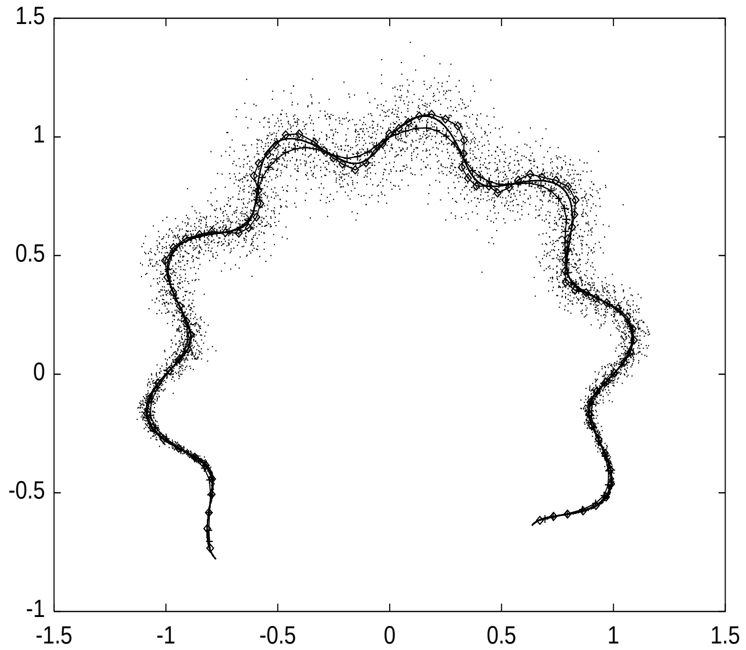


Figure 1. *Mapping a one-dimensional chain of neurons into a nonlinear pseudo-one-dimensional data distribution with the scattering width of the data points varying by one order of magnitude. We used our algorithm which adapts the individual neighborhood width of the neurons so that the chain (crosses) can follow the ideal principal curve (solid line). We started with a random initialization of synaptic weights and* $\sigma_r = 0 \quad \forall r$. *The squares denote the centers of the Voronoi cells. Parameters of the run were* $N = 100$, $\epsilon = 0.1$, $T_\sigma = 100$.

In our scenario one may start with arbitrary values for the individual neighborhood widths σ_r. In fact, we start with σ_r close to zero for all r. The random initialization of the synaptic vectors engenders long ranging topology violations so that the values of the σ_r are subsequently reset to large values due to the reset mechanism of Eq. (4). The decay time T_σ on its hand plays a role largely different from that of the conventional cooling procedure. Quite generally, the cooling is much more rapid, i.e. $T_\sigma \ll T$ so that T_σ may be much smaller than the average healing time of a topology violation of range (in neuron space) α. This is consistent because of the reset mechanism of Eq. (4) which keeps the σ_r at a level corresponding to the range of the pertinent topology violations. In other words the decrease of the σ values is governed by the decay of the average range α of the topology violations. This is clearly demonstrated in Fig. 2 where the time course of the values of σ_r for the map of Fig. 1 is given.

It is one of the peculiarities of our algorithm that the dynamics of the map is largely independent of the value of the control parameters, which is mainly the cooling time T_σ. In fact, in the example considered in Fig. 1 we could vary T_σ by several orders of

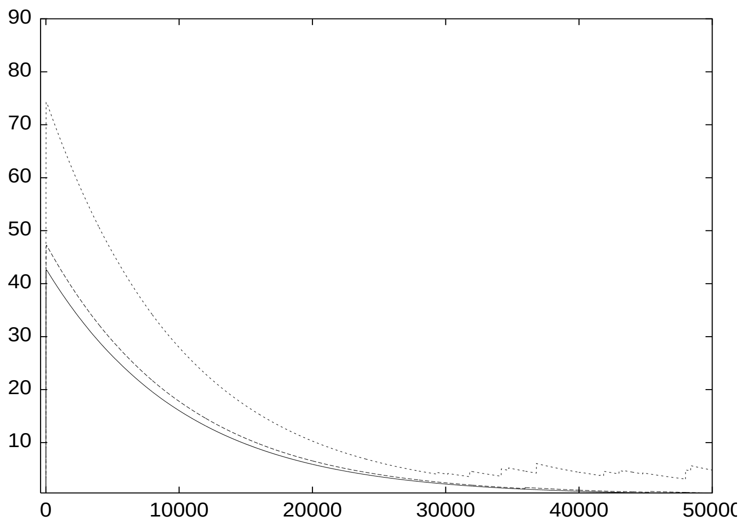


Figure 2. *Time course of the σ_r for the self-organization process of the map shown in the previous figure. Depicted are the neighborhood widths over time for neurons at $r = 1, 5, 50$ of a chain of 100 neurons. The upper curve is that of the neuron in the middle of the chain which converges to a large value of σ because of the large scattering width of the data points in its domain.*

magnitude without jeopardizing the stability of the learning procedure. This is what was to be expected of a meta-parameter like T_σ in a second-order learning scenario. A further discussion of the specific role of meta-parameters may be found in Sec. 3.3 below.

3.2. Revealing the structure of data distributions

One of the most prominent applications of the SOM is dimension reduction of noisy data. Mathematically this corresponds to the problem of extracting principal curves – principal manifolds (PM) in the general case – from higher dimensional data distributions. A PM is defined self-consistently by the requirement that each point on the PM is the average of the data points projecting to it, cf. [8]. Thus, it minimizes the mean square deviations of the data from the PM subject to some smoothness constraint. The smoothness condition restricts the curvature of the PM. It is the precise formulation of these criteria which makes the problem highly nontrivial.

Hastie and Stuetzle [8] described an algorithm for the construction of principal curves (PC) which works iteratively starting from the principal component of the data set. In each iteration a new estimate of the PC is obtained from the calculation of the centers of gravity of the data points with respect to the current estimate of the PC. This procedure is combined with a smoothing operation controlled via cross validation in order to avoid the over-fitting catastrophe. The authors gave some evidence in favor of the convergence to a stable solution, mainly by referring to the linear case. However, so far there are no general criteria for the existence and uniqueness of the PM for arbitrary data distributions.

In the Hastie-Stuetzle algorithm and other algorithms known so far the smoothness and hence the stability are guaranteed by local averaging, the span of the average being guided *globally* by cross validation. Our SOM based algorithm avoids this restriction and moreover leads to a stable though possibly suboptimal principal curve, see Fig. 1 for a demonstration. Moreover the new algorithm is not restricted to the case of principal *curves* but also has been successfully applied to the problem of two-dimensional PMs, see Fig. 3.

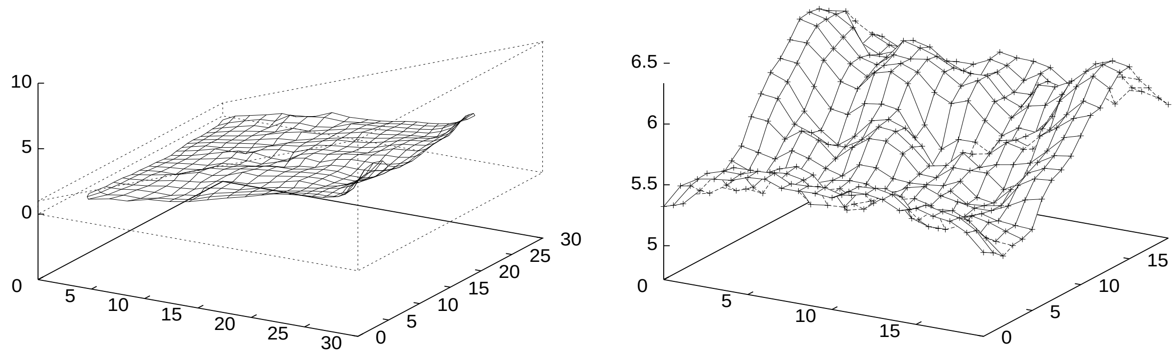


Figure 3. *Embedding a two-dimensional neural lattice in a three-dimensional data set with automatic learning of the individual values of* σ_r.

3.3. Hierarchically structured data distributions

In Sec. 3.1 we have emphasized the robustness of the parameter dynamics with respect to the meta cooling parameter T_σ. In the present section we want to demonstrate further potentialities of SOM second order learning. In the language of dynamical systems, with the parameter dynamics integrated the learning process of the map is a (stochastic) dynamical process with the asymptotic configuration of the map as an attractor. If the map is topology preserving one may say, that the map represents the principal structure (in the sense of a principal component) of the data distribution.

In the case of hierarchically structured data distributions like the one of Fig. 4 there are more than one principal components imaginable which so to say reveal the principal structure of the distribution as seen on different length scales.

The interesting point now is that our generalized map dynamics may converge to both of these principal components depending on the value of the meta cooling parameter. We may say that the learning dynamics is bistable or multi-stable in the general case in the sense that there are several attractors to which the dynamics may converge. The meta parameter in this case plays the role of a switch by which we can choose the attractor and hence the level of the structural hierarchy which is to be displayed by the map, an example being given by Fig. 4.

4. Sparse data sets

In the above algorithm we directly check the onset of the phase transition by monitoring the topology violations. This procedure works well if there are enough data points close to the boundaries of the data distribution, since it is mainly these data points which are not mapped to neighboring first and second winners. Therefore the above algorithm may

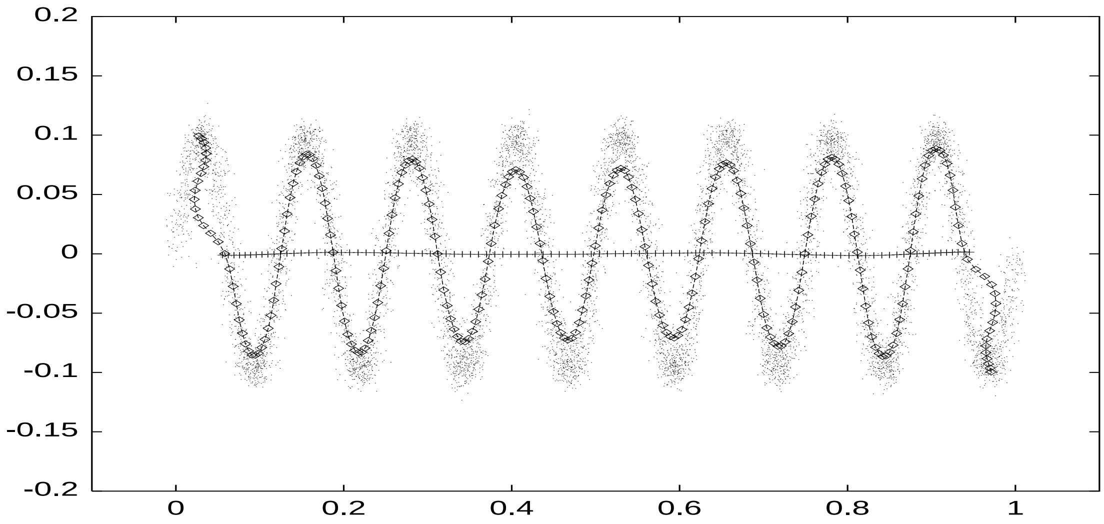


Figure 4. *Two alternatives of a principal component found by the algorithm by making use of its multi stability property. The data distribution is hierarchically structured in the sense that on a coarse length scale it is more or less a rectangle and on the fine scale we see a noisy* sin*-function. The algorithm finds both solutions for the principal curve problem. For a sufficiently large value of T_σ in Eq. (3) the straight line solution is obtained as a stable final state of the learning procedure. If T_σ is chosen at least one order of magnitude smaller the map converges with the same initialization to represent the fine structure of the data distribution. Both representations are topology preserving solutions.*

fail if the number of data points is small. For this case, a very sensitive criterion for the emergence of the critical fluctuations was found to be a frequency adapted wavelet transform [5] of the map. For a one-dimensional SOM we calculate for each neuron r the Gabor transform

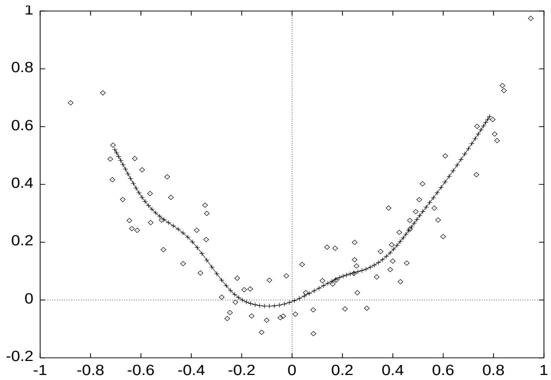
$$g_r = \frac{1}{\sqrt{2\pi}u_r} \left\| \sum_{k=1}^{N} \mathbf{w}_r \, \exp\left(\frac{-(k-r)^2}{2u_r^2}\right) \exp\left(-i \; k\omega_r\right) \right\| \tag{7}$$

where both the frequency ω_r of the kernel and the width are functions of the current values of σ_r so that the wavelength of the kernel always does agree with that of potential foldings. At the critical point $\sigma = \sigma^{\mathrm{crit}}$ the wavelength of the emerging folds is $\lambda = 4.04\sigma l$, where l is the average distance between the neurons in that region, cf. [12]. Choosing $\omega_r = u_r = 4\sigma_r$ causes g_r to jump by an order of magnitude when σ_r crosses σ_r^{crit}. Hence, g_r is the desired sensitive criterion for detecting the onset of the phase transition.

In the algorithm we use (3) as before. If for the winner g_r exceeds a small threshold we use $\alpha = \kappa\sigma_r$ in (4), observing $1 \leq \alpha \leq \sigma_{\max}$, where $\kappa = 1.2$ is an empirical factor. In the simulations a control of κ is obtained from monitoring the fluctuations of σ which optimally should stay in the region of a few percent.

5. Concluding remarks

In the present paper we discussed new algorithms devoted to the general problem of learning the parameters of learning. For the case of Kohonen's feature map we have



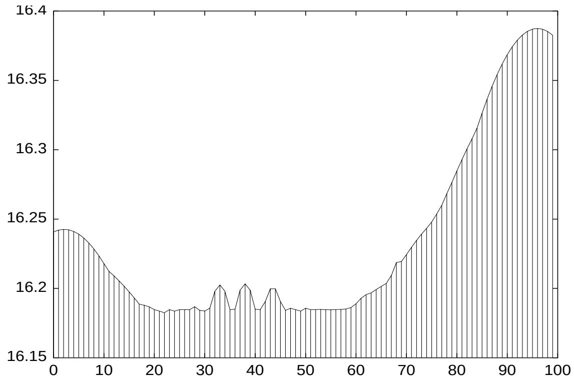


Figure 5. *Using wavelet transform to control the neighborhood widths in a sparse data set. A data set of 80 points and a chain of 100 neurons are chosen. Display of the emerged map (left) and σ_r-values (right).*

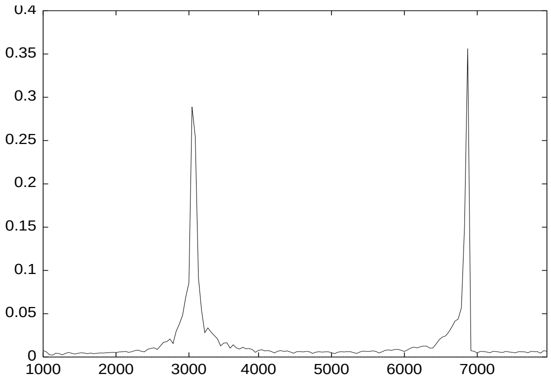


Figure 6. *Time course of the wavelet transform of one of the neurons during 7000 steps.*

demonstrated by way of some toy examples that the algorithms are stable and depend not too sensitive on the meta-parameters like the cooling rate T_σ of the local values of σ. We have studied the algorithm also in a few applications with real world data and found the same behavior. We may therefore conclude that our parameter adaptation procedure makes Kohonen's algorithm an even more general tool for nonlinear data analysis.

REFERENCES

1. L. L. H. Andrew and M. Palaniswami. A unified approach to selecting optimal step lengths for adaptive vector quantizers. *IEEE Transactions on Communications* 44(4):434-439, 1996.
2. H.-U. Bauer, R. Der, and M. Herrmann. Controlling the magnification factor of self-organizing feature maps. *Neural Computation*, 8(4):757–771, 1996.
3. H.-U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. To appear in *Neural Networks*, 1999.
4. H.-U. Bauer and K. R. Pawelzik. Quantifying the neighborhood preservation of Self-Organizing Feature Maps. *IEEE Trans. on Neural Networks*, 3(4):570–579, 1992.

5. C. K. Chui. *An Introduction to Wavelets*, volume 1 of *Wavelet Analysis and its Applications*. Academic Press, Inc., 1992.
6. R. Der and M. Herrmann. Critical phenomena in self-organizing feature maps: A Ginzburg-Landau approach. *Phys. Rev. E*, 49(5):5840–5848, 1994.
7. R. Der and M. Herrmann. Instabilities in self-organizing feature maps with short neighborhood range. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'94)*, pages 271–276, 1994.
8. T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.
9. M. Herrmann. Self-organizing feature maps with self-organizing neighborhood widths. In *Proceedings IEEE Int. Conf. on Neural Networks (ICNN'95)*, IEEE Service Center, Piscataway, NJ, pages 2998-3003, 1995.
10. K. Kiviluoto. Topology preservation in self-organizing maps. In *ICNN 96. IEEE International Conference on Neural Networks*, volume 1, pages 294–9. IEEE, New York, 1996.
11. T. Kohonen. *The self-organizing map*. Springer, 1995.
12. H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wesley, Reading, MA, 1992.
13. V. N. Vapnik. *The nature of statistical learning theory*. Springer, 1995.
14. T. Villmann, R. Der, M. Herrmann, and T. M. Martinetz. Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256–266, 1997.