ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 59-72

Mathematical Aspects of Neural Networks

B. Hammer[1] and T. Villmann[2]
[1] Dept. of Math./Comp.Science, University of Osnabrück, Germany
hammer@informatik.uni-osnabrueck.de,
[2] Clinic for Psychoterapy, University of Leipzig, Germany,
villmann@informatik.uni-leipzig.de

Abstract. In this tutorial paper about mathematical aspects of neural networks, we will focus on two directions: on the one hand, we will motivate standard mathematical questions and well studied theory of classical neural models used in machine learning. On the other hand, we collect some recent theoretical results (as of beginning of 2003) in the respective areas. Thereby, we follow the dichotomy offered by the overall network structure and restrict ourselves to feedforward networks, recurrent networks, and self-organizing neural systems, respectively.

1 Introduction

Many good neural algorithms are designed using mathematics or they are formulated in terms of mathematics. Some good neural models can be accompanied by a rigid mathematical investigation. And we would be happy to find a rigid mathematical analysis for other good neural models for which a theoretical investigation has not yet been possible. Hence this paper should, per definition, cover a huge amount of existing and future papers on neural networks. Since this is obviously impossible, we restrict ourselves to a subjective choice of classical results which we believe are of importance. In addition, we incorporate pointers to recent results in the literature. We will tackle classical neural models used for machine learning: feedforward networks, recurrent architectures, and self-organizing systems, neglecting more recent models like cellular networks or spiking models [96, 130], statistical counterparts like Gaussian processes or Bayes point machines [53, 64], and other learning scenarios like reinforcement learning [128]. Focusing on machine learning tasks, we further neglect neural architectures which model biological neural networks or cognitive systems.

Naturally, mathematics is introduced in neural networks literature for different aims: some mathematics directly yields efficient and well founded learning algorithms like support vector machines (SVMs) [24]; some mathematics tries to explain effects of training or to achieve guarantees for training – and often finally fails to describe the *initial* setting like in the case of the loading problem for feedforward networks [34] or the convergence problem of the self-organizing map [67]; some mathematics is done for esthetical reasons (people do often not agree on which papers fall within this category); and finally some rare mathematics describes the real life [85]. In this article, we will include mathematical results that help to understand the respective setting regardless of their direct practical applicability.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 59-72

2 Feedforward networks

Feedforward networks (FNNs) compute a possibly highly nonlinear function $f_W : \mathbb{R}^n \to \mathbb{R}^o$, composing simple functions provided by the single neurons which are connected in a directed acyclic graph. Thereby, the single neurons typically compute a function of the form $\boldsymbol{x} \mapsto \sigma(\boldsymbol{w}^t \boldsymbol{x})$ for sigmoidal networks, or $\sigma(|\boldsymbol{w} - \boldsymbol{x}|)$ for radial basis function networks, where $\sigma : \mathbb{R} \to \mathbb{R}$ is the activation function and $\boldsymbol{w}$ are the neuron's weights. f_W is parameterized by the collection of all neurons' weights W. FNNs can be used to learn an unknown regularity f based on a couple of examples $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i)) \mid i = 1, \ldots, m\}$. Commonly, training takes place in three steps: selection of the neural architecture and the hyperparameters of training, mostly done by crossvalidation; optimization of the weights on the given training set via error minimization, often done by some gradient methods; and estimation of the generalization ability using the test error on a test set not taken for training. Often, training and further optimization of the architecture are mixed, e.g. using growing and pruning algorithms or training with additional regularization. In the case of SVM, training is even explicitly formulated in terms of optimizing architectural quantities, the margin, under the constraint that the given data points are mapped correctly. The standard way of training poses three mathematical questions which are to be answered for the respective neural architecture and training algorithm: 1) What are the approximation capabilities of the respective architecture? If the neural architecture is not capable of approximating the (unknown) function to be learned then we cannot, in principle, find an appropriate architecture to start training. 2) How do good error minimization algorithms look like and what is the complexity of training? We have to find guarantees for the convergence of the algorithm and we should investigate the expected runtime. And finally: 3) Can the generalization ability of neural architectures be guaranteed such that we can expect adequate behavior for future data we would like to predict?

2.1 Approximation properties

The universal approximation property of FNNs with various activation functions has been established in [70, 102, 116], for example. In general, one hidden layer is sufficient for approximating any continuous or measurable function, respectively, up to any desired degree of accuracy on compact sets. Hence the search space for appropriate architectures in a learning task can be limited to one or two hidden layers. If only a finite number of points is given, the number of neurons sufficient for interpolation can be limited, as demonstrated in [124]. Note that the proofs are, at least in theory, constructive. This fact is used in [81] to design an alternative (though possibly not yet very efficient) training algorithm. An extension of the universal approximation result to networks with functional inputs is presented in [111]. Recently, the universal approximation capability of SVMs has been established, too [57, 127]. However, achieving a small margin is thereby not possible for a large number of concepts [14] such that the capacity of the architecture cannot be bounded.

Apart from the in principle guarantee that neural networks can approximate any reasonable function and apart from concrete bounds for finite training sets, approximation rates are of particular interest. They characterize the quality of the approximation which can be achieved if a function (possibly with additional constraints) is approximated by n neurons. One of the first results in this direction can be found in [5, 73] where convergence of order $1/n$ is established for functions with limited norm

which is measured incorporating the activation function. Generalizations thereof can be found in [83, 86], deriving e.g. dimension independent geometric convergence of neural networks for specific function classes. Starting from these results, further mathematical questions concerning the capacity of possibly restricted architectures can be investigated, such as the capacity of FNNs with restricted weights [42], the uniqueness of parameters [44], or the design of alternative transfer functions for FNNs respectively kernels for SVMs [43, 72]. A problem adapted number of neurons can be achieved with alternative techniques which increase the number of neurons during training, such as proposed for RBF networks by Li, Luo, and Qi in this volume.

2.2 Complexity of training

Usually, network training aims at finding weights such that the given data set is mapped correctly to the desired output values. FNNs are often trained by different gradient descent methods, for which convergence can be established [82]. However, the algorithm might get stuck in local optima of the error function. Hence, the question of the in principle complexity of neural network training arises. This loading problem is a prime example for a process where mathematics tries to get nearer and nearer to scenarios as they occur in practical problems, without actually achieving this goal up to now. It follows directly e.g. from [99] that training a fixed FNN with the Heavyside activation function can be done in polynomial time. However, most existing algorithms do not take the specific architecture into account. Hence the loading problem should be considered in a more general setting taking arbitrary architectures as input. Starting with the work [20, 75], it is known that general neural network training is NP-hard. More precisely, the paper of Blum and Rivest [20] states the fact that the loading problem is NP-hard for multilayer architectures with the Heavyside activation function even if the number of hidden neurons is fixed to two and only the number of inputs is allowed to vary from one instance to the next one. Training a single perceptron is, of course, polynomial as a specific instance of linear programming, although the (convergent) perceptron algorithm might take exponential time. However, achieving optimum solutions in the presence of errors is an NP-hard problem even for the simple perceptron as shown e.g. in [69].

People have argued that these situations are not realistic with respect to several aspects: usually, the sigmoidal function and not the perceptron function is considered. Moreover, good solutions instead of optimum ones would be sufficient. The number of hidden neurons in the networks is usually correlated to the number of available training samples. Hence a couple of results try to generalize the setting to larger architectures [55, 104], sigmoidal networks [55, 74, 120], or approximate settings [6, 34] establishing NP-hardness even for the seemingly simplest training problem within this line, the training of a single sigmoidal neuron [121]. As a consequence of this list of NP- results, researchers try to design or identify specific and possibly restricted learning scenarios where polynomial bounds on the training time can be guaranteed [26]. In addition, focusing on large margins might help for training FNNs [15]. Note that SVM training can be written as a quadratic optimization problem with very simple constraints, such that, unlike FNNs, SVM training is polynomial. Nevertheless, the original algorithm needs access to all training pattern. Hence alternative training methods for large scale problems are investigated in the literature even for SVM such as efficient online training methods or decomposition schemes [47, 52, 90].

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 59-72

2.3 Learnability

It should be mentioned that error minimization often already takes the generalization ability of the final network into account: Incorporation of statistical interpretation and training via Bayesian inference, for example, gives (approximately) optimum values [122]. Regularization terms may be added to the error function such that robust solutions are found [17, 142]. The SVM explicitly solves a regularization task, formulating the correctness of training data as a constraint, to name just a few examples. However, after obtaining a small training error, we are now interested in the generalization ability of the trained networks to new examples, together with mathematical guarantees for this property. The question occurs of how such guarantees can be stated in mathematical terms. There exist several different formalisms within this area. Methods of statistical physics, for example, allow to compute learning curves of simple iterative training rules which quantize the average learning effect after presenting a number of examples [101, 123]. One very popular formalism for neural networks training is offered by the notion of PAC (probably approximately correct) learnability as introduced in [133] and the mathematical counterpart of uniform convergence of empirical risks as introduced in [136].

PAC learnability states the fact that at least one learning algorithm (e.g. error minimization) can be found such that the probability of poor outputs of the learning algorithm, i.e. networks which do not generalize to unseen examples, approaches zero if enough training data are available. Uniform convergence of the empirical error of a function towards the real error on all possible inputs guarantees that *all* training algorithms which yield a small training error are PAC. In [136] uniform convergence for a given function class is connected to the capacity of the considered function class and concrete bounds on the training error are derived. The capacity of the function class is thereby measured in terms of the so-called VC-dimension. Remarkably, a function class is PAC learnable if and only if the capacity in terms of the VC dimension is finite. Starting from these results, the in principle generalization ability of network training can be established estimating the VC dimension of neural architectures. Different directions of research continue these results: The VC dimension of various neural architectures has been estimated, in some cases based on advanced mathematical methods [117, 118, 125]. Note that the VC dimension quantizes the capacity of the respective architecture, hence it can be used to investigate the approximation capability of networks, too [56]. Generalizations of the original setting to general outputs and various loss functions have been derived [1, 138]. Moreover, the bounds obtained via this general setting are tight in the limit, but they do not yet lead to useful bounds for realistic scenarios and training sets. Hence refinements and possibilities to take specific knowledge of the concrete setting or training algorithm into account as well as alternative statistical estimations are a topic of ongoing research [2, 7, 65, 150].

3 Recurrent networks

Recurrent networks (RNNs) combine neurons in a possibly cyclic graph such that time dependent dynamics can be observed, all neurons computing their output based on the activations in the previous time step: $x_i(t) = \sigma(\boldsymbol{w}_i^t \boldsymbol{x}(t-1))$ for discrete time or $\dot{x}_i(t) = \sigma(\boldsymbol{w}_i^t \boldsymbol{x}(t))$, respectively, for continuous time. Depending on the respective dynamic properties, RNNs are used for sequence prediction, transduction, and

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 59-72

generation, as associative memory, or for computation tasks such as binding, grouping, or cost minimization. Training RNNs for an unknown function f for possibly sequential data based on given training data can essentially be formulated like FNN training: selection of the architecture, optimization of the empirical error on the given training data, and estimation of the generalization ability on a test set. Hence the same mathematical questions as in the case of FNNs arise for RNNs.

The temporal structure of RNNs adds further mathematical questions in particular if RNNs are used for alternative tasks which rely on the computation capabilities and dynamic properties of these networks. We name just a few further aspects: What is the overall dynamic behavior of the network with respect to convergence and stability? Can the capacity of RNNs used as associative memories be estimated? Which different long-term behavior can be realized with a network? We will here only shortly mention mathematical aspects of RNNs and refer the reader to the tutorial paper [59].

3.1 Dynamic behavior

Depending on the respective task, the overall dynamic behavior of an RNN is of major importance. Many applications require that the network converges in some sense: if used as associative memory, it should converge to a stable state; for robotics, convergence to a possibly cyclic attractor might be appropriate; for control tasks, stability of the RNN should be guaranteed. Hence stability and convergence constitute one major issue investigated for different kinds of RNNs. Using Lyapunov functions, global convergence to a stable state can be established for the classical Hopfield network which restricts weights to a symmetric weight matrix. The stability of more general networks as well as convergence rates is investigated e.g. in [27, 28, 103, 146]. Conditions on the weight matrix for local stability of RNNs via linear matrix inequalities have been established in [89, 126]. Note that training approaches can use these conditions to design networks with appropriate dynamic behavior.

3.2 Capacity

RNNs might be used to approximate a continuous or measurable function on time series based on given training data. With respect to this aspect, RNNs are universal approximators on a finite time horizon in the discrete as well as the continuous scenario [51, 116]. Moreover, their approximation capability as operators is investigated e.g. in [4]. Upper bounds on the number of neurons which are sufficient to interpolate a given finite training set can be established [55]. Related questions such as uniqueness of weights constitute an interesting topic of research [77]. Turning from a limited time horizon to the long-term behavior, one can on the one hand relate RNNs to classical symbolic mechanisms like Turing machines or, in restricted scenarios, definite memory machines [61, 119]. On the other hand, their rich behavior as dynamic systems which are capable of producing stable, periodic, and chaotic behavior has been demonstrated e.g. in [131, 143].

If RNNs are used as associative memories, the notion of capacity refers to the number of patterns which can be stored in an appropriate RNN as stable states. This number, of course, depends on the characteristics of the pattern. Sparse or nearly orthogonal pattern can usually more easily be stored. In addition, this depends on the respective RNN model which is considered [25]. Interestingly, the notion of Lyapunov functions for specific network architectures such as Hopfield-type networks allow to

inject optimization problems into RNNs. One classical example for this procedure has been done with the TSP: if the weights of a Hopfield network are chosen appropriately, the global energy minima of the Hopfield network correspond to solutions of the TSP. Various different optimization problems can be tackled in this way: Hopfield type networks for the TSP [33, 129], RNNs for invariant recognition of geometrical objects [88], for graph coloring [41], or, as proposed by Jain and Wysotzki in this volume, RNNs for solving graph isomorphisms respectively representing automorphisms of structures via the stable states in the network.

3.3 Learning

If used for function approximation, RNN training is in principle identical to FNN learning: the empirical error is minimized on a given training set e.g. with a gradient descent method. Naturally, all NP-hardness results for FNN training transfer directly to RNNs, such that difficulties can be expected in some situations. In addition, it has been proved in [16] that gradient methods do not seem appropriate if long term dependencies are to be learned: it is impossible to latch information over a long time period. Hence the question of efficient training algorithms for RNNs is still an open topic of research. The exponential decrease of gradient information is used for a particularly efficient gradient truncation in [3]. The approach of Schiller and Steil in this volume investigates a different alternative for RNN training, which does not follow the gradient but starts from an alternative formulation of RNN training as constraint optimization problem. Interestingly, this yields linear interior weight updates.

The generalization capability of RNNs constitutes another not yet satisfactorily solved research problem. Since the VC dimension of RNNs depends on the given inputs [80], distribution independent PAC learnability cannot be guaranteed in principle. However, the articles [55, 59] provide an alternative way to derive distribution dependent bounds or to derive posterior generalization bounds, respectively. These results justify the in principle learnability of RNNs in an appropriate sense. Unfortunately, the derived bounds are far from being useful in practical scenarios. Additional constraints on the networks might help in this setting [61].

If RNNs are used as associative memory, training algorithms can be based on stability constraints. The contribution [144] derives a new algorithm for stochastic Hopfield networks. Extensions to learn more complex pattern such as sequences, as well as extensions to more complex network structures are currently investigated in the literature [87, 145].

4 Self-organizing networks and vector quantization

Neural networks for vector quantization (VQ) comprise a broad variety of models ranging from statistically motivated approaches to strong biologically realistic models. The latter ones should not be discussed here, we refer to the respective community. We focus on those models which are developed for data processing. The main task of these approaches is to describe given data in a faithful way such that the main properties are preserved as good as possible. This properties could be the probability density [134], the shape of data in the sense of possibly non-linear principle component analysis (PCA) [100, 112] or visualization like multi-dimensional scaling (MDS) [106], topology preserving mapping [141], the usual reconstruction error, the classification error etc. For the different goals several approaches exist, whereby we have to

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 59-72

differentiate according to the type of adaptation between supervised and unsupervised learning schemes.

4.1 Unsupervised models

The goal of unsupervised VQ is to approximate the data $\mathbf{v} \in V \subseteq \mathbb{R}^{D_V}$ by a substantially smaller set $\mathbf{W}$ of reference vectors $\mathbf{w_r}$ (codebook vectors). Thereby a data vector $\mathbf{v} \in V$ is coded by the reference vector $\mathbf{w_{s(v)}}$ the norm $\delta = \left\|\mathbf{v} - \mathbf{w_{s(v)}}\right\|$ of which is minimum compared to all elements of $\mathbf{W}$. Let A be the index set of $\mathbf{W}$, i.e. we have a unique mapping between $\mathbf{W}$ and A. Then we can take the coding procedure as a mapping from the input space V to A: $\Psi_{V \to A} : \mathbf{v} \mapsto \mathbf{s} = \mathrm{argmin}_{\mathbf{r} \in A} \left(d\left(\mathbf{v}, \mathbf{w_r}\right)\right)$ where $d\left(\mathbf{v}, \mathbf{w_r}\right) = \|\mathbf{v} - \mathbf{w_r}\|$is based on an appropriate norm. Usually the Euclidean norm is chosen. The crucial point in VQ is how one can find a good codebook $\mathbf{W}$. From a mathematical point of view an appropriate criterion is the expectation value of the squared reconstruction error $E\left[\mathbf{W}\right] = \int \|\mathbf{v} - \mathbf{w_s}\|^2 \mathcal{P}\left(\mathbf{v}\right) d\mathbf{v}$ where $\mathcal{P}\left(\mathbf{v}\right)$ is the probability density of the data distribution of the data vectors. The Linde-Buzo-Gray-algorithm (LBG or *k-means*) constitutes one basic approach [93]. Stochastic realizations as neural networks [67] use the convergence theorems from Kushner&Clark or Ljung [95, 84]. One main aspect is the convergence improvement by neighborhood learning as it occurs in biological neural maps. A very popular algorithm is the Self-Organizing Map (SOM) introduced by Kohonen in [78]. Beside a huge amount of models derived from the original one (for an overview see [79]), the SOM also inspired other (neighborhood oriented) VQ schemes.

4.1.1 The Self-Organizing Map

The main feature of the SOM is the topological structure in the index set A[1] and the neighborhood learning based on it to achieve a topographic mapping. The treatment of the simple adaptation process is mathematically very difficult [31]. Most results have only been established for the one-dimensional case. Thereby, mathematical question include the topics of 1.) convergence and ordering, 2.) topology preservation and 3.) probability density matching and magnification.

Convergence and ordering: For continuous inputs Ritter and colleagues investigated the stationary state and convergence properties after ordering under the assumption of a continuous index set A, the results are summarized in [109]. Erwin et al. have proved that it is impossible to associate a global potential function to SOM for continuous inputs and studied the role of the neighborhood function [45, 46]. Thereby, the learning is taken as Markov process [66]. An intuitive straightforward definition of a potential function to be minimized, derived from the usual SOM update rule, leads to a redefinition of the winning unit which now takes the neighborhood into account [67]. Before convergence in SOM, an ordering process takes place. The ordering conditions are investigated in [22]. For discrete index A the first proof of ordering and convergence under certain conditions was given in [113] by Sadeghi, whereby the SOM was considered as a Robbins-Monro-algorithm [110] and the respective differential equations are shown to be absorbing. Further ordering theorems for several,

[1] The elements of A are called neurons.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 59-72

more general, parameter settings are studied by Cottrell, Flanagan, Fort, Pagé and colleagues, verifying the almost sure convergence in dependence of the concrete choice of the neighborhood shape and range, learning rate etc. A review of the results can be found in [31]. Meta-stable states during SOM-learning may occur for certain configurations (non-vanishing learning rate) [50]. Sufficient conditions for convergence are given in [48]. Lebesque continuous inputs are studied in [49], discrete inputs distributions in [92]. For the higher-dimensional cases results can be found in [31, 91, 114]. Depending, for instance, on grid configurations non-stable situations may occur [31]. Moreover for short range interactions instabilities may be observed in an initial ordered configuration [37]. The time behavior of ordering is considered in [38] based on an analysis of the dynamical spectral density of the weight vectors.

Topology preservation: The definition of an ordered state depends on the definition of what is topographic (or topology preserving) mapping $\Psi_{V \to A}$. A mathematical exact definition is given in [141]. For this purpose properly chosen topological spaces in both the input space V as well as the output space A are defined. If the map Ψ and its inverse, but now taken as mapping between the respective topological spaces, are both continuous the SOM is called topology preserving. Several measures have been established to judge the degree of topology preservation. Thereby, the topographic function follows the exact definition [141]. However, the topographic function takes much computational effort. Although not based on the mathematical exact definition, the topographic product [10] and its derivatives [139] seem to be the best tools for practical use [9].

Violations of topology preservation do not only arise because of convergence problems: if the lattice dimension D_A differs from the effective data dimension $D_{\text{eff}} \leq D_V$ topological mismatches occur. The respective theory of meta- and instable states is initially based on Fokker-Planck approaches [109]. Further studies also use the Ginzburg-Landau-theory to describe the phenomena in more detail [36]. The high-dimensional analysis was pointed out in [11] using phase diagrams.

To overcome the topological mismatch problem growing variants of SOM have been developed [12] or input pruning was tried [21]. Structure adaptation is closely related to the connection of SOM and PCA [35] and its non-linear extension of principal curves [63]. Ritter has shown that (in case of topology preserving mapping) SOM can be taken as an approximation of principal curves [109].

Probability density matching and magnification: As mentioned above, the SOM is not an optimal vector quantizer in the sense of the error $E[\mathbf{W}]$ [149]. Deviations are due to the incorporation of neighborhood learning and topology preservation [31, 141]. This fact leads to a different magnification of the SOM in comparison to the usual VQ [39, 107]. Therefore, several modifications of SOM exist to achieve optimal magnification [40] or, more generally, to control the magnification by local learning rates according to local estimates of the data probability density [8]. In the winner relaxing magnification control the local learning based on density estimation problem is substituted by adding relaxing terms in the learning rule [30]. However, due to stability problems, here only positive magnification can be achieved with increasing instabilities if magnification approaches zero. SOM and maximum mutual information with respect to additional knowledge (auxiliary data) is discussed in [76] using metric adaptation to minimize the Kullback-Leibler-divergence between the auxiliary data space and the weight vector density.

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 59-72

4.1.2 Further VQ schemes

Certainly, SOM itself has been generalized to deal with various alternative scenarios such as variants for time-series [132, 137, 147] or more general data structure [58], alternative grid topologies [108]. An approach to include statistical properties into SOM learning is given by contingency anlysis as studied in detail by Cottrell&Letremy in this volume. For an overview of SOM extension as well as applications we refer to [79].

Many further schemes of VQ were developed. Historically earlier but closely related to SOM is the Elastic Net with well defined mathematical properties (convergence) and biological motivation [148]. A further topographic mapping scheme is the Generative Topographic Mapping [18], based on a constraint Gaussian mixture model, the parameters of which are determined by a maximum likelihood procedure using a simplified Expection-Maximum-principle (EM) procedure which is more complicate than SOM learning. The magnification for both is not optimal in the sense of maximum entropy [19, 29]. Furthermore, problems in the convergence of the EM approach may occur in case of outliers (see Archambeau, Lee & Verleysen in this volume). Linsker proposed a topographic VQ network with optimal information transfer [94] based on information theoretic learning [105]. Equiprobabilistic topographic map formation based on kernel methods are extensively studied by van Hulle [134] also under the constraint of maximum entropy [135].

Vector quantization based on potential dynamics are of great interest because of their clear mathematical treatment, for instance based on the EM [23]. Such topographic approaches are the stochastic topographic mapping and its variants [54, 67, 135]. Thereby, the basic mathematical trick is the mean field approximation to derive the EM-steps. However, the neuron lattice remains a hypercube. A fundamental alternative is the Neural Gas (NG) or its extension the Topology Representing Network. It combines the advantage of a potential dynamic with an otptimum topology preserving mapping. The potential dynamic is according to a diffusing gas, whereby the neighborhood range of the neurons plays the role of the virtual temperature [98]. The potential dynamic ensures convergence as well as stability. To achieve topology preservation, the connections between the neurons are adaptively determined based on the non-vanishing intersection of the receptive fields [97]. Moreover, the NG yields the same magnification as the usual VQ and a control scheme can be established in complete analogy to SOM by local learning rates [140] and winner relaxing terms, see Claussen&Villmann in this volume.

Another type of unsupervised models is due to blind source separation or independent component analysis [13]. The main feature is the determination of *statistically independent* sources of a mixed high-dimensional signal time series using higher moments, and, in particular the kurtosis. A comprehensive overview is given in [71]. For further unsupervised VQ methods we refer to the text book [68].

4.2 Supervised Methods

Self-organizing learning can also be used for supervised task where input-output pairs are given and the goal is to minimize the classification. Naturally, the same questions as for supervised feedforward and recurrent neural networks can be stated for these models. Since the methods are based on the self-organizing paradigm, further aspects

arise such as the question of potential functions of the dynamics or the possibility to include neighborhood cooperation.

Popular models are the learning vector quantizers (LVQ) proposed in [79]: LVQ1, LVQ2 and LVQ3 which try to minimize the classification error (CE). Thereby, LVQ optimizes the class margins [32]. However, the CE does not give a potential descent learning dynamic. Moreover, instabilities may occur [62]. A natural extension to achieve this goal is the Generalized LVQ (GLVQ) [115] which slightly modifies the classification task to ensure the potential property. GLVQ pushes the classification borders near to the optimum Bayesian decision. To improve the GLVQ classification a metric adaptation can be introduced in GLVQ thereby preserving the potential dynamic and including metric adaptation in the potential dynamic [62]. A combination of relevance learning in GLVQ and neighborhood cooperation has been recently developed [60]. The neighborhood cooperation thereby reduces the problem of local minima and accelerates the convergence.

References

[1] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[2] A. Antos, B. Kegl, T. Linder, and G. Lugosi. Data-dependent margin-based generalization bounds for classification. *Journal of Machine Learning Research*, 3:73–98, 2002.

[3] A. Aussem. Sufficient conditions for error backflow convergence in dynamical recurrent neural networks. *Neural Computation*, 14:1907–1927, 2002.

[4] A. Back and T. Chen. Universal approximation of multiple nonlinear operators by neural networks. *Neural Computation*, 14:2561–2566, 2002.

[5] A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39:930–945, 1993.

[6] P. Bartlett and S. Ben-David. Hardness results for neural network approximation problems. *Theoretical Computer Science*, 284:53–66, 2002.

[7] P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[8] H.-U. Bauer, R. Der, and M. Herrmann. Controlling the magnification factor of self–organizing feature maps. *Neural Computation*, 8(4):757–771, 1996.

[9] H.-U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12(4–5):659–676, 1999.

[10] 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.

[11] H. U. Bauer, M. Riesenhuber, and T. Geisel. Phase diagrams of self-organizing maps. *Physical Review E* , 54(3):2807–10, 1996.

[12] H.-U. Bauer and T. Villmann. Growing a Hypercubical Output Space in a Self–Organizing Feature Map. *IEEE Transactions on Neural Networks*, 8(2):218–226, 1997.

[13] A. Bell and T. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[14] S. Ben-David, N. Eiron, and H. Simon. Limitations of learning via embeddings in Euclidian half-spaces. *Journal of Machine Learning Research*, 3:441–461, 2002.

[15] S. Ben-David and H.-U. Simon. Efficient learning of linear perceptrons. In *NIPS'2000*, pages 189–195. 2000.

[16] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE TNN*, 5(2):157–166, 1994.

[17] C. M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7:108–116, 1995.

[18] C. M. Bishop, M. Svensen, and C. K. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21(1):203–224, 1998.

[19] C. M. Bishop, M. Svensen, and C. K. I. Williams. Magnification factors for the SOM and GTM algorithms. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4–6*, pages 333–338, 1997.

[20] A. Blum and R. Rivest. Training a 3-node neural network is NP-complete. *Neural Networks*, 9:1017–1023, 1988.

[21] T. Bojer, B. Hammer, M. Strickert, and T. Villmann. Determining relevant input dimensions for the self-organizing map. In *Proc. International Conf. on Neural Networks and Soft Computing (ICNNSC)*, Lecture Notes in Computer Science, page to appear. Springer Verlag, 2003.
[22] M. Budinich and J. G. Taylor. On the ordering conditions for Self-Organizing Maps. In M. Marinaro and P. G. Morasso, editors, *Proc. ICANN'94, International Conference on Artificial Neural Networks*, volume I, pages 347–349, London, UK, 1994. Springer.
[23] J. Buhmann and H. Kühnel. Vector quantization with complexity costs. *IEEE Transactions on Information Theory*, 39:1133–1145, 1993.
[24] C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
[25] B. Caputo and H. Niemann. Storage capacity of kernel associative memories. In J. Dorronsoro, editor, *ICANN'02*, pages 51–56. Springer, 2002.
[26] E. Castillo, O. Fontenla-Romero, B. Guijarro-Berdiñas, and A. Alo-Betanzos. A global optimum approach for one-layer neural networks. *Neural Computation*, 14:1429–1449, 2002.
[27] T. Chen, W. Lu, and S.-I. Amari. Global convergence rate of recurrently connected neural networks *Neural Computation*, 14:2947–2957, 2002.
[28] Y. Chen. Global stability of neural networks with distributed delays. *Neur. Netw.*, 15:867–871, 2002.
[29] J. Claussen and H. Schuster. Asymptotic level density of the elastic net self-organizing faeture map. In J. Dorronsoro, editor, *Proc. International Conf. on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science 2415, pages 939–944. Springer Verlag, 2002.
[30] J. C. Claussen. Generalized winner relaxing Kohonen feature maps. *e-print cond-mat*, 2002. (http://arXiv.org/cond-mat/0208414)
[31] M. Cottrell, J. C. Fort, and G. Pages. Theoretical aspects of the som algorithm. *Neurocomputing*, 21(1):119–138, 1998.
[32] K. Crammer, R. Gilad-Bachrach, A.Navot, and A.Tishby. Margin analysis of the lvq algorithm. In *Proc. NIPS 2002*, 2002.
[33] C. Dang and L. Xu. A Lagrange multiplier and Hopfield-type barrier function method for the traveling salesman problem. *Neural Computation*, 14:303–324, 2002.
[34] B. DasGupta and B. Hammer. On approximate learning by multi-layered feedforward circuits. Theoretical Computer Science, to appear.
[35] P. Demartines and J. Hérault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Trans. on Neural Networks*, 8(1):148–154, January 1997.
[36] R. Der and M. Herrmann. Critical phenomena in self-organizing feature maps: Ginzburg-Landau approach. *Physical Review E* , 49(6):pt. B, June 1994.
[37] R. Der and M. Herrmann. Reordering transitions in Self-Organized Feature Maps with short-range neighbourhood. In M. Marinaro and P. G. Morasso, editors, *Proc. ICANN'94, International Conference on Artificial Neural Networks*, volume I, pages 322–325, London, UK, 1994. Springer.
[38] R. Der, M. Herrmann, and T. Villmann. Time behaviour of topological ordering in self-organized feature mapping. *Biological Cybernetics*, 77(6):419–427, 1997.
[39] D. Dersch and P. Tavan. Asymptotic level density in topological feature maps. *IEEE Trans. on Neural Networks*, 6(1):230–236, January 1995.
[40] D. DeSieno. Adding a conscience to competitive learning. In *Proc. ICNN'88, International Conference on Neural Networks*, pages 117–124, Piscataway, NJ, 1988. IEEE Service Center.
[41] A. DiBlas, A. Jagota, and R. Hughuy. Energy function-based approaches to graph coloring. *IEEE Transactions on Neural Networks*, 13:81–91, 2002.
[42] S. Draghici. On the capability of neural networks using limited precision weights. *Neural Networks*, 15:395–414, 2002.
[43] W. Duch and N. Jankowski. Survey of neural transfer functions. *Neural Computing Surveys*, 2:163–212, 1999.
[44] T. Elsken. Even on finite test sets smaller nets may perform better. *Neur. Netw.*, 10:369–385, 1997.
[45] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: Ordering, convergence properties and energy functions. *Biol. Cyb.*, 67(1):47–55, 1992.
[46] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: Stationary states, metastability and convergence rate. *Biol. Cyb.*, 67(1):35–45, 1992.
[47] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.
[48] J. A. Flanagan. Sufficient conditions for self-organization in the one-dimensional SOM with a reduced width neighbourhood. *Neurocomputing*, 21(1–3):51–60, 1998.
[49] J. A. Flanagan. Self-organization in the SOM and Lebesque continuity of the input distribution. In *Proceedings of the International Joint Conference on Neural Networks*, volume 6, pages 26–31, Piscataway, NJ, 2000. Helsinki Univ of Technology, IEEE.
[50] J. A. Flanagan and M. Hasler. Self-organization, metastable states and the ODE method in the Kohonen neural network. In M. Verleysen, editor, *Proc. ESANN'95, European Symp. on Artificial Neural Networks*, pages 1–8, Brussels, Belgium, 1995. D facto conference services.
[51] K. Funahashi and Y. Nakamura. Approximation of dynamical systems by continuous time recurrent

neural networks. *Neural Networks*, 6(6):801–806, 1993.
[52] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.
[53] M. Gibbs. *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, Cambridge University, 1997.
[54] T. Graepel, M. Burger, and K. Obermayer. Self-organizing maps: generalizations and new optimization techniques. *Neurocomputing*, 21(1–3):173–90, 1998.
[55] B. Hammer. *Learning with recurrent neural networks*. Springer Lecture Notes in Control and Information Sciences. Springer, 2000.
[56] B. Hammer. Recurrent networks for structured data - a unifying approach and its properties. *Cognitive Systems Research*, 3:145–165, 2002.
[57] B. Hammer and K. Gersmann. A note on the universal approximation capability of support vector machines. *Neural Processing Letters*, to appear.
[58] B. Hammer, A. Micheli, and A. Sperduti. A general framework for unsupervised processing of structured data. In M. Verleysen, editor, *Proc. Of European Symposium on Artificial Neural Networks (ESANN'2002)*, pages 389–394, Brussels, Belgium, 2002. d-side.
[59] B. Hammer and J. Steil. Perspectives on learning with recurrent neural networks. In M. Verleysen, editor, *ESANN'02*, pages 357–368. D-side pucblications, 2002.
[60] B. Hammer, M. Strickert, and T. Villmann. Learning vector quantization for multimodal data. In J. Dorronsoro, editor, *Proc. International Conf. on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science 2415, pages 370–376. Springer Verlag, 2002.
[61] B. Hammer and P. Tiňo. Neural networks with small weights implement definite memory machines. *Neural Computation*, to appear.
[62] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.
[63] T. Hastie and W. Stuetzle. Principal curves. *J. Am. Stat. Assn.*, 84:502–516, 1989.
[64] R. Herbrich, T. Graepel, and C. Campbell. Bayes point machines. *Journal of Machine Learning Research*, 1:245–279, 2001.
[65] R. Herbrich and R. Williamson. Algorithmic luckiness. *Journal of Machine Learning Research*, 3:175–212, 2002.
[66] J. A. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*, volume 1 of *Santa Fe Institute Studies in the Sciences of Complexity: Lecture Notes*. Addison-Wesley, Redwood City, CA, 1991.
[67] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–316. Elsevier, Amsterdam, 1999.
[68] G. Hinton and T. Sejnowski. *Unsupervised Learning*. MIT Press, 1998.
[69] K.-U. Höffgen, H.-U. Simon, and K. VanHorn. Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50:114–125, 1995.
[70] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, pages 359–366, 1989.
[71] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. J. Wiley Sons, 2001.
[72] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7:95–114, 2000.
[73] L. Jones. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural networks. *Annals of Statistics*, 20:608–613, 1992.
[74] L. Jones. The computational intractability of training sigmoidal neural networks. *IEEE Transactions on Information Theory*, 43:161–173, 1997.
[75] J. Judd. *Neural network design and the complexity of learning*. MIT-Press, 1990.
[76] S. Kaski and J. Sinkkonen. A topography-preserving latent variable model with learning metrics. In N. Allison, H. Yin, L. Allison, and J. Slack, editors, *Advances in Self-Organising Maps*, pages 224–9. Springer, 2001.
[77] M. Kimura. On unique representations of certain dynamical systems produced by continuous-time recurrent neural networks. *Neural Computation*, 14:2981–2996, 2002.
[78] T. Kohonen. Automatic formation of topological maps of patterns in a self-organizing system. In E. Oja and O. Simula, editors, *Proc. 2SCIA, Scand. Conf. on Image Analysis*, pages 214–220, Helsinki, Finland, 1981. Suomen Hahmontunnistustutkimuksen Seura r. y.
[79] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).
[80] P. Koiran and E. D. Sontag. Vapnik-Chervonenkis dimension of recurrent neural networks. In *Proc. of the 3rd Eur. Conf. on Comp. Learning Theory*, pages 223–237, 1997.
[81] M. Köppen. On the training of Kolmogorov networks. In J. Dorronsoro, editor, *ICANN'02*, pages 474–479. Springer, 2002.
[82] C.-M. Kuan and K. Hornik. Convergence of learning algorithms with constant learning rates. *IEEE Transactions on Neural Networks*, 2:484–489, 1991.
[83] V. Kurkova, P. Savicky, and K. Hlavackova. Representations and rates of approximations of real-

valued Boolean functions by neural networks. *Neural Networks*, 11:651–659, 1998.
[84] H. Kushner and D. Clark. *Stochastic Appproximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, 1978.
[85] D. LaLoudouanna and M. Tarare. Data set selection. Most original submission of NIPS'2002.
[86] E. Lavretsky. On the geometric convergence of neural approximations. *IEEE Transactions on Neural Networks*, 13:274–282, 2002.
[87] D.-L. Lee. Pattern sequence recognition using a time-varying Hopfield network. *IEEE Transactions on Neural Networks*, 13:330–342, 2002.
[88] W.-J. Li and T. Lee. Hopfield neural networks for affine invariant matching. *IEEE Transactions on Neural Networks*, 12:1400–1410, 2001.
[89] X. Liao, G. Chen, and E. Sanchez. Delay-dependent exponential stability analysis of neural networks: an LMI approach. *Neural Networks*, 15:855–866, 2002.
[90] C.-J. Lin. On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*, 12:1288–1298, 2001.
[91] S. Lin and J. Si. Weight convergence and weight density of the multi- dimensional SOFM algorithm. In *Proceedings of the 1997 American Control Conference*, volume 4, pages 2404–8. American Autom. Control Council, Evanston, IL, USA, 1997.
[92] S. Lin and J. Si. Weight-value convergence of the SOM algorithm for discrete input. *Neural Computation*, 10(4):807–14, 1998.
[93] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.
[94] R. Linsker. How to generate maps by maximizing the mutual information between input and output signals. *Neural Computation*, 1:402–411, 1989.
[95] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, 1983.
[96] W. Maass and C. Bishop, editors. *Pulsed Neural Networks*. MIT-Press, 1998.
[97] T. Martinetz and K. Schulten. Topology representing networks. *Neur. Netw.*, 7(3):507–522, 1994.
[98] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.
[99] N. Megiddo. On the complexity of polyhedral separability. *Discrete and Computational Geometry*, 3:325–337, 1988.
[100] E. Oja and J. Lampinen. Unsupervised learning for feature extraction. In J. M. Zurada, R. J. M. II, and C. J. Robinson, eds., *Computational Intelligence Imitating Life*, pages 13–22. IEEE Press, 1994.
[101] M. Opper. Statistical mechanics of generalization. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 1087–1090. MIT-Press, 2nd edition, 2003.
[102] J. Park and I. Sandberg. Approximation and radial-basis-function networks. *Neural Computation*, 5:305–316, 1993.
[103] J. Peng, H. Qiao, and Z.-b. Xu. A new approach to stability of neural networks with time-varying delays. *Neural Networks*, 15:95–103, 2002.
[104] C. Pinter. Complexity of network training for classes of neural networks. In K. P. Jantker, T. Shinohara, and T. Zeugmann, editors, *ALT'95*, pages 215–227. Springer, 1995.
[105] J. C. Principe, J. F. III, and D. Xu. Information theoretic learning. In S. Haykin, editor, *Unsupervised Adaptive Filtering*. Wiley, New York, NY, 2000.
[106] B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
[107] H. Ritter. Asymptotic level density for a class of vector quantization processes. *IEEE Trans. on Neural Networks*, 2(1):173–175, January 1991.
[108] H. Ritter. Self-organizing maps on non-euclidean spaces. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 97–110. Elsevier, Amsterdam, 1999.
[109] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wesley, Reading, MA, 1992.
[110] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22:400–407, 1951.
[111] F. Rossi, B. Conan-Guez, and F. Fleuret. Theoretical properties of functional multi layer perceptrons. In M. Verleysen, editor, *ESANN'02*, pages 7–12. d-side publications, 2002.
[112] J. Rubner and P. Tavan. A self–organizing network for principle–component analysis. *Europhys. Letters*, 7(10):693–698, 1989.
[113] A. A. Sadeghi. Asymptotic behaviour of self-organizing maps with non-uniform stimuli distribution. Technical Report 166, Universität Kaiserslautern, FB Mathematik, Germany, July 1996.
[114] A. A. Sadeghi. Convergence in distribution of the multi-dimensional Kohonen algorithm. *Journ. of Appl. Prob.*, 38(1):136–151, MAR 2001.
[115] A. S. Sato and K. Yamada. Generalized learning vector quantization. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 423–429. MIT Press, 1995.
[116] F. Scarselli and A. Tsoi. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, 11:15–37, 1998.
[117] M. Schmitt. Descartes' rule of signs for radial basis function neural networks. *Neural Computation*,

14:2997–3011, 2002.
[118] M. Schmitt. On the complexity of computing and learning with multiplicative neural networks. *Neural Computation*, 14:241–301, 2002.
[119] H. T. Siegelmann and E. D. Sontag. Analog computation, neural networks, and circuits. *Theoretical Comp. Science*, 131:331–360, 1994.
[120] J. Sima. Back-propagation is not efficient. *Neural Networks*, pages 1017–1023, 1996.
[121] J. Sima. Training a single sigmoidal neuron is hard. *Neural Computation*, 14:2709–2728, 2002.
[122] M. Soerens, P. Latinne, and C. Decaestecker. Any reasonable cost function can be used for a posteriori probability approximation. *IEEE Transactions on Neural Networks*, 13:1204–1210, 2002.
[123] P. Sollich and A. Halees. Learning curves for Gaussian process regression: Approximations and bounds. *Neural Computation*, 14:1393–1428, 2002.
[124] E. Sontag. Feedforward nets for interpolation and classification. *Journal of Computer and System Sciences*, 45:20–48, 1992.
[125] E. Sontag. VC dimension of neural networks. In C. Bishop, editor, *Neural Networks and Machine Learning*, pages 69–95. Springer, 1998.
[126] J. Steil. Local structural stability of recurrent networks with time-varying weights. *Neurocomputing*, 48:39–51, 2002.
[127] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2002.
[128] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT-Press, 1998.
[129] P. Talavan and J. Yañez. Parameter setting of the Hopfield network applied to TSP. *Neural Networks*, 15:363–373, 2002.
[130] R. Tetzlaff, editor. *Cellular Neural Networks and their Applications*. World Scientific, 2002.
[131] P. Tiňo, B. G. Horne, and C. L. Giles. Attractive periodic sets in discrete-time recurrent networks (with emphasis on fixed-point stability and bifurcations in two-neuron networks). *Neural Computation*, 13:1379–1414, 2001.
[132] K. Torkkola and W. Campbell. Mutual information in learning feature transformations. In *Proc. Of International Conference on Machine Learning ICML'2000*, Stanford, CA, 2000.
[133] P. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.
[134] M. M. van Hulle. *Faithful Representations and Topographic Maps From Distortion- to Information-based Self-organization*. J. Wiley & Sons, Inc., 2000.
[135] M. M. Van Hulle and D. Martinez. On an unsupervised learning rule for scalar quantization following the maximum entropy principle. *Neural Computation*, 5(6):939–953, 1993.
[136] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.
[137] T. Voegtlin. Recursive self-organizing maps. *Neural Networks*, 15(8-9):979–991, 2002.
[138] M. Vidyasagar. *A Theory of Learning and Generalization*. Springer, 1997.
[139] T. Villmann. Topology preservation in self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 279–292, Amsterdam (Holland), June 1999. Helsinki, Elsevier.
[140] T. Villmann. Controlling strategies for the magnification factor in the neural gas network. *Neural Network World*, 10(4):739–750, 2000.
[141] T. Villmann, R. Der, M. Herrmann, and T. Martinetz. Topology Preservation in Self–Organizing Feature Maps: Exact Definition and Measurement. *IEEE Trans. on Neural Networks*, 8(2):256–266, 1997.
[142] G. Wahba. Generalization and regularization in nonlinear learning systems. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 426–430, Cambridge, MA, 1995. The MIT Press.
[143] X. Wang. Period-doublings to chaos in a simple neural network: An analytic proof. *Complex Systems*, 5:425–442, 1991.
[144] M. Welling and G. Hinton. A new learning algorithm for mean field Boltzmann machines. In J. Dorronsoro, editor, *ICANN'02*, pages 351–256. Springer, 2002.
[145] S. Weng and J. Steil. Data driven generation of interactions for feature binding. In J. Dorronsoro, editor, *ICANN'02*, pages 432–437. Springer, 2002.
[146] H. Wersing, W.-J. Beyn, and R. H. Dynamical stability conditions for recurrent neural networks with unsaturating piecewise linear transfer function. *Neural Computation*, 13:1811–1825, 2001.
[147] W. Wiegerinck and T. Heskes. On-line learning with time-correlated patterns. *Europhysics Letters*, 28(6):451–5, Nov 1994.
[148] D. J. Willshaw and C. V. der Malsburg. How patterned neural connections can be set up by self–organization. *Proceedings of the Royal Society of London, Series B*, 194:431–445, 1976.
[149] P. L. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Transaction on Information Theory*, (28):149–159, 1982.
[150] T. Zhang. Effective dimension and generalization of kernel learning. In *NIPS'02*.