Scaling Properties of Neural Networks for the Prediction of Time Series

Axel Röbel
Technical University Berlin, Einsteinufer 17, Sekr. EN-8,
10587 Berlin, Germany
roebel@kgw.tu-berlin.de

Abstract: Scaling properties of neural networks, that are the relations between the number of hidden units and the training or generalization error, recently have been investigated theoretically with encouraging results. In our paper we investigate experimentally, whether the theoretic results may be expected in practical applications. We investigate different neural network structures with varying number of hidden units for solving two time series prediction tasks. The results show a considerable difference of the scaling behavior of multi layer perceptrons and radial basis function networks.

Introduction

Time series prediction is one of the applications for which recent methods of nonlinear signal processing have led to new and interesting possibilities. Especially the use of neural networks for prediction [20, 19] and even modeling of nonlinear dynamical systems [11, 13] has been successfully investigated. The basic idea of this methods has been developed by a number of groups independently around 1980, for example [2], and has been given a strong theoretical background later by [17, 14].

One open question concerning the behavior of neural networks is the relation between the number of hidden units and the network error, in the following denoted as scaling properties. There exists some recent encouraging theoretical results concerning this scaling properties of neural networks [1, 6], however, it is unknown whether these results may be obtained by existing training algorithms. Therefore, we have investigated the scaling of the neural network prediction error solving two time series prediction tasks and report our results in the following sections.

The following article is organized as follows. In the first two sections we give a short introduction into the foundation of time series prediction by phase space reconstruction with delayed coordinate vectors and summarize on the theoretical results concerned with the scaling properties of neural networks. In section three and four we describe the network structures and training algorithms we have used in the experiments. In section five we report the results for predicting a computer generated time series from the Mackey Glass differential equation and a real world saxophone tone, respectively and we study the relation between the number of hidden units of the networks and the prediction error on a hold out test set.

1 Reconstructing Attractors

Assume an n-dimensional dynamical system $f(\cdot)$ evolving on an attractor A. A has fractal dimension d, which often is considerably smaller then n. The system state $\vec{z}$ is observed through a sequence of measurements $h(\vec{z})$, resulting in a time series of measurements $y_t = h(\vec{z}(t))$. Under weak assumptions concerning $h(\cdot)$ and $f(\cdot)$ the fractal embedding theorem[14] ensures that, for $D > 2d$, the set of all *delayed coordinate vectors*

$$Y_{D,T} = \{t > t_0 : (y_t, y_{t-T}, \ldots, y_{t-(D-1)T})\}, \tag{1}$$

with an arbitrary delay time T, forms an embedding of A in the D-dimensional *reconstruction* space. Because an embedding preserves characteristic features of A, especially it is one to one, it may be employed for prediction or modeling of the system dynamics. The delayed coordinate vectors are easily obtained from the time series by a tapped delay line. Provided the dimension of this tapped delay line is chosen large enough, the embedding theorem ensures that there exists a nonlinear autoregressive (NAR) model that predicts the future behavior of the time series.

2 Scaling Properties of Basic Neural Network Structures

The approximation capabilities of the different neural network structures have been investigated by various authors [5, 10, 3]. These investigations agree in the point that the multi layer perceptron as well as the radial basis function network with a sufficient number of hidden units are capable to approximate smooth functions to any degree of accuracy. Therefore, neural networks are well suited to build the NAR time series model introduced in the previous section.

Up to now, the number of hidden units that has to be used for a desired approximation error is still unknown. Some recent results address the relation between the training error of a neural network and the number of hidden units [1, 6]. These results are encouraging in that for restricted classes of target functions it is shown that there exists sequences of networks with increasing number of hidden units such the minimal training error depends on the number of hidden units following

$$mean\ squared\ error \leq \frac{C}{number\ of\ hidden\ units}\,, \tag{2}$$

with a problem dependent constant C and does not depend on the dimension of the domain of the target function. In a practical situation the existing training algorithms will generally fail to achieve the optimal solution of the problem and, therefore, the question arises, to what extend the theoretical scaling results apply for practical applications. In the following sections we report our results concerned with an experimental investigation of the scaling behavior of neural network for time series prediction. Here we restrict ourselves to the investigation of the training process in a somewhat ideal situation with a sufficient number of training samples (no overfitting) and small noise level. Any severe limitation of the training set size or a considerable level of noise in the data would certainly affect the theoretical bounds for the scaling behavior of the neural networks and, therefore, is beyond the scope of this investigation.

3 Network Topologies

Many different network structures have been proposed in the literature for predicting time series. For the following investigation, however, we restrict ourselves to the basic topologies, the multi layer perceptron and radial basis function networks, which are frequently extended to build more sophisticated approaches, for example the FIR-network [19, pp. 195] or Fuzzy-Neuro-Systems [7].

Multi layer perceptron: The application of multi layer perceptrons (MLP) to time series prediction has first been investigated by Lapedes and Farber [8] and still is frequently used today [20]. Lapedes and Farber used a network with sigmoid activation function $a_s(x)$ in two hidden layers and a linear output unit. In our experiments we found that the scaling behavior of the MLP does not depend on the number of hidden layers. Therefore, we restrict ourselves to use a MLP with a single hidden layer in the following experiments. The network function of the MLP is

$$n(\vec{x}) = \sum_i W_i a_s(\vec{w_i} * \vec{x} + b_i), \quad a_s(x) = \frac{1}{1+e^{-x}} \tag{3}$$

where W_i is the hidden to output weight for hidden unit i, $\vec{w_i}$ and b_i are input weights and bias of hidden unit i and $\vec{x}$ is the input vector presented to the network.

Radial basis function networks: Radial basis function (RBF) networks have been proposed for time series prediction by Moody and Darken [9]. In this case the activation function is Gaussian shaped

$$a_r(\vec{x}, \vec{w}, \sigma) = e^{-(\frac{\vec{x}-\vec{w}}{\sigma})^2}, \tag{4}$$

with $\vec{w}$ and σ describing the center and width of the Gaussian. The overall network function may be structured equivalently to (3), by replacing a_s with a_r. Alternatively an additional normalization is incorporated leading to the network function

$$n(\vec{x}) = \sum_i W_i \frac{a_r(\vec{x}, \vec{w_i}, \sigma_i)}{\sum_i a_r(\vec{x}, \vec{w_i}, \sigma_i)}. \tag{5}$$

In the following we will reference the former type by RBF-G and the latter by RBF-N. For both cases Moody and Darken proposed an hybrid training algorithm, composed of a vector quantization (VQ) algorithm for determining the centers $\vec{w_i}$ and the widths σ_i and a linear optimization of the output layer. This algorithm is very fast and, provided the σ_i are selected properly, results in a network with localized response of the hidden units. However, due to the independent optimization of the different layers this algorithm generally does not obtain optimal performance. An improvement may be obtained by employing a global nonlinear optimization to adapt all parameters after the vector quantization scheme [18]. Note, however, that the global optimization may destroy the local response of the hidden units, by considerably increasing the respective σ_i [9].
In our comparison we used both, the RBF-G and the RBF-N, networks with VQ and linear or VQ and nonlinear optimization. The type of optimization employed in the second step of the hybrid training scheme is indicated by an lower letter following the network type as in RBF-Nl for normalized Gaussian units with VQ and linear optimization of the output layer or RBF-Gn for Gaussian units with VQ and nonlinear optimization of all parameters.

It is widely accepted that due to the local response of a_r the RBF-networks require more hidden units to achieve the same performance as an MLP. To improve the RBF-Nl networks for a small number of hidden units an extended RBF network has been proposed [16]. In this approach the weights W_i are replaced by local linear models yielding a network function

$$n(\vec{x}) = \sum_i (\vec{W_i}\vec{x} + B_i) \frac{a_r(\vec{x}, \vec{w_i}, \sigma_i)}{\sum_i a_r(\vec{x}, \vec{w_i}, \sigma_i)}, \tag{6}$$

which is equivalent to a recently established Neuro-Fuzzy Modell (ANFIS) [7]. For the following experiments we employed this structure with initialization by VQ and global nonlinear optimization of all parameters (RBF-Ln) to investigate whether this structure is able to achieve an improvement compared to the more traditional RBF networks.

4 Training Algorithms

Most of the chosen neural network structures need to be initialized by a VQ scheme. For our experiments we started that scheme by randomly choosing n input vectors from the training data to initialize the n hidden unit weights $\vec{w_i}$. Then we apply all the p training data input vectors and try to minimize the vector quantization error

$$E_{vq}(\vec{w_1}, \ldots, \vec{w_n}) = \sum_i^n \sum_j^p \delta_{ij} |\vec{x_j} - \vec{w_i}|^2 \tag{7}$$

proposed in [9]. In this equation δ_{ij} is the cluster membership function being 1 if pattern $\vec{x_j}$ is nearest to $\vec{w_i}$ and 0 otherwise. E_{vq} is locally minimized by an VQ iterative algorithm that randomly presents the training patterns and adjusts the nearest hidden unit position by

$$\vec{w_i}(t+1) = \vec{w_i}(t) + \epsilon(t)(\vec{x_j} - \vec{w_i}(t)). \quad (8)$$

The learning rate $\epsilon(t)$ is initialized to 0.5 and decreased for each training pass until the error E_{vq} converged. According to [18] the width factors σ_i of the RBF units are initialized to zero and are also adapted during the vector quantization scheme following

$$\sigma_i(t+1) = (1 - \epsilon(t)\sigma_i(t) + 2\epsilon(t)|\vec{x_j} - \vec{w_i}(t)|. \quad (9)$$

For the nonlinear optimization part of the training we have chosen to apply a state of the art training algorithm called *RPROP*, which has been demonstrated to outperform many known algorithms in various test settings [15, 12]. This algorithm is easy to implement and does not need any critical parameter settings to be selected. Moreover, it incorporates local learning rates, a feature which is expected to be useful especially for optimizing the various RBF networks. In our setting this algorithm has been applied to the MLP, RBF-Nn, RBF-Gn and RBF-Ln network nonlinear optimization as well as for the adaptation of the output weights W_i of the RBF-Nl networks.

5 Experimental Results

The first time series we used for our experiments consists of computer generated data from the well known Mackey Glass equation

$$\dot{x}(t) = \frac{a \cdot x(t - T_d)}{1 + x(t - T_d)^{10}} - b \cdot x(t) \;, \quad (10)$$

which exhibits chaotic behavior for the parameters $a = 0.2$, $b = 0.1$ and $T_d = 30$ [4]. For integration of the system we apply a second-order Runge-Kutta method with an integration time step of $dt = 0.05$. The resulting time series has been subsampled to obtain a sample period of 1. After skipping 1000 samples to allow the transients to decay we recorded 10000 samples for the training data set and the following 5000 samples as hold out test set. The Mackey Glass attractor is known to have fractal dimension $d \approx 3$.[1] According to the embedding theorem we choose the neural network input dimension to be $2d + 1 = 7$.
As a second example we have chosen a real world time series, a saxophone tone consisting of 16000 samples sampled at 32kHz. The neural modeling of this sound has been described in [13], where we have shown that the sound may be modeled by a neural network with 9 dimensional inputs from the time series and an additional input which allows to consider the nonstationarity of the signal. Due to this nonstationarity the training set and test set both have to span the entire time series and are selected to contain the odd and even samples respectively. Both signals have been

[1]Note that the Mackey Glass equation describes a time delay system and, therefore, has infinite dimensional phase space.

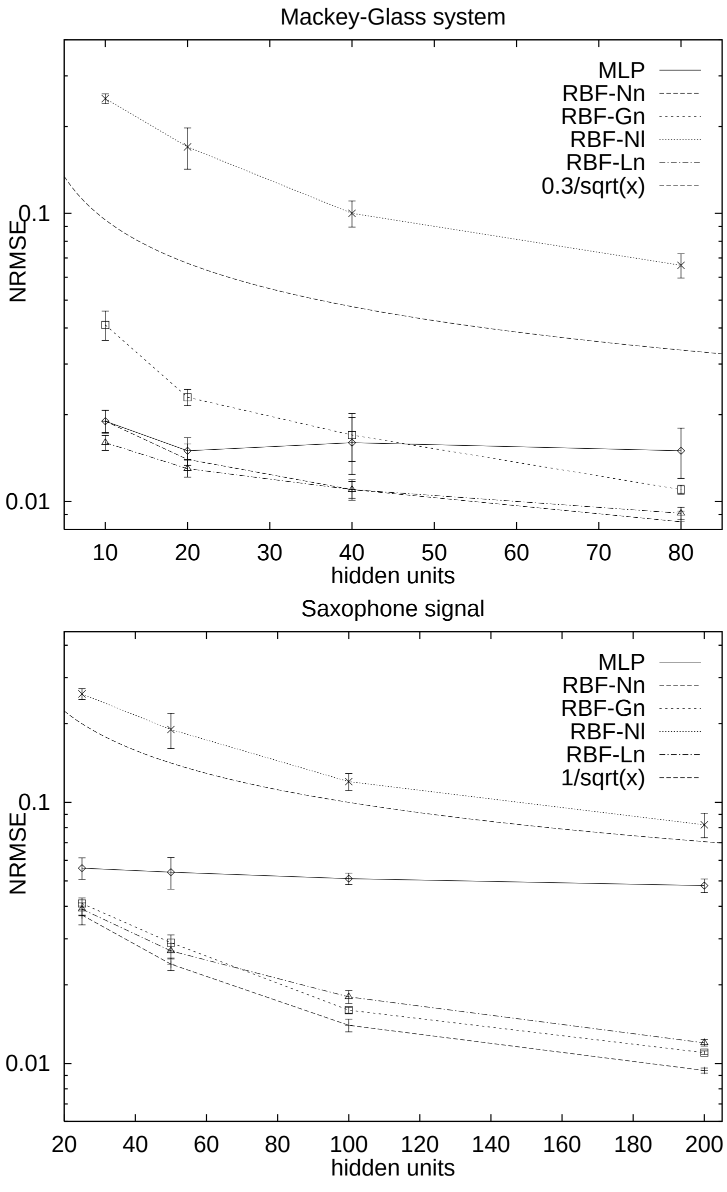

Fig. 1: NRMSE of various neural network structures for predicting a computer generated Mackey Glass time series and a saxophone sound signal.

normalized between $[-1:1]$ and in both cases the delay time T of the tapped delay line has been chosen to be 5.

Obviously, due to the lack of an optimal training algorithm the following experiments does not hold for a strong validation of the theoretical results mentioned in section **2**. However, the experiments are intended to show, how close the various network structures will achieve the theoretical limits in a practical setting. One of the main practical limitations is given by finite computational power. Therefore, it is natural to supply a fixed iteration count for the nonlinear optimization (RPROP) of the different networks. In the following experiments we have performed 5000 batch-cycles for the nonlinear optimization of the parameters. At the end of training the decrease of the training error for one batch-cycle normalized by the training error has been analyzed to be of the same order for all networks. Therefore, we expect that a moderate increase of training cycles will not qualitatively change the results. In case of the linear RBF-Xl networks the linear optimization has been verified to reach the global minimum.
We apply the 6 different network structures with increasing number of hidden units to both of the selected time series prediction tasks. For each task and each network we used 5 different initializations for optimization. After optimization we calculate the average normalized prediction error (NRMSE)[2] and its variance in using the hold out test set. As mentioned earlier, by using a sufficient amount of training data we prevent any overfitting, which would have been indicated by a difference between training and test error.

The prediction error obtained by the various networks on the hold out test sets of both time series are shown in figure 1. In both cases we observe, that the MLP network does not show the expected scaling behavior at all. While in case of the saxophone time series the average prediction error of the MLP and its variance remains nearly constant. For the Mackey Glass time series the results are even more in contrast to the expectations, with an increasing average training error beyond a network complexity of 20 hidden units. In this case the increasing prediction error is accompanied by an considerable increase in the variance of the results. This leads to the explanation that the complexity of the optimization problem and the number of local minima prevents any reasonable improvement of the solution. Note, that the behavior is not confined to the special training algorithm used, but, has been observed with more sophisticated algorithms, too.
As a second result we find that the scaling of all the RBF-related networks is comparable and, moreover, it is comparable to the expected relation given in (2). The error for both types of RBF-Xl networks is similar and we present only the RBF-Nl networks, which obtained in both applications a slightly lower prediction error. In the logarithmic scaling of figure 1 the pairwise difference of the prediction error for the various RBF networks is roughly independent of the network complexity, which is related to the expectation that the constant C in (2) will differ for the different type of networks. The comparison of the RBF-Xn type networks shows, that

[2]For normalization we use the variance of the time series.

all this networks obtain similar results. For both tasks, however, the lowest prediction error has been obtained with the RBF-Nn networks. Compared to the RBF-Nl network the nonlinear optimization yields an improvement in the NRMSE of about an order of magnitude.
As a last result we compare the prediction error achieved by the RBF-Xx networks and the MLP. As expected the RBF-Xl network, due to the heuristic optimization of the parameters of the first layer, given by (8) and (9), and due to the strictly local activation function, needs lots of hidden units more than the MLP to obtain comparable prediction quality. However, the nonlinear global optimization of the RBF-Xn networks has outperformed the MLP in almost all cases[3]. Moreover, the RBF-Nn networks converged considerably faster and, therefore, may be expected to need less computation expense to be optimized.

At first glance it might be surprising that the optimization of RBF-Xn networks does not suffer from the increase in complexity and local minima as the MLP. As an explanation one might argue that the VQ initialization of the RBF networks achieves a fairly good initialization of the network. The remaining optimization does not start from scratch and, therefore, does not end in a local minima as often as in the case of the MLP. A further advantage of the optimization of the RBF-Xx networks is the local behavior of the hidden units. If we accept the number of interacting units for a specific location in input space as an indicator for the optimization complexity, we may argue, that due to the local response of the RBF hidden units the number of overlapping hidden units remains nearly constant even for large networks. In contrast to this in case of the MLP nearly all hidden units contribute to any location of the target function and, therefore, the number of interactions and the complexity of the optimization raises with the power of two of the number of hidden units.

6 Conclusion

The results presented in this paper indicate that the time series prediction error of RBF neural networks obtained by a practical algorithm does show the relation to the number of hidden units that is expected from recent theoretical results. By predicting a computer generated and a real world time series we have demonstrated that the global nonlinear optimization of RBF networks improves the prediction error by an order of magnitude compared to the RBF networks that are optimized only in the output layer. Compared to MLP networks with the same number of hidden units, we obtained the result that RBF networks with nonlinear optimization give rise to lower generalization error, even in the case of small networks. For the MLP networks we do not find the expected scaling behavior and conjecture that the random initialization and the global response of the hidden units are the reasons for the optimization not being able to adequately solve the problem for an increasing number of hidden units.

[3] A similar result has been described for the logistic function by Moody and Darken [9].

References

[1] Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

[2] J.P. Crutchfield, J.D. Farmer, N.H. Packard, and R.S. Shaw. Geometry from a time series. *Phys. Rev. Lett.*, 45(9):712–716, 1980.

[3] G. Cybenko. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, (2):303–314, 1989.

[4] J.Doyne Farmer. Chaotic attractors of an infinite-dimensional dynamical system. *Physica D*, 4:366–393, 1982.

[5] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[6] Kurt Hornik, Maxwell Stinchcombe, Halbert White, and Peter Auer. Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. NeuroCOLT Series NC-TR-95-004, ESPRIT Working Group in Neural and Computational Learning, ftp://ftp.dcs.rhbnc.ac.uk/pub/neurocolt/tech_reports/nc-tr-95-004.ps.Z, 1995.

[7] J.-S. Roger Jang. ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. *IEEE Trans. on Systems, Man & Cybernetics*, 23(3):665–685, 1993.

[8] A. Lapedes and R. Farber. How neural nets work. *IEEE Conference on Neural Information Systems*, 1:442–457, 1987.

[9] J. Moody and C.J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[10] J. Park and I.W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, 1991.

[11] Jose C. Principe and Jyh-Ming Kuo. Dynamic modelling of chaotic time series with neural networks. In G. Tesauro, D. S. Touretzky, and T.K. Leen, editors, *Neural Information Processing Systems 7 (NIPS 94)*, 1995.

[12] Martin Riedmiller. Advanced supervised learning in multilayer perceptrons – From backpropagation to adaptive learning algorithms. *Computer Standards and Interfaces, Special Issue on Neural Networks*, 5, 1994.

[13] Axel Röbel. Neural networks for modeling time series of musical instruments. In *Proc. of the International Computer Music Conference, ICMC*, pages 424–428, Banff, Canada, 1995.

[14] Tim Sauer, James A. Yorke, and Martin Casdagli. Embedology. *Journal of Statistical Physics*, 65(3/4):579–616, 1991.

[15] W. Schiffmann, M.Joost, and R. Werner. Optimization of the backpropagation algorithm for training multilayer perceptrons. Technical Report , University of Koblenz, Institute of Physics, Rheinau 3-4, 5400 Koblenz, Germany, 1993. FTP Server archive.cis.ohio-state.edu (128.146.8.52) in /pub/neuroprose/schiff.bp_speedup.ps.Z.

[16] K. Stokbro, D.K. Umberger, and J.A. Hertz. Exploiting neurons with localized receptive fields to learn chaos. *Complex Systems*, 4:603–622, 1990.

[17] F. Takens. *Detecting Strange Attractors in Turbulence*, volume 898 of *Lecture Notes in Mathematics (Dynamical Systems and Turbulance, Warwick 1980)*, pages 366–381. D.A. Rand and L.S. Young, Eds. Berlin: Springer, 1981.

[18] Michel Verleysen and Katerina Hlavackova. An optimized RBF network for approximation of functions. In *Proceedings of the European Symposium on Artificial Neural Networks, ESANN'94*, 1994.

[19] Andreas S. Weigend and Neil A. Gershenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley Pub. Comp., 1993.

[20] A.S. Weigend, B.A. Huberman, and D.E. Rumelhart. Predicting the future: a connectionist approach. *Intern. Jou. of Neural Systems*, 1(3):193–209, 1990.