Dynamic Pattern Selection for Faster Learning and Controlled Generalization of Neural Networks

Axel Röbel
Technical University of Berlin, Sekr. INI FR 5-9,
Franklinstr 28/29
10587 Berlin, Germany

Abstract

We address the question of selecting a proper training set for neural network time series prediction or function approximation. As a result of analyzing the relation between approximation and generalization, a new measure, the generalization factor, is introduced. Using this factor and cross validation we develop the *dynamic pattern selection* algorithm. By employing two time series prediction tasks, we compare the results for dynamic pattern selection training to results obtained with fixed training sets. The favorable properties of the dynamic pattern selection, namely lower computational expense and control of generalization, are demonstrated.

1. Introduction

It is well known that the generalization properties of neural networks used in function approximation are strongly affected by the size and distribution of the training set. However, the problem of selecting the optimal training set has not yet been solved. For good generalization the training set has to contain enough information to fix the network function f_n not only at the training patterns, but on the domain $\mathbb{X}$ of the target function f_t.

To achieve this we want to adapt the training set during training and employ the net function to decide which pattern should be chosen. Plutowski and White [6] have done some work on *active pattern selection*, but did not employ cross validation to assess the generalization properties obtained by the training set. In contrast to their algorithm, the *dynamic pattern selection* proposed here, achieves concise training sets by continually validating the generalization properties of the net [7], [8].

2. Dynamic Pattern Selection

First some definitions concerning approximation, generalization and the training set $\mathbb{D}_s$ of a neural net are established. It is assumed that $\mathbb{D}_s$ contains only a finite number of elements out of $\mathbb{X}$. Given a real number $\epsilon \geq 0$, we distinguish between three subsets of $\mathbb{C}^{\infty}$, which is the set of functions with continuous derivatives of every order. First, there exists the set of ϵ-approximating functions, denoted $\mathbb{F}_a(\epsilon)$, which contains all functions f approximating the members in $\mathbb{D}_s$ to a given precision

$$\sup_{\vec{x} \in \mathbb{D}_s} \|f_t(\vec{x_i}) - f(\vec{x_i})\| \leq \epsilon. \tag{1}$$

Second, the set of ϵ-interpolating functions $\mathbb{F}_i(\epsilon)$, with distance

$$\sup_{\vec{x} \in \mathbb{X}} \|f_t(\vec{x}) - f(\vec{x})\| \leq \epsilon \tag{2}$$

to f_t. Third, the set of functions representable by the the neural net denoted as $\mathbb{F}_n$. While the sets $\mathbb{F}_a(\epsilon)$ depend on $\mathbb{D}_s$, the sets $\mathbb{F}_i(\epsilon)$ are completely defined by f_t. For every ϵ_0 the set $\mathbb{F}_i(\epsilon_0)$ is subset of $\mathbb{F}_a(\epsilon_0)$.

The goal is to select a training set $\mathbb{D}_s$ such that minimizing the net error $N(\mathbb{D}_s)$ forces f_n to converge to f_t all over $\mathbb{X}$. To achieve this we try to adapt the cutting $\mathbb{F}_a(\epsilon) \cap \mathbb{F}_n$ closely to $\mathbb{F}_i(\epsilon)$. For faithful generalization we demand that the generalization error is lower then the training error. Formally $f_n \in \mathbb{F}_a(\epsilon)$ should implicate $f_n \in \mathbb{F}_i(\epsilon)$. To be able to rank the generalization properties of f_n it is reasonable to define the generalization factor

$$\rho(f_n) = \frac{\epsilon_i(f_n)}{\epsilon_a(f_n)}, \tag{3}$$

where $\epsilon_a(f_n)$ is the minimal ϵ such that $f_n \in \mathbb{F}_a(\epsilon)$ and $\epsilon_i(f_n)$ is minimal such that $f_n \in \mathbb{F}_i(\epsilon)$. The generalization factor indicates the error we make in optimizing on $\mathbb{D}_s$ instead of $\mathbb{X}$. As a result we conclude, that

$$\rho(f_n) \leq 1.0 \tag{4}$$

is a reasonable condition for valid generalization.

The training set may now be controlled to achieve a faithful generalization. The straightforward strategy, namely to enlarge $\mathbb{D}_s$ by inserting the maximum error pattern whenever the generalization factor is beyond one, results in very small training sets [7]. For high precision training, however, the selection process turns out to be to slow. A more sophisticated approach, using a varying upper bound for the generalization factor, fixes the problems [8].

The generalization factor $\rho(f_n)$, namely the distance $\epsilon_i(f_n)$, has to be estimated by means of a validation set $\mathbb{D}_v$. Employing a cross validation strategy, the available data is split into a training/validation repertoire. The training set $\mathbb{D}_s$ is selected from the the training repertoire. The validation set $\mathbb{D}_v$ is randomly chosen out of the validation repertoire such that $|\mathbb{D}_s| = |\mathbb{D}_v|$[1]. Employing the net error function $N()$ the generalization factor is effectively estimated by

$$\rho_v = \frac{N(\mathbb{D}_v)}{N(\mathbb{D}_s)}. \tag{5}$$

[1] $|\mathbb{A}|$ denotes the cardinality of $\mathbb{A}$

training set size	training error $E(\mathbb{D}_s)$	generalization error $E(\mathbb{D}_u)$	generalization factor ρ_u	for/backward propagations
15	4.29e-3 ± 2.0e-3	2.50e-2 ± 1.7e-2	5.83	0.54e+6
25	3.77e-3 ± 1.8e-3	4.80e-3 ± 2.0e-3	1.27	0.90e+6
35	2.46e-3 ± 1.2e-3	2.90e-3 ± 1.4e-3	1.18	1.26e+6
50	3.02e-3 ± 1.6e-3	3.46e-3 ± 1.8e-3	1.15	1.80e+6
75	2.99e-3 ± 1.5e-3	3.14e-3 ± 1.7e-3	1.05	2.70e+6
100	2.90e-3 ± 1.6e-3	3.21e-3 ± 1.7e-3	1.10	3.60e+6
150	4.53e-3 ± 2.1e-3	5.08e-3 ± 2.4e-3	1.12	5.40e+6
69 ± 10	3.69e-3 ± 1.7e-3	3.06e-3 ± 1.1e-3	0.83	0.74e+6

Tab. 1. Predicting the Henon model. Comparison of the average training and generalization rms error using a 2-7-1 neural net and different training sets. The patterns in the fixed sets are approximately equally spaced on the henon attractor.

3. Experimental Results

The properties of the *dynamic pattern selection* will be demonstrated by solving two nonlinear signal prediction tasks. The results will be compared to neural networks trained on fixed training sets. For training a batch mode backpropagation algorithm with adapted learning rate and momentum [10], [9] is used. The generalization performance of the neural networks is estimated by means of an independent validation set $\mathbb{D}_u$, as has been suggested by Hecht-Nielsen [3].

In the first experiment the chaotic time series of the henon model [2]

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} y_n + 1 - a \cdot x_n^2 \\ b \cdot x_n \end{pmatrix} \quad \text{with} \quad (a = 1.4, b = 0.3) \tag{6}$$

is predicted. In table (1) the results for a number of fixed training sets are compared with the dynamic pattern selection. The normalized rms error $E()$ for the training set $\mathbb{D}_s$ and the independent validation set $\mathbb{D}_u$ is given. The fixed training sets with 35-100 patterns give the best results. Using dynamic pattern selection, on average, 69 patterns are selected. The total cost for training is estimated by the number of forward/backward propagations through the net. In the case of dynamic pattern selection the cost is considerably lower than that for training on any of the appropriate fixed training sets.

In figure (1) the relation between the generalization factor ρ_u and the number of training epochs is shown. The small fixed training sets do not guarantee a valid generalization. The fixed training set with 75 patterns achieves a nearly stable generalization factor. Compared to this, the dynamically selected mean of 69 patterns is quite reasonable.

Along with the size of the training set, the distribution of the training patterns has a considerable effect on the generalization results. A typical distribution obtained by the dynamic pattern selection is shown in figure (2). The distribution is obviously not uniform, but reflects the error distribution of the net function.

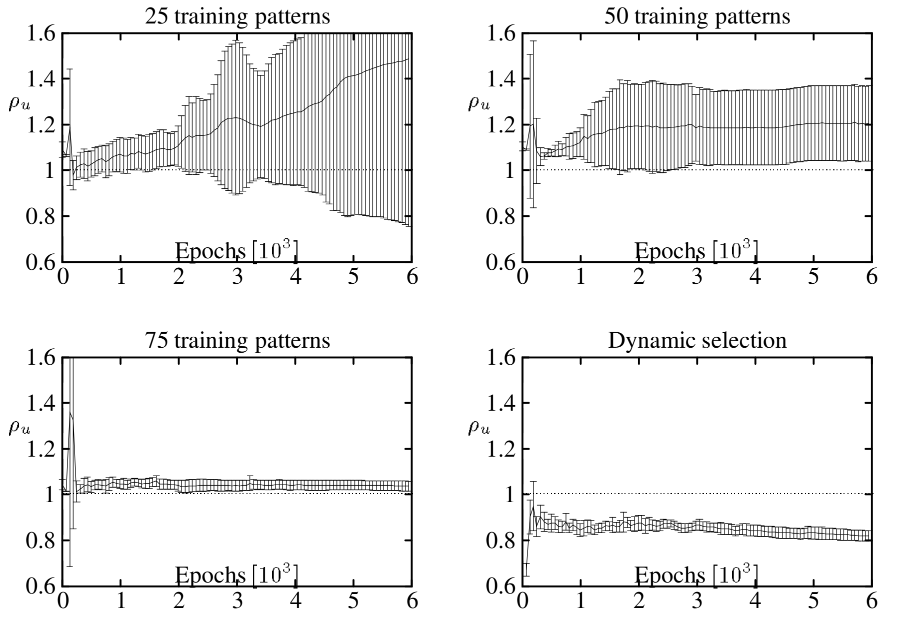

Fig. 1. Generalization factor ρ_u as a function of training epochs for learning to predict the henon model. A fixed training set with 75 patterns is necessary to achieve a generalization factor near one.

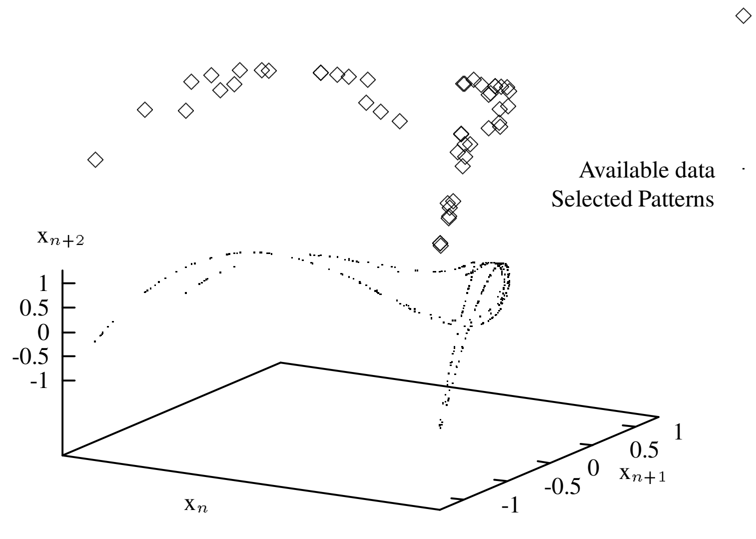

Fig. 2. A typical distribution of training patterns on the prediction function of the henon model. Depicted is the distribution of training patterns subsequent to 1000 training epochs.

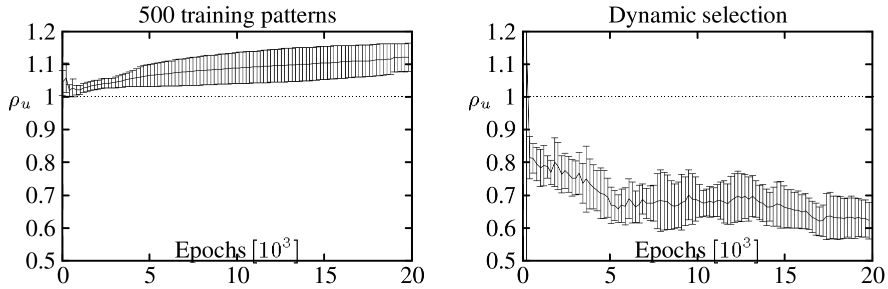

Fig. 3. Generalization factor ρ_u as a function of training epochs for learning to predict the Mackey-Glass model.

training set size	training error $E(\mathbb{D}_s)$	generalization error $E(\mathbb{D}_u)$	generalization factor ρ_u	for/backward propagations
500	2.47e-2 $\pm$ 3.6e-3	2.73e-2 $\pm$ 3.4e-3	1.11	60e+6
207 $\pm$ 11	4.62e-2 $\pm$ 3.9e-3	2.74e-2 $\pm$ 1.4e-3	0.59	18e+6

Tab. 2. Predicting the Mackey-Glass model. Comparison of the average training and generalization rms error using a 6-10-10-1 neural net and different training sets.

The second example is the prediction of the Mackey-Glass model

$$\dot{x}(t) = \frac{a \cdot x(t-\tau)}{(1 + x(t-\tau)^{10})} - b \cdot x(t) \quad \text{with} \quad (a = 0.2, b = 0.1). \tag{7}$$

Lapedes and Farber [4], [5] demonstrated the prediction of the Mackey-Glass time series ($\tau = 30$) using a neural net with six input units, two hidden layers with ten units each and a linear output unit. These settings are chosen here, too. Lapedes and Farber used a fixed training set with 500 examples. Results for this training set and the dynamic pattern selection are shown in table (2). Achieving the same average generalization precision, the dynamically selected training sets contain, on average, 207 training patterns. The cost is lower by a factor of three. In figure (3) the average generalization factor with respect to the training epoch is depicted. In case of the fixed training sets the generalization factor is steadily increasing, this is a consequence of the suboptimal distribution of patterns in those sets.

4. Comparison to online training

It is widely accepted that, in the case of redundant data, the online mode of the backpropagation algorithm will yield superior results than the batch mode. Therefore we compared the computational expenses for online training with *Search-Then-Converge* learning rate schedule [1] and batch mode training with dynamic pattern selection. The neural nets have been trained to predict a piano signal given 15000 training patterns. Despite the preceding optimization for the online training algorithm and the fact that in the case of dynamic training sets only 50 out of the 15000 patterns have been selected (resulting in a considerable overhead for the selection procedure), the total expense for the batch mode, dynamic pattern selection has been lower then that of the online training by a factor of three.

References

[1] C. Darken and J. Moody. Note on learning rate schedules for stochastic optimization. In R. L. and. J.E. Moody and D. Touretzky, editors, *Neural Information Processing Systems 3 (NIPS 90)*, pages 832–838. Morgan Kaufmann Pub., 1991.

[2] P. Grassberger and I. Procaccia. Estimation of the Kolmogerov entropy from chaotic signal. *Physical Review A*, 28(4):2591–2593, 1983.

[3] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley Publishing Company, 1990.

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

[5] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks: Prediction and system modelling. Technical Report LA-UR–87-2662, Los Alamos National Laboratory, 1987.

[6] M. Plutowski and H. White. Selecting concise training sets from clean data. *IEEE Transactions on Neural Networks*, 4(2):305–318, 1993.

[7] A. Röbel. Dynamic selection of training patterns for neural networks: A new method to control the generalization. Technical Report 92-39, Technical University of Berlin, 1992. In German.

[8] A. Röbel. The dynamic pattern selection algorithm: Effective training and controlled generalization of backpropagation neural networks. Technical Report 93-23, Technical University of Berlin, 1993.

[9] A. Röbel. *Neural models of nonlinear dynamical systems and their application to musical signals*. PhD thesis, Technical University of Berlin, 1993.

[10] R. Salomon. *Improving connectionsts learning, based on gradient descend*. PhD thesis, Technical University of Berlin, 1991. In German.