Bootstrapping neural networks

Jürgen Franke
Universität Kaiserslautern

Michael H. Neumann
Humboldt-Universität zu Berlin

May 5, 1998

Abstract

Knowledge about the distribution of a statistical estimator is important for various purposes like, for example, the construction of confidence intervals for model parameters or the determination of critical values of tests. A widely used method to estimate this distribution is the so-called bootstrap which is based on an imitation of the probabilistic structure of the data generating process on the basis of the information provided by a given set of random observations. In this paper we investigate this classical method in the context of artificial neural networks used for estimating a mapping from input to output space. We establish consistency results for bootstrap estimates of the distribution of parameter estimates.

1 Introduction

Neural networks provide a tool for learning an unknown mapping, say m, from input space to output space. Like orthogonal series estimates or smoothing splines, neural networks provide a flexible class of functions which are able to approximate quite general mappings (Hornik et al., 1989). Therefore, in the presence of noise, they provide function estimators which are nonparametric in spirit. This feature has been investigated in detail by White (1990) who showed consistency of the connectionist function estimators provided that the size of the network grows with the size of the training set in a suitable manner. As pointed out by White (1989a), a single feed-forward neural network with independent inputs and noisy outputs is a particular nonlinear regression model with parameters coinciding with the weights of the network connections. These weights are estimated from a given data set by training the network. The reliability of these estimates is important for the ability of the trained network to generalize. White (1989a) has given the asymptotic normal distribution of the estimated weights, including also the case of misspecifications where the mapping m of interest cannot be exactly written as a network function.

The bootstrap of Efron (1979) is an alternative to asymptotic considerations if one is interested in quantifying the performance of statistical estimates. It is now a well tested tool in many areas of linear and nonlinear statistics, e.g., in the context of learning a mapping, for linear regression (Freedman, 1981), nonlinear nonparametric regression (Härdle and Bowman, 1988), linear time series models (Kreiss and Franke, 1992) and nonlinear nonparametric time series models (Franke et al. 1997, Neumann and Kreiss, 1996). The bootstrap provides an adequate and in many cases better approximation to the actual distribution of parameter estimates than standard asymptotic considerations. It can be used for calculating confidence intervals for predictions and critical values for statistical tests, and it is also helpful for model selection and automatic choice of the amount of smoothing in semi- and nonparametric situations (Efron and Tibshirani, 1993, Hall, 1992, Shao and Tu, 1995). For the particular case of a parametric nonlinear regression model, the validity and second-order efficiency of the bootstrap has been described and investigated in practice by Huet and Jolivet (1989), Huet et al. (1990) and Bunke et al. (1995). These results can be applied directly to feedforward neural networks in the correctly specified case, where the mapping m can be represented exactly by a network of the form considered.

Typically, however, finite-dimensional neural networks used for learning a particular mapping are misspecified. In this paper, we describe bootstrap procedures for feedforward neural networks which also cover this misspecified case. For sake of simplicity we restrict ourselves to networks with only one hidden layer and with one linear output unit which, for real-valued mappings, already have the universal approximation property (Hornik et al., 1989). However, our arguments can be generalized in a straightforward manner to multilayer-multioutput networks, where the output nodes do not have to be linear. We admit an arbitrary activation function for the neurons of the hidden layer, satisfying only certain smoothness conditions, such that our results cover multilayer perceptrons with sigmoid activation function as well as radial basis function networks with kernel-type activation function.

In recent years, practitioners like Refenes et al. (1996) have already used the standard bootstrap for, e.g., estimating sampling variablility of neural networks and investigated its performance using simulations. In chapter 2, we provide the theoretical basis for these applications making the difference to correctly specified nonlinear regression models transparent and discussing some pitfalls related to identifiability of network parameters. This residual-based bootstrap is compared with the asymptotic normal approximation in a short simulation study in chapter 3. In chapter 4, we present a different "wild" Bootstrap procedure which is able to cope with situations where the noise in the data and in particular its variance depends on the input.

2 The bootstrap procedure

We consider a training set of independent identically distributed random row vectors (X_t', Y_t), $t = 1, \ldots, N$, where X_t is of dimension p and Y_t is real-valued. Suppose we are interested in the relationship between Y_t and X_t, and we want to estimate the conditional expectation of Y_t given $X_t = x$ (see White, 1989b):

$$m(x) = \mathcal{E}\{Y_t | X_t = x\}.$$

We assume that the residuals $\varepsilon_t = Y_t - m(X_t)$ are independent random variables with mean 0 and finite variance $\mathcal{E}\{\varepsilon_t^2 | X_t = x\} = \sigma_\varepsilon^2(x)$. We want to approximate m by a single hidden layer feedforward network with H hidden units, $H \geq 1$. We write its output given input x as

$$f_H(x, \vartheta) = v_0 + \sum_{h=1}^{H} v_h \psi(\tilde{x}' w_h)$$

where $\vartheta = (w_1', \ldots, w_H', v_0, \ldots, v_H)'$ is the vector of network weights with $w_h' = (w_{0h}, \ldots, w_{ph})$, $h = 1, \ldots, H$, ψ is the (fixed) hidden unit activation function, and $\tilde{x}' = (1, x')$ is the input vector augmented by a bias component 1. The parametrization of the network function is not unique, as certain simple symmetry operations applied to the weight vector obviously do not change the value of $f_H(x, \vartheta)$. For a sigmoid activation function ψ centered around 0, i.e. $\psi(-x) = -\psi(x)$, these symmetry operations correspond to exchange of hidden units and multiplying all weights of connections going into and out of a particular hidden unit by -1. To avoid this ambiguity we consider only weight vectors ϑ lying in a fundamental domain in the sense of Rüger and Ossen (1997). For the case of sigmoid activation functions with $\psi(-x) = -\psi(-x)$, this means that we restrict our attention to parameter vectors ϑ with $v_1 \geq v_2 \geq \ldots \geq v_H \geq 0$. To simplify the proofs, we consider only a compact subset Θ_H of such a fundamental domain.

Now, we train the given network to get the nonlinear least squares estimate $\widehat{\vartheta}_N$ of the weight vector, i.e. the solution of

$$\widehat{D}_N(\vartheta) \equiv \frac{1}{N} \sum_{t=1}^{N} (Y_t - f_H(X_t, \vartheta))^2 = \min_{\vartheta \in \Theta_H}!$$

In the correctly specified case where $m(x) = f_H(x, \vartheta_0)$ for some $\vartheta_0 \in \Theta_H$, it is well-known from nonlinear regression that $\sqrt{N}(\widehat{\vartheta}_N - \vartheta_0)$ is asymptotically for $N \to \infty$ normally distributed with mean 0 and covariance matrix

$$\sigma_\varepsilon^2 [\mathcal{E}\{\nabla f_H(X_t, \vartheta_N) \nabla f_H(X_t, \vartheta_N)'\}]^{-1}.$$

Here and in the following, $\nabla f_H(x, \vartheta)$ denotes the gradient of f_H with respect to ϑ. The analogous result for the misspecified situation is given by White (1989a). Here, there is no true ϑ_0, but $\widehat{\vartheta}_N$ converges to the parameter of the best network function approximator for $m(x)$, i.e. to the solution ϑ_0 of

$$\begin{aligned} D_0(\vartheta) &\equiv \mathcal{E}\,(Y_t - f_H(X_t,\vartheta))^2 \\ &\equiv \int [(m(x) - f_H(x,\vartheta))^2 + \sigma_\varepsilon^2(x)]p(x)dx = \min_{\vartheta\in\Theta_H}! \end{aligned}$$

where we assume that the random vector X_t has a density $p(x)$. If the size of the training set grows ($N \to \infty$), then $\sqrt{N}(\widehat{\vartheta}_N - \vartheta_0)$ is again asymptotically normally distributed with mean 0 and covariance matrix $A(\vartheta_0)^{-1}B(\vartheta_0)A(\vartheta_0)^{-1}$, where

$$\begin{aligned} &A(\vartheta) = \mathcal{E}\,\{\nabla^2 d(\vartheta)\}\,,\ B(\vartheta) = \mathcal{E}\,\{\nabla d(\vartheta)\nabla d(\vartheta)'\}, \\ &d(\vartheta) \equiv d(X_t, Y_t, \vartheta) = (Y_t - f_H(X_t,\vartheta))^2. \end{aligned}$$

Here, ∇^2 denotes the Hessian with respect to ϑ. For these results to hold, some assumptions have to be satisfied:

(A1) The activation function ψ is bounded and twice continuously differentiable with bounded derivatives, and m is bounded.

(A2) $D_0(\vartheta)$ has a unique global minimum at ϑ_0 lying in the interior of Θ_H, and $\nabla^2 D_0(\vartheta_0) = A(\vartheta_0)$ is positive definite.

Besides the asymptotic normal distribution, the bootstrap provides an alternative approximation for the distribution of $\widehat{\vartheta}_N - \vartheta_0$. Following Huet and Jolivet (1989), we first consider the following approach which is adequate for identically distributed noise ε_t. For initializing the bootstrap procedure, we need any uniformly consistent estimator $\widehat{m}_N$ for m, i.e. an estimator for which $\sup_{x\in\text{supp}(p)}\{|\widehat{m}_N(x) - m(x)|\} \longrightarrow 0$ in probability for $N \to \infty$. $\widehat{m}_N$ could be chosen, e.g., as the type of connectionist sieve estimator considered by White (1990) where the complexity of the network is allowed to grow with N, as a spline smoother with roughness penalty converging slowly to 0, or as a kernel-type smoother with bandwidth decreasing with N. Then, the noise variables ε_t can be approximated by

$$\widehat{\varepsilon}_t = Y_t - \widehat{m}_N(X_t)\,,\ \ t = 1, \ldots, N.$$

We know from our assumptions that $\mathcal{E}\,\varepsilon_t = 0$. To avoid a systematic error in the bootstrap we follow Freedman (1981) and center the $\widehat{\varepsilon}_t$:

$$\widetilde{\varepsilon}_t = \widehat{\varepsilon}_t - \frac{1}{N}\sum_{k=1}^{N} \widehat{\varepsilon}_k\,,\ \ t = 1, \ldots, N.$$

Let $\widetilde{F}_N$ denote the sample distribution given by $\widetilde{\varepsilon}_1, \ldots, \widetilde{\varepsilon}_N$. We draw independent bootstrap errors $\varepsilon_1^*, \ldots, \varepsilon_N^*$ from $\widetilde{F}_N$, i.e. for all t :

$$\varepsilon_t^* = \widetilde{\varepsilon}_k \text{ with probability } \frac{1}{N},\ \ k = 1, \ldots, N.$$

In the same manner, we draw bootstrap input vectors $X_1^*, \dots, X_N^*$ randomly with replacement from $X_1, \dots, X_N$, i.e. for all t :

$$X_t^* = X_k \text{ with probability } \frac{1}{N}, \ k = 1, \dots, N.$$

Finally, we form bootstrap outputs as

$$Y_t^* = \widehat{m}_N(X_t^*) + \varepsilon_t^*, \ t = 1, \dots, N,$$

to get a bootstrap training set $(X_1^*, Y_1^*), \dots, (X_N^*, Y_N^*)$.

The basic idea of the bootstrap is the following. As the ε_t and X_t are independent and identically distributed, their sample distributions approximate for N large enough the true distributions. Therefore, the ε_t^* and X_t^* behave similar as the ε_t and X_t. As, by construction, $\widehat{m}_N$ is close to m, the bootstrap outputs show similar random variations as the true outputs. Therefore, the behaviour of estimates like the weights of a network after training should behave similar for the original training set and for the bootstrap training set. The random mechanism generating the bootstrap training set is, however, known to us, and we can repeat the above procedure as often as we like to get a whole family of independent bootstrap training sets $(X_1^*(i), Y_1^*(i)), \dots, (X_N^*(i), Y_N^*(i)), \ i = 1, \dots, B$. Using standard Monte Carlo techniques we can mimic the behaviour of any quantity of interest which is calculated from the training set.

For sake of illustration, let us assume that we are interested in the mean-squared error

$$\mathrm{mse}(x) = \mathcal{E}\left(m(x) - f_H(x, \widehat{\vartheta}_N)\right)^2$$

which we get if we train a network with H hidden units from the original random training set $(X_t, Y_t), \ t = 1, \dots, N$, and use it to estimate the value $m(x)$ of the function of interest for given input x. We get a bootstrap approximation for $\mathrm{mse}(x)$ as

$$\mathrm{mse}^*(x) = \frac{1}{B}\sum_{i=1}^{B}(\widehat{m}_N(x) - f_H(x, \widehat{\vartheta}_{N,i}^*))^2$$

where $\widehat{\vartheta}_{N,i}^*$ is the weight vector after training the network using the i-th bootstrap training set $(X_t^*(i), Y_t^*(i)), \ t = 1, \dots, N$.

Before we illustrate the procedure with some examples in the next chapter, we first state the results guaranteeing that our above handwaving arguments can be made rigorous and that the bootstrap really works for neural network function estimators. We first state a slight extension of White's (1989a) results on the asymptotic distribution of $\widehat{\vartheta}_N$. In particular, we admit dependence of the noise variance on the input which we discuss in chapter 4 in more detail.

(A3) $X_1, X_2, \ldots$ are independent identically distributed random vectors with density $p(x)$. $\varepsilon_1, \varepsilon_2, \ldots$ are independent random variables and

$$\mathcal{E}\left\{\varepsilon_t | X_t = x\right\} = 0\,,\; \mathcal{E}\left\{\varepsilon_t^2 | X_t = x\right\} = \sigma_\varepsilon^2(x) < \infty.$$

We also impose some further technical assumptions to simplify our proofs. They could be relaxed considerably without changing the validity of our results.

(A4) (i) $\sigma_\varepsilon^2(x)$ is continuous and $0 < \delta \leq \sigma_\varepsilon^2(x) \leq \Delta < \infty$ for all x.

(ii) There exist constants C_n such that $\mathcal{E}\left\{|\varepsilon_t|^n | X_t = x\right\} \leq C_n < \infty$ is satisfied for all x, n.

(iii) m is continuous.

In the definition of $D_0(\vartheta)$, the expectation is taken over Y_t and X_t. However, there is a second natural candidate for an optimal parameter, namely, that value of ϑ which provides the best approximation for a given realization of the input variables $X_1, \ldots, X_N$. We consider

$$D_N(\vartheta) = \frac{1}{N}\sum_{t=1}^{N}\left[\left(m(X_t) - f_H(X_t,\vartheta)\right)^2 + \sigma_\varepsilon^2(X_t)\right],$$

and we define ϑ_N as the solution to the minimization problem

$$D_N(\vartheta) = \min_{\vartheta \in \Theta_H}!$$

Instead of $\widehat{\vartheta}_N - \vartheta_0$, we consider its two components $\widehat{\vartheta}_N - \vartheta_N$ and $\vartheta_N - \vartheta_0$ separately which asymptotically are independent.

Theorem 1: *Suppose that (A1) - (A4) are satisfied. Then, for $N \to \infty$,*

$$\sqrt{N}\begin{pmatrix} \widehat{\vartheta}_N - \vartheta_N \\ \vartheta_N - \vartheta_0 \end{pmatrix} \xrightarrow[d]{} \mathcal{N}\left(0, \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}\right),$$

i.e. $\sqrt{N}(\widehat{\vartheta}_N - \vartheta_N)$ and $\sqrt{N}(\vartheta_N - \vartheta_0)$ are asymptotically independent normal random vectors with covariance matrices Σ_1 and Σ_2, respectively, where

$$\begin{aligned} & \Sigma_1 = A(\vartheta_0)^{-1}B_1(\vartheta_0)A(\vartheta_0)^{-1}, \;\; \Sigma_2 = A(\vartheta_0)^{-1}B_2(\vartheta_0)A(\vartheta_0)^{-1} \\ \textit{with} \quad & B_1(\vartheta) = 4 \cdot \int \sigma_\varepsilon^2(x)\nabla f_H(x,\vartheta)\cdot\nabla f_H(x,\vartheta)'p(x)dx \\ & B_2(\vartheta) = 4 \cdot \int (m(x) - f_H(x,\vartheta))^2\nabla f_H(x,\vartheta)\cdot\nabla f_H(x,\vartheta)'p(x)dx \end{aligned}$$

and $A(\vartheta) = \nabla^2 D_0(\vartheta)$ as above.

As an immediate consequence, $\sqrt{N}(\hat{\vartheta}_N - \vartheta_0)$ is asymptotically normal with mean 0 and covariance matrix $\Sigma_1 + \Sigma_2$. In the correctly specified case, Σ_2 is equal to the zero matrix, as there is no effect due to the randomness of the X_t's, that is $\vartheta_N = \vartheta_0$. In contrast, in the misspecified case the randomness of the inputs causes a difference of order $N^{-1/2}$ between the two optimal parameters ϑ_N and ϑ_0.

One manner to prove that the bootstrap works in a particular situation, where the limit distribution of the quantity of interest is known, is to show that the corresponding bootstrap quantity has the same asymptotic behaviour. Let $\widehat{\vartheta}^*_N$ be the weight vector after training the network from the bootstrap training set, i.e. the solution of

$$\widehat{D}^*_N(\vartheta) \equiv \frac{1}{N}\sum_{t=1}^{N}(Y_t^* - f_H(X_t^*, \vartheta))^2 = \min_{\vartheta\in\Theta_H}!$$

The bootstrap procedure of this chapter is based on the assumption that the distribution of ε_t does not depend on X_t. In this case, the analogon of ϑ_N is given by the solution ϑ^*_N of

$$D^*_N(\vartheta) \equiv \frac{1}{N}\sum_{t=1}^{N}\left[\Big(\hat{m}_N(X_t^*) - f_H(X_t^*, \vartheta)\Big)^2 + \hat{\sigma}_\varepsilon^2\right] = \min_{\vartheta\in\Theta_H}!$$

where $\hat{\sigma}_\varepsilon^2 = \frac{1}{N}\sum_{t=1}^{N}\tilde{\varepsilon}_t^2$ estimates σ_ε^2. In contrast to ϑ_N, ϑ^*_N can be calculated as $\hat{m}_N$ and $\hat{\sigma}_\varepsilon^2$ are known. To get the bootstrap version of ϑ_0 we have to replace expectation with respect to the joint distribution of X_t and ε_t by expectation with respect to the joint distribution of X_t^* and ε_t^*. Then, ϑ_0^* is the solution of

$$\begin{aligned} D_0^*(\vartheta) &\equiv \mathcal{E}^*\left(Y_t^* - f_H(X_t^*, \vartheta)\right)^2 \\ &= \frac{1}{N}\sum_{t=1}^{N}\left(\hat{m}_N(X_t) - f_H(X_t, \vartheta)\right)^2 + \hat{\sigma}_\varepsilon^2 = \min_{\vartheta\in\Theta_H}! \end{aligned}$$

as ε_t^* and X_t^* are independent and $\mathcal{E}^*\varepsilon_t^* = \frac{1}{N}\tilde{\varepsilon}_t = 0$. We have to impose an assumption on $\hat{m}_N$:

(A5) $\widehat{m}_N$ is uniformly consistent on the support of p, that is

$$\sup_{x\in\mathrm{supp}(p)}\{|\widehat{m}_N(x) - m(x)|\} = o_P(1).$$

Then, the bootstrap, described above, works by the following theorem which shows that the conditional distribution of $\widehat{\vartheta}^*_N - \vartheta^*_N$ and $\vartheta^*_N - \vartheta^*_0$ given the original data $(X_1, Y_1), \ldots, (X_N, Y_N)$, from which the bootstrap training sets are generated, coincides asymptotically with the distribution of $\widehat{\vartheta}_N - \vartheta_N$ and $\vartheta_N - \vartheta_0$ as given in Theorem 1.

Theorem 2: *Suppose that ε_t is independent of X_t, $t = 1, \dots, N$, i.e. in particular $\sigma_\varepsilon^2(x) \equiv \sigma_\varepsilon^2$, and assume (A1) - (A5). Then,*

$$\mathcal{L}\left(\sqrt{N} \begin{pmatrix} \widehat{\vartheta}^*_N - \vartheta^*_N \\ \vartheta^*_N - \vartheta^*_0 \end{pmatrix} \middle| (X_1, Y_1), \dots, (X_N, Y_N) \right) \xrightarrow[d]{} \mathcal{N}\left(0, \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}\right),$$

where this convergence holds in a uniform manner for $((X_1, Y_1), \dots, (X_N, Y_N)) \in \mathcal{X}_N$ for a suitable sequence of sets $\mathcal{X}_N$ with $P(\mathcal{X}_N) \to 1$.

Remark. The bootstrap also works in the case of deterministic inputs $x_1, \dots, x_N$ of the training set which are systematically selected by the experimenter. We have only to require that the inputs $x_1, \dots, x_N$ behave for increasing N similar to a random sample up to a certain degree. In particular, we need

$$\frac{1}{N} \sum_{t=1}^{N} (m(x_t) - f_H(x_t, \vartheta))^2 \underset{N \to \infty}{\longrightarrow} \int (m(x) - f_H(x, \vartheta))^2 p(x) dx$$

for some probability density $p(x)$. If the x_t are, e.g. equispaced over a fixed finite p-dimensional cube C then the above condition is satisfied for a constant function $p(x) \equiv p_C$.

Remark. At the beginning of this section, we have restricted the weight vector to a fundamental domain by, e.g., requiring $v_1 \geq v_2 \geq \dots \geq v_H \geq 0$ for sigmoid activation functions with $\psi(-x) = -\psi(x)$. This avoids those identifiability problems caused by the common symmetry properties of a feedforward neural network independently of the function which we want to approximate. It may, however, happen that for the optimal weights of the parameter vector ϑ certain weights of outgoing connections coincide, say $v_h = v_{h+1}$. This situation may happen in practice if the function m itself has certain symmetries related to the symmetries of ψ. In such a situation, the bootstrap breaks down if it is used for approximating the random fluctuations of the weights of connections going into the hidden units no. h and $h + 1$ provided one does not take additional precautions to guarantee identifiability. To illustrate the problem, let us consider $m(x) = \psi(1 + x) + \psi(x - 1)$ for the centered logistic function $\psi(u) = (1 + e^{-x})^{-1} - \frac{1}{2}$. m is itself a network function for $H = 2$ hidden units with $w^\circ_{11} = w^\circ_{12} = 1$, $w^\circ_{01} = 1$, $w^\circ_{02} = -1$, $v_0 = 0$, $v_1 = v_2 = 1$, and m is symmetric around 0. The estimate $\widehat{m}_N(x)$ will have similar properties as in the case of identifiable parameters of the network. If we train the network repeatedly, using independent bootstrap samples, we get randomly $\widehat{w}^*_{01} \approx 1$, $\widehat{w}^*_{02} \approx -1$ and $\widehat{w}^*_{01} \approx -1$, $\widehat{w}^*_{02} \approx 1$ with approximately equal probabilities. Therefore the bootstrap estimate of the variance of those weights is too large due to the nonidentifiability of the parameters of m and the corresponding approximate nonidentifiability of the parameters of the network which provides the best approximation of $\widehat{m}_N$. In practice, it is easy to detect such situations as they are characterized by almost identical bootstrap estimates of outgoing weights, say $\widehat{v}_h^*(i) \approx \widehat{v}_{h+1}^*(i),\ \ i = 1, \ldots, B$. To be more precise one can check if the differences $\widehat{v}_h^*(i) - \widehat{v}_{h+1}^*(i),\ \ i = 1, \ldots, B$, are small compared to the sample standard deviation of $\widehat{v}_h^*(i)$ and $\widehat{v}_{h+1}^*(i),\ \ i = 1, \ldots, B$, and in such a case take additional precautions to make the parametrization of the network function by its weights unique.

3 A bootstrap procedure for input-dependent noise

The bootstrap procedure of chapter 2 can be applied only to situations where the noise ε_t does not depend on the input, i.e. if it is additive in the sense of Murata et al. (1994). If we consider their general model, $\varepsilon_t = \varepsilon_t(X_t)$ and, in particular, its variance $\sigma_\varepsilon^2(X_t) = \mathcal{E}\left\{\varepsilon_t^2 | X_t\right\}$ depends on the input X_t which is a common situation in many practical problems. One way to bootstrap function estimates under that circumstances is the "wild bootstrap" or "external bootstrap". In the context of nonparametric nonlinear regression, which we are considering here, it has been discussed by Härdle (1990) and Härdle and Marron (1991). For the connectionist regression estimate, it has the following form.

We generate independent and identically distributed random variables $\eta_1, \ldots, \eta_N$ with mean 0 and variance 1. Then, with $\widehat{\varepsilon}_t = Y_t - \widehat{m}_N(X_t)$ as in chapter 2, we draw pairs $(X_t^*, \widehat{\varepsilon}_t^*)$ randomly from the set $\{(X_1, \widehat{\varepsilon}_1), \ldots, (X_N, \widehat{\varepsilon}_N)\}$, i.e. for all $t = 1, \ldots, N$:

$$(X_t^*, \widehat{\varepsilon}_t^*) = (X_k, \widehat{\varepsilon}_k) \text{ with probability } \frac{1}{N},\ \ k = 1, \ldots, N.$$

We transform $\widehat{\varepsilon}_t^*$ randomly by multiplying it with η_t, which does not change the mean and the variance. Then, we define bootstrap outputs as

$$Y_t^* = \widehat{m}_N(X_t^*) + \eta_t \widehat{\varepsilon}_t^*,\ \ t = 1, \ldots, N.$$

In contrast to the standard bootstrap of chapter 2, the bootstrap noise $\varepsilon_t^* = \eta_t \cdot \widehat{\varepsilon}_t^*$ is generated in a manner depending on the bootstrap input X_t^* to reflect the dependence of ε_t on the input in the original training set.

Let $\hat{\vartheta}_N^{WB}$ denote the wild bootstrap version of $\hat{\vartheta}_N$. As now the distribution of the ε_t depends on X_t, the wild bootstrap analogon of $D_N(\vartheta)$ is given by

$$D_N^*(\vartheta) \equiv \frac{1}{N} \sum_{t=1}^{N} \left[\left(\hat{m}_N(X_t^*) - f_H(X_t^*, \vartheta) \right)^2 + \hat{\varepsilon}_t^2 \right]$$

and, correspondingly, the analogon of $D_0(\vartheta)$ is

$$D_0^*(\vartheta) \ \ \equiv \ \ \mathcal{E}^* \Big(Y_t^* - f_H(X_t^*, \vartheta) \Big)^2$$

$$
\begin{aligned}
&= \frac{1}{N}\sum_{t=1}^{N}\left\{\Big(\hat{m}_N(X_t)-f_H(X_t,\vartheta)\Big)^2 + 2\Big(\hat{m}_N(X_t)-f_H(X_t,\vartheta)\Big)\hat{\varepsilon}_t\,\mathcal{E}\ \eta_t + \hat{\varepsilon}_t^2\,\mathcal{E}\ \eta_t^2\right\}\\
&= \frac{1}{N}\sum_{t=1}^{N}\left[\Big(\hat{m}_N(X_t)-f_H(X_t,\vartheta)\Big)^2 + \hat{\varepsilon}_t^2\right].
\end{aligned}
$$

They differ from $D_N^*(\vartheta), D_0^*(\vartheta)$ of chapter 2 only by terms not depending on ϑ, and, therefore, the wild bootstrap versions ϑ_N^{WB}, ϑ_0^{WB} of ϑ_N, ϑ_0 coincide with $\vartheta_N^*, \vartheta_0^*$ of chapter 2. Analogously to Theorem 2, we obtain consistency of the bootstrap.

Theorem 3: *Suppose that (A1) - (A5) are fulfilled. Then,*

$$
\mathcal{L}\left(\sqrt{N}\begin{pmatrix}\widehat{\vartheta}_N^{WB}-\vartheta_N^{WB}\\ \vartheta_N^{WB}-\vartheta_0^{WB}\end{pmatrix}\middle|(X_1,Y_1),\ldots,(X_N,Y_N)\right)\ \xrightarrow[d]{}\ \mathcal{N}\left(0,\begin{pmatrix}\Sigma_1 & 0\\ 0 & \Sigma_2\end{pmatrix}\right),
$$

where this convergence holds in a uniform manner for $((X_1,Y_1),\ldots,(X_N,Y_N))\in\mathcal{X}_N$ *for a suitable sequence of sets* $\mathcal{X}_N$ *with* $P(\mathcal{X}_N)\to 1$.

4 Two numerical examples

In this section we present the results of a small simulation study to illustrate the performance of the common residual-based bootstrap of chapter 2 and of the wild bootstrap of chapter 3. In both cases we consider a feedforward neural network with one hidden layer and complete connections. As the sigmoid activation function, we choose the centered logistic function

$$
\psi(x) = \frac{1}{1+e^{-x}} - \frac{1}{2}.
$$

The generation of the data and the training of the network have been done using GAUSS 3.1, where, in particular, we have used the BFGS-method as a batch mode algorithm for determining the network weights.

As a first example, we consider as the true function to be estimated

$$
m(x_1,x_2) = (1-x_1^2)_+ + x_1/2,
$$

which is shown in Figure 1.

[Please insert Figure 1 about here]

We approximate m by a network with 2 hidden units, that is

$$f(x,\vartheta) = v_0 + \sum_{h=1}^{2} v_h \psi\left(w_{0h} + w_{1h}x_1 + w_{2h}x_2\right).$$

Figure 2a shows the best approximation of m by $f(x,\vartheta_0)$, where

$$\vartheta_0 = \arg\min_{\vartheta} \int_{-1}^{1}\int_{-1}^{1} |m(x) - f(x,\vartheta)|^2\, dx,$$

corresponding to inputs $X_t = (X_{t1}, X_{t2})$ which are uniformly distributed over the square $[-1,1]\times[-1,1]$. The difference between m and $f(.,\vartheta_0)$ is displayed in Figure 2b.

[Please insert Figures 2a and 2b about here]

We have to estimate the 9-dimensional parameter vector $\vartheta_0 = (w_{01}, w_{11}, w_{21}, w_{02}, w_{12}, w_{22}, v_0, v_1, v_2)'$. We use the training set $(X_1,Y_1),\ldots,(X_N,Y_N)$ with sample size $N = 100$ which obeys the regression model

$$Y_t = m(X_t) + \varepsilon_t,$$

where $X_t \sim Unif([-1,1]^2)$ and $\varepsilon_t \sim \mathcal{N}(0,(0.1)^2)$ are all independent. To approximate the true distribution of $\widehat{\vartheta}_N - \vartheta_0$, we carried out 500 Monte Carlo runs.

To find a typical bootstrap distribution $\mathcal{L}\left(\widehat{\vartheta}^*_N - \vartheta^*_0\middle| (X_1,Y_1),\ldots,(X_N,Y_N)\right)$, we first looked for something like a "most typical" of our 500 samples. We defined this most typical sample as that one which minimizes the difference between the loss $\|\widehat{\vartheta}_N - \vartheta_0\|^2$ and the mean of all of them. Based on this particular sample, we constructed 500 bootstrap samples $((X^*_1(i),Y^*_1(i)),\ldots,(X^*_{100}(i),Y^*_{100}(i)))$, $i = 1,\ldots,100$, according to the description in Section 2. For $\widehat{m}_N$ we used a Nadaraya-Watson kernel estimator with bivariate Gaussian kernel and bandwidths $h_1 = 0.2$ and $h_2 = 1.0$. For each of the bootstrap training sets we calculated the bootstrap estimate $\widehat{\vartheta}^*_{N,i}$, which leads to an estimate of $\mathcal{L}\left(\widehat{\vartheta}^*_N - \vartheta^*_0\middle| (X_1,Y_1),\ldots,(X_N,Y_N)\right)$. Finally, we also consider the asymptotic limit distribution of $\sqrt{N}(\widehat{\vartheta}_N - \vartheta_0)$ as given by White (1989a).

Figure 3a shows estimates of $\mathcal{L}(\widehat{w}_{21} - w_{21})$ (solid line), of $\mathcal{L}(\widehat{w}^*_{21} - w^*_{21}|(X_1,Y_1),\ldots,(X_N,Y_N))$ based on the most typical sample (dashed line) as well as the cumulative distribution function of $\mathcal{N}(0,(\Sigma_1+\Sigma_2)_{33}/N)$ (dashed-dotted line). Estimates of the corresponding densities are displayed in Figure 3b.

[Please insert Figures 3a and 3b about here]

Analogously, Figure 4a shows estimates of $\mathcal{L}(|\widehat{w}_{21} - w_{21}|)$, $\mathcal{L}(|\widehat{w}^*_{21} - w^*_{21}|\;|(X_1,Y_1),\ldots,(X_N,Y_N))$ and $|\mathcal{N}(0,(\Sigma_1+\Sigma_2)_{33}/N)|$, whereas corresponding estimates of the densities are displayed in Figure 4b.

[Please insert Figures 4a and 4b about here]

Generally, we observed a surprisingly good approximation of the true distributions by the bootstrap and the normal approximations. For the other network weights we got similar results, which seems to indicate that the bootstrap of chapter 2 works reasonably well even for moderate sample sizes.

As the second example, we consider heteroscedastic residuals ε_t. In this case, neither the standard asymptotics of White (1989a) nor the residual-based bootstrap of chapter 2 are applicable. To facilitate the graphical representation of the results we consider a one-dimensional input x. As the true function to be estimated we choose the bump function

$$m(x) = \frac{1}{2}x + \frac{2}{3}\varphi(8x), \quad -1 \leq x \leq 1,$$

where φ denotes the density of the standard normal distribution. As a training set, we use independent identically distributed $(X_1, Y_1), \ldots, (X_N, Y_N)$ with $N = 100$ where $X_1, \ldots, X_N$ are uniformly distributed over $[-1, 1]$ and

$$Y_t = m(X_t) + \varepsilon_t$$

with independent zero-mean Gaussian residuals $\varepsilon_1, \ldots, \varepsilon_N$ with standard deviation

$$(\mathcal{E}\,\{\varepsilon_t^2/X_t\})^{\frac{1}{2}} = \sigma(X_t) = \frac{1}{5}\sqrt{0.01 + (\frac{1}{2} + m(X_t))^2}$$

i.e. the variance of a residual is large if the function value to be estimated is large which is a typical heteroscedastic situation. As the basis for the wild bootstrap, we consider a "typical" sample selected in a similar manner as in the first example. Figure 5 shows these data, the true function m and the Nadaraya-Watson kernel estimate $\hat{m}_N$ with bandwidth 0.07.

[Please insert Figure 5 about here]

We approximate $m(x)$ by the network function

$$f(x, \vartheta) = v_0 + \sum_{h=1}^{3} v_h\,\psi(w_{0h} + w_{1h}x),$$

which provides quite a good fit for appropriately chosen weights. We train the corresponding network with 3 neurons in its hidden layer and consider the estimates $f(x_i,\ \hat{\vartheta}_N)$ at the points $x_i = -1 + i/50,\ \ i = 0, \ldots, 100$. We are interested in confidence intervals for $m(x_i),\ \ i = 0, \ldots, 100$. Figure 6a shows the true 90%-confidence intervals joined to form a band together with the true function m based on a Monte Carlo simulation with 200 independent runs.

[Please insert Figure 6a about here]

Based on the one "typical" sample, we approximated the distribution of $f(x_i, \hat{\vartheta}_N)$ by means of the wild bootstrap of chapter 3 with standard normally distributed η_t. Using 500 bootstrap replications we determined 90%-confidence intervals as above. Figures 6b and 6c show these bootstrap confidence intervals together with the kernel estimate $\hat{m}_N$ and with the true function m, resp. For sake of better comparability, Figure 6d shows the true confidence band (solid lines) and the bootstrap confidence bands (dashed lines) in one plot.

[Please insert Figures 6b-d somewhere here]

The wild bootstrap captures the heteroscedasticity of the data quite nicely, and it provides quite a good approximation to the true confidence intervals. There are only two problematic areas as can be seen from Figure 6d: at the right end and around the peak. The former defect is easily explained as we have not corrected the initial kernel estimate $\hat{m}_N$ for the well-known boundary effects for ease of calculations. Using boundary kernels would immediately improve the estimate $\hat{m}_N$ and the bootstrap confidence intervals around the boundary. The fact that the upper limits of the bootstrap confidence intervals around the peak are too large is due to pure chance. Looking at Figure 5, it can be seen that all the residuals (with one exception) around the peak happen to be positive and rather large, and they draw the peak of $\hat{m}_N$ and the corresponding upper bootstrap confidence limit upwards. Apart from these explainable effects, the wild bootstrap provides a good method to quantify the reliability of neural network function estimates in the presence of heteroscedasticity. Mark also, that the wild bootstrap does not assume any knowledge about the particular form of the dependence of the variability of ε_t on X_t at all as, e.g., a classical parametric asymptotic approach similar to White (1989a) but using nonlinear weighted least-squares would have to.

5 Proofs

Proof of Theorem 1:

(i) It is easy to see that for all $\delta > 0$ and $\lambda < \infty$

$$P(|\widehat{D}_N(\vartheta) - D_N(\vartheta)| + |D_N(\vartheta) - D_0(\vartheta)| > N^{\delta - \frac{1}{2}}) = O(N^{-\lambda})$$

uniformly in $\vartheta \in \Theta_H$, using (A1), (A3), (A4). Since $\widehat{D}_N, D_N$ and D_0 are continuous in ϑ, we obtain, by showing that the above result holds simultaneously on a sequence of increasingly fine grids $\Theta_N \subseteq \Theta_H$, that

$$\sup_{\vartheta \in \Theta_H} \{|\widehat{D}_N(\vartheta) - D_N(\vartheta)| + |D_N(\vartheta) - D_0(\vartheta)|\} = o_p(1),$$

which implies that

$$|\widehat{\vartheta}_N - \vartheta_N| + |\vartheta_N - \vartheta_0| = o_p(1).$$

Hence, by (A2) with increasing probability, $\widehat{\vartheta}_N$ and ϑ_N are interior points of Θ_H, that is we have in particular that

$$\nabla \widehat{D}_N(\widehat{\vartheta}_N) = \nabla D_N(\vartheta_N) = \nabla D_0(\vartheta_0) = 0$$

with probability converging to 1 for $N \to \infty$.

(ii) Hence,

$$\begin{aligned} 0 &= \nabla D_N(\vartheta_N) - \nabla D_N(\vartheta_0) + \nabla D_N(\vartheta_0) \\ &= \nabla^2 D_0(\vartheta_0)(\vartheta_N - \vartheta_0) + \frac{2}{N}\sum_{t=1}^{N} \nabla f_H(X_t, \vartheta_0)(f_H(X_t, \vartheta_0) - m(X_t)) + R_1, \quad (1) \end{aligned}$$

where

$$\begin{aligned} R_1 &= \nabla D_N(\vartheta_N) - \nabla D_N(\vartheta_0) - \nabla^2 D_N(\vartheta_0)(\vartheta_N - \vartheta_0) \\ &\quad + [\nabla^2 D_N(\vartheta_0) - \nabla^2 D_0(\vartheta_0)](\vartheta_N - \vartheta_0) \\ &= o_p(||\vartheta_N - \vartheta_0||) \end{aligned}$$

by (A1), (A3). As

$$2\mathcal{E}\{\nabla f_H(X_t, \vartheta_0)(f_H(X_t.\vartheta_0) - m(X_t))\} = \nabla D_0(\vartheta_0) = 0,$$

the middle term of (1) is $O_P(N^{-1/2})$. Thus, we get

$$\nabla^2 D_0(\vartheta_0)(\vartheta_N - \vartheta_0) + o_p(||\vartheta_N - \vartheta_0||) = O_p(N^{-\frac{1}{2}}),$$

which implies, since $\nabla^2 D_0(\vartheta_0)$ is positive definite, that

$$||\vartheta_N - \vartheta_0|| = O_p(N^{-\frac{1}{2}}).$$

Inserting this into (1), we obtain that

$$\vartheta_N - \vartheta_0 = -(\nabla^2 D_0(\vartheta_0))^{-1} \frac{2}{N}\sum_{t=1}^{N} \nabla f_H(X_t, \vartheta_0)(f_H(X_t, \vartheta_0) - m(X_t)) + o_p(N^{-\frac{1}{2}}). \quad (2)$$

(iii) Analogously to the above calculations, we have with probability tending to 1 that

$$\begin{aligned} 0 &= \nabla \widehat{D}_N(\widehat{\vartheta}_N) = \nabla D_N(\widehat{\vartheta}_N) - \frac{2}{N}\sum_{t=1}^{N} \varepsilon_t \nabla f_H(X_t, \widehat{\vartheta}_N) \\ &= \nabla^2 D_0(\vartheta_0)(\widehat{\vartheta}_N - \vartheta_N) - \frac{2}{N}\sum_{t=1}^{N} \nabla f_H(X_t, \vartheta_0)\,\varepsilon_t + R_2, \end{aligned} \tag{3}$$

where, as $\nabla D_N(\vartheta_N) = 0$ with probability tending to 1,

$$\begin{aligned} R_2 &= \nabla D_N(\widehat{\vartheta}_N) - \nabla D_N(\vartheta_N) - \nabla^2 D_0(\vartheta_0)(\widehat{\vartheta}_N - \vartheta_N) \\ &\quad + \frac{2}{N}\sum_{t=1}^{N} \varepsilon_t \left\{ \nabla f_H(X_t, \vartheta_0) - \nabla f_H(X_t, \hat{\vartheta}_N) \right\} \\ &= o_p(||\widehat{\vartheta}_N - \vartheta_N||) + O_p(||\widehat{\vartheta}_N - \vartheta_0|| N^{-\frac{1}{2}}) = o_p(||\hat{\vartheta}_N - \vartheta_N||). \end{aligned}$$

That means

$$\nabla^2 D_0(\vartheta_0)(\widehat{\vartheta}_N - \vartheta_N) + o_p(||\widehat{\vartheta}_N - \vartheta_N||) = O_p(N^{-\frac{1}{2}}),$$

which implies

$$||\widehat{\vartheta}_N - \vartheta_N|| = O_p(N^{-\frac{1}{2}}),$$

and therefore,

$$\widehat{\vartheta}_N - \vartheta_N = (\nabla^2 D_0(\vartheta_0))^{-1} \frac{2}{N}\sum_{t=1}^{N} \nabla f_H(X_t, \vartheta_0)\,\varepsilon_t + o_p(N^{-\frac{1}{2}}) \tag{4}$$

The assertion follows now from (2) and (4) by a multivariate central limit theorem for functions of i.i.d. random vectors. In particular, the asymptotic covariance matrix of $\widehat{\vartheta}_N - \vartheta_N$ and $\vartheta_N - \vartheta_0$ vanishes as $\mathcal{E}\ \{\varepsilon_t \mid X_t = x\} = 0$.

■

Proof of Theorem 2: This proof is very similar to that of Theorem 1. Since $\widehat{D}^*_N(\vartheta)$, $D^*_N(\vartheta)$ and $D^*_0(\vartheta)$ converge uniformly to $D_0(\vartheta)$, one can easily show that

$$|\widehat{\vartheta}^*_N - \vartheta_0| + |\vartheta^*_N - \vartheta_0| + |\vartheta^*_0 - \vartheta_0| = o_p(1). \tag{5}$$

Then we obtain in complete analogy to the previous proof that

$$\mathcal{L}\left(\sqrt{N} \begin{pmatrix} \widehat{\vartheta}^*_N - \vartheta^*_N \\ \vartheta^*_N - \vartheta^*_0 \end{pmatrix} \middle| (X_1, Y_1), \ldots, (X_N, Y_N) \right) \underset{d}{\sim} \mathcal{N}\left(0, \begin{pmatrix} \Sigma^*_1 & 0 \\ 0 & \Sigma^*_2 \end{pmatrix}\right),$$

where Σ^*_1 and Σ^*_2 are analogous to Σ_1 and Σ_2, respectively with $A^*(\vartheta) = \nabla^2 D^*_0(\vartheta)$ and

$$B^*_1(\vartheta) = \frac{4}{N} \cdot \sum_{t=1}^{N} \nabla f_H(X_t, \vartheta) \nabla f_H(X_t, \vartheta)' \hat{\sigma}^2_\varepsilon,$$

$$B_2^*(\vartheta) = \frac{4}{N} \cdot \sum_{t=1}^{N} \Big(\widehat{m}_N(X_t) - f_H(X_t, \vartheta)\Big) \nabla f_H \Big(X_t, \vartheta\Big) \nabla f_H(X_t, \vartheta)'.$$

Because of (5) and (A5), we obtain

$$\Sigma_i^* = \Sigma_i + o_P(1),$$

which finishes the proof. ■

Proof of Theorem 3: This proof is completely analogous to that of Theorem 2, and therefore omitted. ■

References

[1] Bunke, O., Droge, B. and Polzehl, J. (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper 52, SFB 373, Humboldt University, Berlin.

[2] Efron, B. (1979). Bootstrap methods: Another look at the jackknife. Ann. Statist. **7**, 1 - 26.

[3] Efron, B. and Tibshirani, R.J. (1993). An Introduction to the Bootstrap. Chapman and Hall, New York.

[4] Franke, J., Kreiss, J.P. and Mammen, E. (1997). Bootstrap of kernel smoothing in nonlinear time series.

[5] Freedman, D.A. (1981). Bootstrapping regression models. Ann. Statist. **9**, 1218 - 1228.

[6] Hall, P. (1992). The Bootstrap and Edgeworth Expansion. Springer, Berlin-Heidelberg-New York.

[7] Härdle, W. (1990). Applied Nonparametric Regression. Cambridge University Press, Cambridge.

[8] Härdle, W. and Marron, J.S. (1991). Bootstrap simultaneous error bars for nonparametric regression. Ann. Statist. **13**, 1465-1481.

[9] Härdle, W. and Bowman, A. (1988). Bootstrapping in nonparametric regression: Local adaptive smoothing and confidence bands. J. Americ. Statist. Assoc. **83**, 102-110.

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

[11] Huet, S. and Jolivet, E. (1989). Exactitude au second order des intervalles de confiance bootstrap pour les paramétres d'un modele de régression non linéaire. C.R. Acad. Sci. Paris Sér. I. Math. **308**, 429-432.

[12] Huet, S., Jolivet, E. and Messean, A. (1990). Some simulation results about confidence intervals and bootstrap methods in nonlinear regression. Statistics **21**, 369-432.

[13] Kreiss, J.P. and Franke, J. (1992). Bootstrapping stationary autoregressive moving average models. J. Time Ser. Anal. **13**, 297-317.

[14] Murata, N., Yoshizawa, S. and Amari, S. (1994). Network information criterion - Determining the number of hidden units for an artificial neural network model. IEEE Trans. Neural Networks **5**, 865-872.

[15] Neumann, M.H. and Kreiss, J.P. (1996). Bootstrap confidence bands for the autoregression function. Discussion paper 75, SFB 373, Humboldt University, Berlin.

[16] Refenes, A.-P.N., Zapranis, A.D. and Utans, J. (1996). Neural model identification, variable selection and model adequacy. In: Neural Networks in Financial Engineering, A. Weigend et al. ed., World Scientific Publ.

[17] Rüger, St. and Ossen, A. (1997). The metric structure of weightspace. Manuscript. Technical University of Berlin.

[18] Shao, J. and Tu, D. (1995). The Jackknife and Bootstrap. Springer, Berlin-Heidelberg-New York.

[19] White, H. (1989a). Some asymptotic results for learning in single hidden-layer feedforward network models. *J. Amer. Statist. Assoc.* **84**, 1008-1013.

[20] White, H. (1989b). Learning in artificial neural networks: a statistical perspective. Neural Computation **1**, 425-464.

[21] White, H. (1990). Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings. Neural Networks **3**, 535-550.