# Analysis of the rate of convergence of neural network regression estimates which are easy to implement [*]

Alina Braun[1], Michael Kohler[1] and Adam Krzyżak[2,†]

[1] *Fachbereich Mathematik, Technische Universität Darmstadt, Schlossgartenstr. 7, 64289 Darmstadt, Germany, email: braun@mathematik.tu-darmstadt.de, kohler@mathematik.tu-darmstadt.de*

[2] *Department of Computer Science and Software Engineering, Concordia University, 1455 De Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8, email: krzyzak@cs.concordia.ca*

September 25, 2019

**Abstract**

Recent results in nonparametric regression show that for deep learning, i.e., for neural network estimates with many hidden layers, we are able to achieve good rates of convergence even in case of high-dimensional predictor variables, provided suitable assumptions on the structure of the regression function are imposed. The estimates are defined by minimizing the empirical $L_2$ risk over a class of neural networks. In practice it is not clear how this can be done exactly. In this article we introduce a new neural network regression estimate where most of the weights are chosen regardless of the data motivated by some recent approximation results for neural networks, and which is therefore easy to implement. We show that for this estimate we can derive rates of convergence results in case the regression function is smooth. We combine this estimate with the projection pursuit, where we choose the directions randomly, and we show that for sufficiently many repititions we get a neural network regression estimate which is easy to implement and which achieves the one-dimensional rate of convergence (up to some logarithmic factor) in case that the regression function satisfies the assumptions of projection pursuit.

*AMS classification:* Primary 62G08; secondary 62G20.

*Key words and phrases:* curse of dimensionality, neural networks, nonparametric regression, rate of convergence, projection pursuit.

## 1. Introduction

For many years neural networks have been considered as one of the best approaches in nonparametric statistics in view of multivariate statistical applications, in particular in

[*] Running title: *Neural network regression estimates*

[†] Corresponding author. Tel: +1-514-848-2424 ext. 3007, Fax:+1-514-848-2830

pattern recognition and in nonparametric regression (see, e.g., the monographs Hertz, Krogh and Palmer (1991), Devroye, Györfi and Lugosi (1996), Anthony and Bartlett (1999), Györfi et al. (2002), Haykin (2008) and Ripley (2008)). In recent years the focus in applications shifted towards so-called deep learning, where multilayer feedforward neural networks with many hidden layers are fitted to observed data (see, e.g., Schmidhuber (2015) and the literature cited therein).

In this article we study neural network estimates in the context of nonparametric regression with random design. Here, $(X, Y)$ is an $\mathbb{R}^d \times \mathbb{R}$–valued random vector satisfying $\mathbf{E}\{Y^2\} < \infty$, and given a sample of $(X, Y)$ of size $n$, i.e., given a data set

$$\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}, \tag{1}$$

where $(X, Y)$, $(X_1, Y_1)$, ..., $(X_n, Y_n)$ are i.i.d. random variables, the aim is to construct an estimate

$$m_n(\cdot) = m_n(\cdot, \mathcal{D}_n) : \mathbb{R}^d \to \mathbb{R}$$

of the regression function $m : \mathbb{R}^d \to \mathbb{R}$, $m(x) = \mathbf{E}\{Y|X = x\}$ such that the $L_2$ error

$$\int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx)$$

is “small” (see, e.g., Györfi et al. (2002) for a systematic introduction to nonparametric regression and a motivation for the $L_2$ error).

It is well–known that one needs smoothness assumptions on the regression function in order to derive non–trivial rate of convergence results for nonparametric regression estimates (cf., e.g., Theorem 7.2 and Problem 7.2 in Devroye, Györfi and Lugosi (1996) and Section 3 in Devroye and Wagner (1980)). To do this we will use the following definition.

**Definition 1** *Let $p = q + s$ for some $q \in \mathbb{N}_0$ and $0 < s \leq 1$, where $\mathbb{N}_0$ is the set of nonnegative integers. A* ***function*** *$f : \mathbb{R}^d \to \mathbb{R}$ is called $(p, C)$****-smooth****, if for every $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$ with $\sum_{j=1}^d \alpha_j = q$ the partial derivative $\frac{\partial^q f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}$ exists and satisfies*

$$\left| \frac{\partial^q f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}(x) - \frac{\partial^q f}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}(z) \right| \leq C \cdot \|x - z\|^s$$

*for all $x, z \in \mathbb{R}^d$, where $\| \cdot \|$ denotes the Euclidean norm.*

Stone (1982) showed that the optimal minimax rate of convergence in nonparametric regression for $(p, C)$-smooth functions is $n^{-2p/(2p+d)}$. In case that $d$ is large compared to $p$ this rate of convergence is rather slow (so-called curse of dimensionality). One way to circumvent this curse of dimensionality is to impose additional constraints on the structure of the regression function. Stone (1985) assumed that the regression function is additive, i.e., that $m : \mathbb{R}^d \to \mathbb{R}$ satisfies

$$m(x^{(1)}, \dots, x^{(d)}) = m_1(x^{(1)}) + \dots + m_d(x^{(d)}) \quad (x^{(1)}, \dots, x^{(d)} \in \mathbb{R})$$

for some $(p, C)$–smooth univariate functions $m_1, \dots, m_d : \mathbb{R} \to \mathbb{R}$, and showed that in this case suitably defined spline estimates achieve the corresponding univariate rate of convergence. Stone (1994) extended this results to interaction models, where the regression function is assumed to be a sum of functions applied to at most $d^* < d$ components of $x$ and showed in this case that suitably defined spline estimates achieve the $d^*$–dimensional rate of convergence. Other classes of functions which enable us to achieve a better rate of convergence results include single index models, where

$$m(x) = g(a^T x) \quad (x \in \mathbb{R}^d)$$

for some $a \in \mathbb{R}^d$ and $g : \mathbb{R} \to \mathbb{R}$ (cf., e.g., Härdle and Stoker (1989), Härdle, Hall and Ichimura (1993), Yu and Ruppert (2002), Kong and Xia (2007) and Lepski and Serdyukova (2014)) and projection pursuit, where

$$m(x) = \sum_{l=1}^{r} g_l(a_l^T x) \quad (x \in \mathbb{R}^d)$$

for some $r \in \mathbb{N}$, $a_l \in \mathbb{R}^d$ and $g_l : \mathbb{R} \to \mathbb{R}$ $(l = 1, \dots, r)$ (cf., e.g., Friedman and Stuetzle (1981) and Huber (1985)). In Section 22.3 in Györfi et al. (2002) it is shown that suitably defined (nonlinear) least squares estimates achieve in a $(p, C)$–smooth projection pursuit model the univariate rate of convergence $n^{-2p/(2p+1)}$ up to some logarithmic factor.

A generalization of projection pursuit was considered in Horowitz and Mammen (2007). In this paper the case of a regression function, which satisfies

$$m(x) = g\left(\sum_{l_1=1}^{L_1} g_{l_1}\left(\sum_{l_2=1}^{L_2} g_{l_1,l_2}\left(\dots \sum_{l_r=1}^{L_r} g_{l_1,\dots,l_r}(x^{l_1,\dots,l_r})\right)\right)\right),$$

where $g, g_{l_1}, \dots, g_{l_1,\dots,l_r}$ are $(p, C)$-smooth univariate functions and $x^{l_1,\dots,l_r}$ are single components of $x \in \mathbb{R}^d$ (not necessarily different for two different indices $(l_1, \dots, l_r)$), was studied. With the use of a penalized least squares estimate, the rate $n^{-2p/(2p+1)}$ was proven.

The estimates in Horowitz and Mammen (2007) and the one for projection pursuit in Section 22.3 in Györfi et al. (2002) are nonlinear (penalized) least squares estimates, so in practice it is unclear how they can be computed exactly. Friedman and Stuetzle (1981) described easily implementable estimates for projection pursuit, but in their definition several times heuristic simplifications are used, and as a consequence it is unlcear whether for these easily implementable estimates any rate of convergence result can be shown.

Recently it was shown in several papers that neural networks can achieve dimensionality reduction in case the regression function is a composition of (sums of) functions, where each of the function is a function of at most $d^* < d$ variables. The first paper in this respect was Kohler and Krzyżak (2017), where it was shown that in this case suitably defined multilayer neural networks achieve the rate of convergence $n^{-2p/(2p+d^*)}$ (up to some logarithmic factor) in case $p \leq 1$. Bauer and Kohler (2019) showed that this result even holds for $p > 1$ provided the squashing function is suitably chosen.

Schmidt-Hieber (2019) obtained similar results for neural networks with ReLU activation function, and Kohler and Langer (2019) showed that the results of Bauer and Kohler (2019) also hold for very simply constructed fully connected feedforward neural networks. In Kohler, Krzyżak and Langer (2019) it was demonstrated that neural networks are able to circumvent the curse of dimensionality in case the regression function has a low local dimensionality. Results concerning estimation of regression functions which are piecewise polynomials with partitions with rather general smooth boundaries by neural networks have been derived in Imaizumi and Fukamizu (2019).

In all articles above the neural network regression estimate is defined as a nonlinear least squares estimate. For instance, an estimate is defined as the function $m_n \in \mathcal{F}_n$ which minimizes the empirical $L_2$ risk

$$\frac{1}{n} \sum_{i=1}^{n} |Y_i - m_n(X_i)|^2 \tag{2}$$

over a nonlinear class $\mathcal{F}_n$ of neural networks. In practice, it is usually not possible to find the global minimum of the empirical $L_2$ risk over a class of neural networks and one usually tries to find a local minimum using, e.g., the gradient descent algorithm (so–called backpropagation).

There exist quite a few papers which try to show that neural network estimates learned by backpropagation have nice theoretical properties. The most popular approach in this context is the so–called landscape approach. Choromanska et al. (2015) used random matrix theory to derive a heuristic argument showing that the risk of most of the local minima of the empirical $L_2$ risk is not much larger than the risk of the global minimum. For neural networks with special activation function it was possible to validate this claim, see, e.g., Arora et al. (2018), Kawaguchi (2016), and Du and Lee (2018), which have analyzed gradient descent for neural networks with linear or quadratic activation function. But for such neural networks there do not exist good approximation results, consequently, one cannot derive from these results rates of convergence comparable to the ones above for the least squares neural network regression estimates.

Du et al. (2018) analyzed gradient descent applied to neural networks with one hidden layer in case of an input with a Gaussian distribution. They used the expected gradient instead of the gradient in their gradient descent routine, and therefore, their result cannot be used to derive a rate of convergence result similar to the results for the least squares neural network estimates cited above for an estimate learned by the gradient descent. Liang et al. (2018) applied gradient descent to a modified loss function in classification, where it is assumed that the data can be interpolated by a neural network. Here, the last assumption is not satisfied in nonparametric regression and it is unclear whether the main idea (of simplifying the estimation by a modification of the loss function) can also be used in regression setting. Recently it was shown in several papers, see, e.g., Allen-Zhu, Li and Song (2019), Kawaguchi and Huang (2019) and the literature cited therein, that gradient descent leads to a small empirical $L_2$ risk in over-parametrized neural networks. Here, it is unclear what the $L_2$ risk of the estimate is (and a bound on this term is necessary in order to derive results like the ones cited above for the least squares neural network regression estimates). In particular, due to the fact that the networks are over-parametrized, a bound on the empirical $L_2$ risk might be not useful for bounding the $L_2$ risk. And the bound on the $L_2$ risk presented in Kawaguchi and Huang (2019) requires that the weights in the network be small, and it is not clear whether this will be satisfied in an over-parametrized neural network learned by the gradient descent.

So although the above theoretical results for the least squares neural network estimates are quite impressive, there is a big gap between the estimates for which one has proven the above mentioned nice rate of convergence results and the estimates which can be used in practice. And until now, the results derived in the literature for backpropagation are unfortunately not strong enough to narrow this gap.

In this paper we are interested in the following question: If we define a neural network regression estimate theoretically exactly as it is implemented in practice, what rate of convergence result can we show for this estimate? This question was already considered in Braun, Kohler and Walk (2019). There neural network regression estimates with one hidden layer have been considered, where the weights were chosen by minimizing a regularized empirical $L_2$ risk via backpropagation with starting values chosen repeatedly randomly from a special structure adopted to projection pursuit. It was shown in a $(p, C)$–smooth projection pursuit model, i.e., in a projection pursuit model with $(p, C)$–smooth functions, that this easily implementable estimate achieves (up to a logarithmic factor) the rate of convergence $n^{-2p/(2p+1)}$, provided $p \leq 1$.

In the sequel we use a different (but related) approach in order to derive rate of convergence results for easily implementable neural network estimates. We use neural networks with several hidden layers where we choose most of the inner weights of the network in a data-independent way, and where we choose the weights of the output level via regularized least squares estimates. Here the choice of the inner weights is motivated by recent approximation results derived for deep neural networks, and the use of the regularized least squares criterion for the choice of the weights of the output layer leads to estimates which are easy to implement (because they can be computed by solving a linear equation system).

Our first main result is that in this way we can define neural network regression estimates which are easy to implement and which achieve the same rate of convergence results as linear regression estimates (e.g., kernel or spline estimates), i.e., they achieve (up to some logarithmic factor) the optimal minimax rate of convergence $n^{-2p/(2p+d)}$ in case of a $(p, C)$–smooth regression function, for any $p > 0$. For our second main result we define in a projection pursuit model a neural network regression estimate, where we choose the directions of this model several times randomly and define the inner weights independent of the data using these random directions, and where the weights of the output layer are computed by using a regularized least squares criterion. For this estimate we show that for sufficiently many repetitions (of the choices of the random directions) we get an estimate which achieves the one-dimensional rate of convergence (up to some logarithmic factor) in case that the regression function satisfies the assumptions of the projection pursuit. To our knowledge this result is the first result in the literature which shows that there exist estimates which can be easily implemented and which achieve (up to a logarithmic factor) the rate of convergence $n^{-2p/(2p+1)}$ in a $(p, C)$–smooth projection pursuit model for arbitary $p > 0$.

Throughout the paper, the following notation is used: The sets of natural numbers, natural numbers including 0, and real numbers are denoted by $\mathbb{N}$, $\mathbb{N}_0$ and $\mathbb{R}$, respectively. For $z \in \mathbb{R}$, we denote the smallest integer greater than or equal to $z$ by $\lceil z \rceil$. Furthermore we set $z_+ = \max\{z, 0\}$. The Euclidean norm of $x \in \mathbb{R}^d$ is denoted by $\|x\|$ and $\|x\|_\infty$ denotes its supremum norm. For $f : \mathbb{R}^d \to \mathbb{R}$

$$\|f\|_\infty = \sup_{x \in \mathbb{R}^d} |f(x)|$$

is its supremum norm. Let $\mathcal{F}$ be a set of functions $f : \mathbb{R}^d \to \mathbb{R}$, let $x_1, \dots, x_n \in \mathbb{R}^d$ and set $x_1^n = (x_1, \dots, x_n)$. A finite collection $f_1, \dots, f_N : \mathbb{R}^d \to \mathbb{R}$ is called an $\varepsilon-$ cover of $\mathcal{F}$ on $x_1^n$ if for any $f \in \mathcal{F}$ there exists $i \in \{1, \dots, N\}$ such that

$$\frac{1}{n} \sum_{k=1}^{n} |f(x_k) - f_i(x_k)| < \varepsilon.$$

The $\varepsilon$–covering number of $\mathcal{F}$ on $x_1^n$ is the size $N$ of the smallest $\varepsilon$–cover of $\mathcal{F}$ on $x_1^n$ and is denoted by $\mathcal{N}_1(\varepsilon, \mathcal{F}, x_1^n)$.

The outline of this paper is as follows: In Section 2 the newly proposed neural network regression estimates for $(p, C)$–smooth regression functions are defined and a result for the rate of convergence of these estimates is presented. In Section 3 we describe how these estimates can be combined with projection pursuit, and present a rate of convergence result where the easily computable estimate achieves (up to some logarithmic factor) the optimal one-dimensional rate of convergence if the regression function satisfies the assumptions of projection pursuit. The finite sample size performance of our newly proposed estimates on simulated data is illustrated in Section 4. The proofs are given in Section 5.

## 2. A linear (regularized) least squares regression estimate

The starting point in defining a neural network is the choice of an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. Here, we use in the sequel so–called squashing functions, which are nondecreasing and satisfy $\lim_{x \to -\infty} \sigma(x) = 0$ and $\lim_{x \to \infty} \sigma(x) = 1$. An example of a squashing function is the so-called sigmoidal or logistic squasher

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (x \in \mathbb{R}). \tag{3}$$

The network architecture $(L, \mathbf{k})$ depends on a positive integer $L$ called the *number of hidden layers* and a *width vector* $\mathbf{k} = (k_1, \dots, k_L) \in \mathbb{N}^L$ that describes the number of neurons in the first, second, …, $L$-th hidden layer. A multilayer feedforward neural network with architecture $(L, \mathbf{k})$ and sigmoidal function $\sigma$ is a real-valued function $f : \mathbb{R}^d \to \mathbb{R}$ defined by

$$f(x) = \sum_{i=1}^{k_L} c_i^{(L)} \cdot f_i^{(L)}(x) + c_0^{(L)} \tag{4}$$

for some $c_0^{(L)}, \dots, c_{k_L}^{(L)} \in \mathbb{R}$ and for $f_i^{(L)}$'s recursively defined by

$$f_i^{(r)}(x) = \sigma\left(\sum_{j=1}^{k_{r-1}} c_{i,j}^{(r-1)} \cdot f_j^{(r-1)}(x) + c_{i,0}^{(r-1)}\right) \tag{5}$$

for some $c_{i,0}^{(r-1)}, \dots, c_{i,k_{r-1}}^{(r-1)} \in \mathbb{R}$ $(r = 2, \dots, L)$ and

$$f_i^{(1)}(x) = \sigma\left(\sum_{j=1}^{d} c_{i,j}^{(0)} \cdot x^{(j)} + c_{i,0}^{(0)}\right) \tag{6}$$

for some $c_{i,0}^{(0)}, \dots, c_{i,d}^{(0)} \in \mathbb{R}$.

In the sequel we want to use the data (1) in order to choose the weights of the neural network such that the resulting function defined by (4)–(6) is a good estimate of the regression function. To do this, we will fix the network architecture of the neural network and all weights except the weights in the output layer and will use the data (1) together with the principle of (regularized) least squares in order to estimate the weights in the output layer.

## 2.1. Definition of the network architecture

Let $a > 0$ be fixed. The choice of the network architecture and of the values values of most of the weights of our neural network is motivated by the following approximation result of a $(p, C)$-smooth function for $x \in [-a, a]^d$ by a local convex combination of Taylor polynomials: For $M \in \mathbb{N}$ and $\mathbf{i} = (i^{(1)}, \dots, i^{(d)}) \in \{0, \dots, M\}^d$ set

$$x_{\mathbf{i}} = \left(-a + i^{(1)} \cdot \frac{2a}{M}, \dots, -a + i^{(d)} \cdot \frac{2a}{M}\right)$$

and let

$$\{\mathbf{i}_1, \dots, \mathbf{i}_{(M+1)^d}\} = \{0, \dots, M\}^d.$$

For $k \in \{1, \dots, (M+1)^d\}$ let

$$p_{\mathbf{i}_k}(x) = \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\} \\ j_1+\dots+j_d \le q}} \frac{1}{j_1! \cdots j_d!} \cdot \frac{\partial^{j_1+\dots+j_d} f}{\partial^{j_1} x^{(1)} \cdots \partial^{j_d} x^{(d)}}(x_{\mathbf{i}_k}) \cdot (x^{(1)} - x_{\mathbf{i}_k}^{(1)})^{j_1} \cdots (x^{(d)} - x_{\mathbf{i}_k}^{(d)})^{j_d}$$

be the the Taylor polynomial of $f$ with order $q$ around $x_{\mathbf{i}_k}$ and set

$$P(x) = \sum_{k=1}^{(M+1)^d} p_{\mathbf{i}_k}(x) \prod_{j=1}^{d} \left(1 - \frac{M}{2a} \cdot |x^{(j)} - x_{\mathbf{i}_k}^{(j)}|\right)_+, \tag{7}$$

where $z_+ = \max\{z, 0\}$ $(z \in \mathbb{R})$. Since $P(x)$ is a local convex combination of Taylor polynomials of $m$, it is possible to show that for a $(p, C)$–smooth function $m$ we have

$$\sup_{x \in [-a,a]^d} |m(x) - P(x)| \le c_1 \cdot \frac{1}{M^p} \tag{8}$$

(cf., Lemma 5 in Schmidt–Hieber (2019)).

$P(x)$ can be written in the form

$$\sum_{k=1}^{(M+1)^d} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\} \\ j_1+\dots+j_d \leq q}} a_{\mathbf{i}_k,j_1,\dots,j_d} \cdot (x^{(1)} - x_{\mathbf{i}_k}^{(1)})^{j_1} \cdots (x^{(d)} - x_{\mathbf{i}_k}^{(d)})^{j_d} \prod_{j=1}^{d} \left(1 - \frac{M}{2a} \cdot |x^{(j)} - x_{\mathbf{i}_k}^{(j)}|\right)_+$$

with appropriately chosen $a_{\mathbf{i}_k,j_1,\dots,j_d} \in \mathbb{R}$. Our main trick in the sequel is to define appropriate neural networks $f_{net,j_1,\dots,j_d,\mathbf{i}_k}$ which approximate the functions

$$x \mapsto (x^{(1)} - x_{\mathbf{i}_k}^{(1)})^{j_1} \cdots (x^{(d)} - x_{\mathbf{i}_k}^{(d)})^{j_d} \prod_{j=1}^{d} (1 - \frac{M}{2a} \cdot |x^{(j)} - x_{\mathbf{i}_k}^{(j)}|)_+,$$

and to choose the network architecture such that neural networks of the form

$$\sum_{k=1}^{(M+1)^d} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\} \\ j_1+\dots+j_d \leq q}} a_{\mathbf{i}_k,j_1,\dots,j_d} \cdot f_{net,j_1,\dots,j_d,\mathbf{i}_k}(x) \quad (a_{\mathbf{i}_k,j_1,\dots,j_d} \in \mathbb{R})$$

are contained in it. To do this, we let $\sigma(x) = 1/(1+\exp(-x))$ $(x \in \mathbb{R})$ be the logistic squasher, choose $R \geq 1$ and define the following neural networks: The neural network

$$f_{id}(x) = 4R \cdot \sigma\left(\frac{x}{R}\right) - 2R \tag{9}$$

which approximates the function $f(x) = x$ (cf., Lemma 1 below), the neural network

$$\begin{aligned} f_{mult}(x,y) &= \frac{R^2}{4} \cdot \frac{(1+e^{-1})^3}{e^{-2}-e^{-1}} \cdot \left(\sigma\left(\frac{2(x+y)}{R}+1\right) - 2 \cdot \sigma\left(\frac{x+y}{R}+1\right)\right. \\ &\qquad \left. -\sigma\left(\frac{2(x-y)}{R}+1\right) + 2 \cdot \sigma\left(\frac{x-y}{R}+1\right)\right), \end{aligned} \tag{10}$$

which approximates the function $f(x,y) = x \cdot y$ (cf., Lemma 2 below), the neural network

$$f_{ReLu}(x) = f_{mult}(f_{id}(x), \sigma(R \cdot x)) \tag{11}$$

which approximates $f(x) = x_+$ (cf., Lemma 3 below), and the neural network

$$f_{hat,y}(x) = f_{ReLU}\left(\frac{M}{2a} \cdot (x-y) + 1\right) - 2 \cdot f_{ReLU}\left(\frac{M}{2a} \cdot (x-y)\right) + f_{ReLU}\left(\frac{M}{2a} \cdot (x-y) - 1\right)$$

which approximates for fixed $y \in \mathbb{R}$ the function $f(x) = (1 - (M/(2a)) \cdot |x-y|)_+$ (cf., Lemma 4 below).

With these networks we can now define $f_{net,j_1,\dots,j_d,\mathbf{i}_k}$ recursively as follows: We choose $N \geq q$, set $s = \lceil \log_2(N+d) \rceil$ and define for $j_1,\dots,j_d \in \{0,1,\dots,N\}$ and $k \in \{1,\dots,(M+1)^d\}$

$$f_{net,j_1,\dots,j_d,\mathbf{i}_k}(x) = f_1^{(0)}(x),$$

where

$$f_k^{(l)}(x) = f_{mult}\left(f_{2k-1}^{(l+1)}(x), f_{2k}^{(l+1)}(x)\right)$$

for $k \in \{1, 2, \dots, 2^l\}$ and $l \in \{0, \dots, s-1\}$, and

$$f_k^{(s)}(x) = f_{id}(f_{id}(x^{(l)} - x_{\mathbf{i}_k}^{(l)}))$$

for $j_1 + j_2 + \dots + j_{l-1} + 1 \leq k \leq j_1 + j_2 + \dots + j_l$ and $l = 1, \dots, d$,

$$f_{j_1+j_2+\dots+j_d+k}^{(s)}(x) = f_{hat, x_{\mathbf{i}_k}^{(k)}}(x^{(k)})$$

for $k = 1, \dots, d$, and

$$f_k^{(s)}(x) = 1$$

for $k = j_1 + j_2 + \dots + j_d + d + 1, j_1 + j_2 + \dots + j_d + d + 2, \dots, 2^s$. It is easy to see that $f_{net, j_1, \dots, j_d, \mathbf{i}_k}$ is a neural network with $s+2$ hidden layers and at most

$$6 \cdot 2^s,\ 12 \cdot 2^s,\ 2 \cdot 2^s, 2^s,\ \dots, 8, 4$$

neurons in the layers $1, 2, \dots, s+2$, resp. Consequently, this network is contained in the class of all fully connected neural networks with $s+2$ hidden layers and $24 \cdot (N+d)$ neurons in each hidden layer. Furthermore it is easy to see that all weights are bounded in absolute value by $c_2 \cdot \max\{1, M/a, R^2\}$.

## 2.2. Definition of the output weights

We define our neural network estimate $\tilde{m}_n(x)$ by

$$\tilde{m}_n(x) = \sum_{k=1}^{(M+1)^d} \sum_{\substack{j_1, \dots, j_d \in \{0, \dots, N\} \\ j_1 + \dots + j_d \leq N}} a_{\mathbf{i}_k, j_1, \dots, j_d} \cdot f_{net, j_1, \dots, j_d, \mathbf{i}_k}(x),$$

where the coefficients $a_{\mathbf{i}_k, j_1, \dots, j_d}$ are chosen by minimizing

$$\frac{1}{n} \sum_{i=1}^n |Y_i - \tilde{m}_n(X_i)|^2 + \frac{c_3}{n} \cdot \sum_{k=1}^{(M+1)^d} \sum_{\substack{j_1, \dots, j_d \in \{0, \dots, N\} \\ j_1 + \dots + j_d \leq N}} a_{\mathbf{i}_k, j_1, \dots, j_d}^2 \tag{12}$$

for some constant $c_3 > 0$. This regularized linear least squares estimate can be computed by solving a linear equation system. To see this, set

$$J = (M+1)^d \cdot \binom{N+d}{d},$$

let

$$\{B_j : j = 1, \dots, J\} = \left\{f_{net, j_1, \dots, j_d, \mathbf{i}_k}(x) : 1 \leq k \leq (1+M)^d \text{ and } 0 \leq j_1 + \dots + j_d \leq N\right\}$$

and set

$$\mathbf{B} = (B_j(X_i))_{1 \leq i \leq n, 1 \leq j \leq J} \quad \text{and} \quad \mathbf{Y} = (Y_i)_{i=1,\dots,n}.$$

It is easy to see (cf., Supplement for a corresponding proof) that the vector of coefficients of our estimate is the unique solution of the linear equation system

$$\left( \frac{1}{n} \mathbf{B}^T \mathbf{B} + \frac{c_3}{n} \cdot \mathbf{1} \right) \mathbf{a} = \frac{1}{n} \mathbf{B}^T \mathbf{Y}. \tag{13}$$

The value of (12) will be also less than or equal to the value which we get for coefficients equal to zero, hence we have

$$\frac{1}{n} (\mathbf{Y} - \mathbf{B}a)^T (\mathbf{Y} - \mathbf{B}a) + \frac{c_3}{n} \cdot \mathbf{a}^T \mathbf{a} \leq \frac{1}{n} \sum_{i=1}^{n} Y_i^2,$$

which will allow us to derive a bound on the maximal absolute value of our coefficients.

### 2.3. Rate of convergence

**Theorem 1** *Assume that the distribution of $(X, Y)$ satisfies*

$$\mathbf{E} \left( e^{c_4 \cdot |Y|^2} \right) < \infty \tag{14}$$

*for some constant $c_4 > 0$ and that the distribution of $X$ has bounded support supp$(X)$, and let $m(x) = \mathbf{E}\{Y|X = x\}$ be the corresponding regression function. Assume that $m$ is $(p, C)$–smooth, where $p = q + s$ for some $q \in \mathbb{N}_0$ and $s \in (0, 1]$. Define the estimate $\tilde{m}_n$ as in Subsection 2.2, where $\sigma$ is the logistic squasher and where $N \geq q$, $M = M_n = \lceil c_5 \cdot n^{1/(2p+d)} \rceil$, $R = R_n = n^{d+4}$ and $a = a_n = (\log n)^{1/(6(N+d)}$. Set $\beta_n = c_6 \cdot \log(n)$ for some suitably large constant $c_6 > 0$ and define $m_n$ by*

$$m_n(x) = T_{\beta_n} \tilde{m}_n(x)$$

*where $T_\beta z = \max\{\min\{z, \beta\}, -\beta\}$ for $z \in \mathbb{R}$ and $\beta > 0$. Then $m_n$ satisfies for $n$ suffciently large*

$$\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \leq c_7 \cdot (\log n)^3 \cdot n^{-\frac{2p}{2p+d}},$$

*where $c_7 > 0$ does not depend on $n$.*

**Remark 1.** It follows from the proof of Theorem 1 that the result also holds for more general squashing functions than the logistic squasher. More precisely, in case that the definitions of $f_{id}$, $f_{mult}$ and $f_{ReLU}$ are modified as in Lemma 1, Lemma 2 and Lemma 3 below, it suffices to assume that $\sigma$ is Lipschitz continuous and 2–admissible according to Definition 2 below.

# 3. Extension to projection pursuit

In this section we assume that the regression function satisfies

$$m(x) = \sum_{l=1}^{r} g_l \left( a_{(l-1)\cdot d+1} \cdot x^{(1)} + \dots + a_{l \cdot d} \cdot x^{(d)} \right) \quad (x^{(1)}, \dots, x^{(d)} \in \mathbb{R})$$

for some $r \in \mathbb{N}$, some $(p, C)$–smooth functions $g_l : \mathbb{R} \to \mathbb{R}$ $(l = 1, \dots, r)$ and some $\mathbf{a}_l = (a_{(l-1)\cdot d+1}, \dots, a_{l\cdot d})^T \in \mathbb{R}^d$ with $\|\mathbf{a}_l\| = 1$ $(l = 1, \dots, r)$. Our goal is to construct a neural network regression estimate of $m$ which achieves the univariate rate of convergence.

## 3.1. Definition of the network architecture

Let $A > 0$ be fixed. The choice of the network architecture and of the values of most of the weigths of our neural network is motivated by the following approximation result for $x \in [-A, A]^d$: For $M \in \mathbb{N}$ and $i \in \{0, \dots, M\}$ set

$$u_i = -\sqrt{d} \cdot A + i \cdot \frac{2 \cdot \sqrt{d} \cdot A}{M}$$

and let

$$\{i_1, \dots, i_{M+1}\} = \{0, \dots, M\}.$$

We will see in Section 5 below that we can approximate a $(p, C)$–smooth projection pursuit model

$$m(x) = \sum_{l=1}^{r} g_l(\mathbf{a}_l^T x)$$

by choosing $\mathbf{b}_l$ close to $\mathbf{a}_l$ and by choosing an appropriate sum of local convex combinations of polynomials of the form

$$\sum_{l=1}^{r} \sum_{k=1}^{M+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\}, \\ j_1+\dots+j_d \le q}} a_{i_k, j_1, \dots, j_d, \mathbf{b}_l} \cdot (x^{(1)})^{j_1} \cdot \dots \cdot (x^{(d)})^{j_d} \cdot \left( 1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}_l^T x - u_{i_k}| \right)_+ .$$

Our main trick in the sequel is to define appropriate neural networks $f_{net, j_1, \dots, j_d, i_k, \mathbf{b}_l}$ which approximate the functions

$$x \mapsto (x^{(1)})^{j_1} \cdots (x^{(d)})^{j_d} \cdot \left( 1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}_l^T x - u_{i_k}| \right)_+$$

and to choose the network architecture such that neural networks of the form

$$\sum_{l=1}^{r} \sum_{k=1}^{M+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\} \\ j_1+\dots+j_d \le q}} a_{i_k, j_1, \dots, j_d, \mathbf{b}_l} \cdot f_{net, j_1, \dots, j_d, i_k, \mathbf{b}_l}(x) \quad (a_{i_k, j_1, \dots, j_d, \mathbf{b}_l} \in \mathbb{R})$$

are contained in it. To do this, we let $\sigma(x) = 1/(1+\exp(-x))$ $(x \in \mathbb{R})$ be the logistic squasher, choose $R \geq 1$ and define the following neural networks: The neural network $f_{id}(x)$ as in (9) which approximates the function $f(x) = x$ (cf., Lemma 1 below), the neural network $f_{mult}(x,y)$ as in (10) which approximates the function $f(x,y) = x \cdot y$ (cf., Lemma 2 below), the neural network $f_{ReLu}(x)$ as in (11) which approximates $f(x) = x_+$ (cf., Lemma 3 below), and the neural network

$$\begin{aligned}\bar{f}_{hat,y}(x) &= f_{ReLU}\left(\frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot (x-y) + 1\right) - 2 \cdot f_{ReLU}\left(\frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot (x-y)\right) \\ &\quad + f_{ReLU}\left(\frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot (x-y) - 1\right)\end{aligned}$$

which approximates for fixed $y \in \mathbb{R}$ the function $f(x) = (1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |x-y|)_+$ (cf., Lemma 4 below).

With these networks we can now define $f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l}$ recursively as follows: We choose $N \geq q$, set $s = \lceil \log_2(N+1) \rceil$ and define for $l \in \{1,\dots,r\}$, $j_1,\dots,j_d \in \{0,1,\dots,N\}$ and $k \in \{1,\dots,M+1\}$

$$f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l}(x) = f_1^{(0)}(x),$$

where

$$f_k^{(t)}(x) = f_{mult}\left(f_{2k-1}^{(t+1)}(x), f_{2k}^{(t+1)}(x)\right)$$

for $k \in \{1,2,\dots,2^t\}$ and $t \in \{0,\dots,s-1\}$, and

$$f_k^{(s)}(x) = f_{id}(f_{id}(x^{(t)}))$$

for $j_1 + j_2 + \dots + j_{t-1} + 1 \leq k \leq j_1 + j_2 + \dots + j_t$ and $t = 1,\dots,d$,

$$f_{j_1+j_2+\dots+j_d+1}^{(s)}(x) = \bar{f}_{hat,u_{i_k}}(\mathbf{b}_l^T x),$$

and

$$f_k^{(s)}(x) = 1$$

for $k = j_1 + j_2 + \dots + j_d + 2, j_1 + j_2 + \dots + j_d + 3, \dots, 2^s$. As before, it is easy to see that $f_{net,k,j_1,\dots,j_d,\mathbf{b}_l}$ is a neural network with $s+2$ hidden layers and at most

$$6 \cdot 2^s,\ 12 \cdot 2^s,\ 2 \cdot 2^s, 2^s,\ \dots, 8, 4$$

neurons in the layers $1,2,\dots,s+2$, resp. Consequently, this network is contained in the class of all fully connected neural networks with $s+2$ hidden layers and $24 \cdot (N+1)$ neurons in each hidden layer. Furthermore it is easy to see that all weights are bounded in absolute value by $c_8 \cdot \max\{1, M/A, R^2\}$.

## 3.2. Definition of the output weights

For given directions $\mathbf{b}_l$ $(l = 1, \dots, r)$ we define our neural network estimate $\tilde{m}_n(x)$ by

$$\tilde{m}_n(x) = \sum_{l=1,\dots,r} \sum_{k=1}^{M+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \le N}} a_{i_k,j_1,\dots,j_d,\mathbf{b}_l} \cdot f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l}(x),$$

where the coefficients $a_{i_k,j_1,\dots,j_d,\mathbf{b}_l}$ are chosen by minimizing

$$\frac{1}{n}\sum_{i=1}^{n} |Y_i - \tilde{m}_n(X_i)|^2 + \frac{c_3}{n} \cdot \sum_{l=1}^{r} \sum_{k=1}^{M+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \le N}} a^2_{i_k,j_1,\dots,j_d,\mathbf{b}_l} \tag{15}$$

for some constant $c_3 > 0$. This regularized linear least squares estimate can be computed by solving a linear equation system. To see this, set

$$J = r \cdot (M+1) \cdot \binom{N+d}{d},$$

let

$$\begin{aligned}&\{B_j \,:\, j = 1, \dots, J\} \\ &= \{f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l}(x) \,:\, 1 \le l \le r,\, 1 \le k \le M+1 \text{ and } 0 \le j_1 + \dots + j_d \le N\}\end{aligned}$$

and set

$$\mathbf{B} = (B_j(X_i))_{1 \le i \le n, 1 \le j \le J} \quad \text{and} \quad \mathbf{Y} = (Y_i)_{i=1,\dots,n}.$$

As in Subsection 2.3 it is easy to see that the vector of coefficients of our estimate is the unique solution of the linear equation system

$$\left(\frac{1}{n}\mathbf{B}^T\mathbf{B} + \frac{c_3}{n} \cdot \mathbf{1}\right)\mathbf{a} = \frac{1}{n}\mathbf{B}^T\mathbf{Y}. \tag{16}$$

The value of (15) will be also less than or equal to the value which we get for coefficients equal to zero, hence we have

$$\frac{1}{n}(\mathbf{Y} - \mathbf{B}a)^T(\mathbf{Y} - \mathbf{B}a) + \frac{c_3}{n} \cdot \mathbf{a}^T\mathbf{a} \le \frac{1}{n}\sum_{i=1}^{n} Y_i^2, \tag{17}$$

which will allow us to derive a bound on the maximal absolute value of our coefficients.

## 3.3. Choice of the directions

In order to choose $\mathbf{b}_l$ $(l = 1, \dots, r)$, we choose them $I_n$ times independent randomly according to a uniform distribution on $[-1, 1]^d$, compute each time the corresponding outer weigths as in Subsection 3.2, and choose the directions and the corresponding outer weights for our estimate $\tilde{m}_n$, where the empirical $L_2$ risk of the estimate is minimal.

### 3.4. Rate of convergence

**Theorem 2** *Assume that the distribution of $(X, Y)$ satisfies (14) for some constant $c_4 > 0$ and that the distribution of $X$ has bounded support supp$(X)$, and let $m(x) = \mathbf{E}\{Y|X = x\}$ be the corresponding regression function. Let $r \in \mathbb{N}$, $p > 0$ and $C > 0$, and assume that the regression function satisfies*

$$m(x) = \sum_{l=1}^{r} g_l(\mathbf{a}_l^T x) \quad (x \in \mathbb{R}^d)$$

*for some $(p, C)$-smooth functions $g_l : \mathbb{R} \to \mathbb{R}$ and some $\mathbf{a}_l \in \mathbb{R}^d$ with $\|\mathbf{a}_l\| = 1$ $(l = 1, \dots, r)$.*

*Define the estimate $\tilde{m}_n$ as in Subsections 3.1-3.3, where $\sigma$ is the logistic squasher and where $I_n = \lceil c_9 \cdot (\log n)^2 \cdot n^{\frac{r \cdot d}{2p+1}} \rceil$, $N \geq p$, $M = M_n = \lceil c_{10} \cdot n^{1/(2p+1)} \rceil$, $R = R_n = n^3$ and $A = A_n = (\log n)^{1/(6(N+d)}$. Set $\beta_n = c_6 \cdot \log(n)$ for some suitably large constant $c_6 > 0$ and define $m_n$ by*

$$m_n(x) = T_{\beta_n} \tilde{m}_n(x).$$

*Then $m_n$ satisfies for $n$ suffciently large*

$$\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \leq c_{11} \cdot (\log n)^3 \cdot n^{-\frac{2p}{2p+1}},$$

*where $c_{11} > 0$ does not depend on $n$.*

**Remark 2.** In order to compute our estimate, we have to solve $I_n$ times a linear equation system with a quadratic matrix of size $M_n$, for which computing time is proportional to

$$I_n \cdot M_n^2 \approx (\log n)^2 \cdot n^{\frac{r \cdot d + 2}{2p+1}}.$$

Hence in case

$$r \cdot d < 4 \cdot p$$

computing time is $O(n^2)$, so in case that the number $r$ of terms in the projection pursuit model and the dimension $d$ of $X$ are not too large, our estimate can be computed in $O(n^2)$ time.

## 4. Application to simulated data

In this section we illustrate the finite sample size performance of our newly proposed estimate by applying it to simulated data using the software *MATLAB*.

The simulated data which we use is defined as follows: We choose $X$ uniformly distributed on $[-1, 1]^d$, where $d$ is the dimension of the input, $\epsilon$ standard normal and independent of $X$, and we define $Y$ by

$$Y = m_j(X) + \sigma \cdot \lambda_j \cdot \epsilon, \tag{18}$$

where $m_j : [-1,1]^d \to \mathbb{R}$ is described below, $\lambda_j > 0$ is a scaling value defined below and $\sigma$ is chosen from $\{0.05, 0.10\}$ $(j \in \{1,2,3,4\})$. As regression functions we use

$$\begin{aligned} m_1(x_1,x_2) &= \log(0.2 \cdot x_1 + 0.9 \cdot x_2) + \cos\left(\frac{\pi}{\log(0.5 \cdot x_1 + 0.3 \cdot x_2)}\right) \\ &\quad + \exp\left(\frac{1}{50} \cdot (0.7 \cdot x_1 + 0.7 \cdot x_2)\right) + \frac{\tan\left(\pi \cdot (0.1 \cdot x_1 + 0.3 \cdot x_2)^4\right)}{(0.1 \cdot x_1 + 0.3 \cdot x_2)^2}, \end{aligned}$$

$$\begin{aligned} m_2(x_1,x_2,x_3,x_4) &= \tan\left(\sin\left(\pi \cdot (0.2 \cdot x_1 + 0.5 \cdot x_2 - 0.6 \cdot x_3 + 0.2 \cdot x_4)\right)\right) \\ &\quad + \left(0.5 \cdot (x_1 + x_2 + x_3 + x_4)\right)^3 \\ &\quad + \frac{1}{(0.5 \cdot x_1 + 0.3 \cdot x_2 - 0.3 \cdot x_3 + 0.25 \cdot x_4)^2 + 4}, \end{aligned}$$

$$\begin{aligned} m_3(x_1,x_2,x_3,x_4,x_5) &= \log\left(0.5 \cdot (x_1 + 0.3 \cdot x_2 + 0.6 \cdot x_3 + x_4 - x_5)^2\right) \\ &\quad + \sin\left(\pi \cdot (0.7 \cdot x_1 + x_2 - 0.3 \cdot x_3 - 0.4 \cdot x_4 - 0.8 \cdot x_5)\right) \\ &\quad + \cos\left(\frac{\pi}{1 + \sin(0.5 \cdot (x_2 + 0.9 \cdot x_3 - x_5))}\right) \end{aligned}$$

and

$$\begin{aligned} & m_4(x_1,x_2,x_3,x_4,x_5,x_6) \\ &= \exp\left(0.2 \cdot (x_1 + x_2 + x_3 + x_4 + x_5 + x_6)\right) \\ &\quad + \sin\left(\frac{\pi}{2} \cdot (x_1 - x_2 - x_3 + x_4 - x_5 - x_6)\right) \\ &\quad + \frac{1}{(0.3 \cdot x_1 - 0.2 \cdot x_2 + 0.8 \cdot x_3 - 0.5 \cdot x_4 + 0.6 \cdot x_5 - 0.2 \cdot x_6)^2 + 6} \\ &\quad + 0.5 \cdot (x_1 + x_3 - x_5)^3 \end{aligned}$$

$\lambda_j$ is chosen approximately as IQR of a sample of size 100 of $m(X)$, and we use the values $\lambda_1 = 5.04$, $\lambda_2 = 5.57$, $\lambda_3 = 6.8$, and $\lambda_4 = 3.71$. From this distribution we generate a sample of size $n = 100$ and apply our newly proposed neural network regression estimate and compare our results to that of six alternative regression estimates on the same data. Then we compute the $L_2$ errors of these estimates approximately by using the empirical $L_2$ error $\varepsilon_{L_2,\bar{N}}(\cdot)$ on an independent sample of $X$ of size $\bar{N} = 10,000$. Since this error strongly depends on the behavior of the correct function $m_j$, we consider it in relation to the error of the simplest estimate for $m_j$ we can think of, a completely constant function (whose value is the average of the observed data according to the least squares approach). Thus, the scaled error measure we use for evaluation of the estimates is $\varepsilon_{L_2,\bar{N}}(m_{n,i})/\bar{\varepsilon}_{L_2,\bar{N}}(avg)$, where $\bar{\varepsilon}_{L_2,\bar{N}}(avg)$ is the median of 50 independent realizations of the value one obtains if one plugs the average of $n$ observations into $\varepsilon_{L_2,\bar{N}}(\cdot)$. To a certain extent, this quotient can be interpreted as the relative part of the error of the constant estimate that is still contained in the more sophisticated approaches. The resulting scaled errors of course depend on the random sample of $(X, Y)$, and to be able to compare these values nevertheless we repeat the whole computation 50 times and report the median and the interquartile range of the 50 scaled errors for each of our estimates.

We choose the parameters for each of the estimates by splitting of the sample. Here we split our sample in a learning sample of size $n_l = 0.8 \cdot n$ and a testing sample of size $n_t = 0.2 \cdot n$. We compute the estimate for all parameter values from the sets described below using the learning sample, compute the corresponding empirical $L_2$ risk on the testing sample and choose the parameter value which leads to the minimal empirical $L_2$ risk on the testing sample.

Our first three estimates are fully connected neural network estimates where the number of layers is fixed and the number of neurons per layer is chosen adaptively. The estimate *fc-neural-1* has one hidden layer, estimate *fc-neural-3* has three hidden layers, estimate *fc-neural-6* has six hidden layers and the number of neurons per layer is chosen from the set $\{5, 10, 25, 50, 75\}$, $\{3, 6, 9, 12, 15\}$, $\{2, 4, 6, 8, 10\}$, respectively.

Our fourth estimate *kernel* is the Nadaraya-Watson kernel estimate with so-called naive kernel where the bandwith is chosen from the set $\{2^k : k \in \{-5, -4, \dots, 5\}\}$.

Our fifth estimate *neighbor* is a nearest neighbor estimate where the number of nearest neighbors is chosen from the set $\{1, 2, 3\} \cup \{4, 8, 12, 16, \dots, 4 \cdot \lfloor \frac{n_l}{4} \rfloor\}$.

Our sixth estimate *RBF* is the interpoland with radial basis functions where the radial basis functions $\Phi(r) = (1 - r)_+^6 \cdot (35 \cdot r^2 + 18 \cdot r + 3)$ is used and the scaling radius is chosen adaptively.

Our seventh estimate *MARS* is a method which makes use of multivariate adaptive regression splines. For this estimate we use the MATLAB ARESLab toolbox.

Our last estimate *proj-neural* is our newly proposed neural network estimate presented in this paper. Here the following parameters of the estimate are fixed: $N$ is set to 2, $A$ is set to 1, and $R$ is set to $10^6$, and $r$ is set to 4. The parameter $M$ of the estimate is chosen from the set $\{2, 4, 8, 16\}$. In order to accelerate the computation of this estimate we use only $I_n = 50$ random choices for the vectors of directions in the computation of the estimate $m_1$ with noise value 0.05 and $I_n = 400$ random choices for the vectors of directions in the computation of each of the other estimates for each parameter value.

The results are summarized in Table 1 and in Table 2. As we can see from the reported scaled errors, our newly proposed neural network estimate outperforms all other estimates in four out of eight cases. In the other settings our proposed neural network is only outperformed by a fully connected network. Still, in these scenarios we can see that the scaled error results of our estimate are able to compete with those of the fully connected neural networks, in the sense that the values of the former lie within a small range of the best error value.

# 5. Proofs

## 5.1. Approximation results for neural networks

We will use the following assumption on the activation function of our neural network.

|  | $m_1$ |  | $m_2$ |  |
|---|---|---|---|---|
| *noise* | 5% | 10% | 5% | 10% |
| $\bar{\varepsilon}_{L_2,\bar{N}}(avg)$ | 2.586 | 2.5892 | 1.7504 | 1.7504 |
| *approach* | median (IQR) | median (IQR) | median (IQR) | median (IQR) |
| fc-neural-1 | 0.0564 (0.035) | 0.0941 (0.036) | 0.0335 (0.033) | **0.0816 (0.036)** |
| fc-neural-3 | 0.0717 (0.0292) | 0.0898 (0.034) | 0.0373 (0.057) | 0.1055 (0.054) |
| fc-neural-6 | 0.0802 (0.041) | 0.0985 (0.044) | 0.0387 (0.018) | 0.0914 (0.048) |
| kernel | 0.1639 (0.052) | 0.1731 (0.056) | 0.1448 (0.058) | 0.1631 (0.062) |
| neighbor | 0.1254 (0.033) | 0.1452 (0.044) | 0.1449 (0.038) | 0.1824 (0.06) |
| RBF | 0.3740 (0.413) | 0.9022 (1.085) | 0.1622 (0.118) | 0.43 (0.185) |
| MARS | 0.0907 (0.08) | 0.1161 (0.105) | 0.0625 (0.084) | 0.1453 (0.106) |
| proj-neural | **0.047 (0.040)** | **0.0732 (0.040)** | **0.0318 (0.017)** | 0.1113 (0.097) |

Table 1: Median and IQR of the scaled empirical $L_2$ error of estimates for $m_1$ and $m_2$ for sample size $n = 100$.

|  | $m_3$ |  | $m_4$ |  |
|---|---|---|---|---|
| *noise* | 5% | 10% | 5% | 10% |
| $\bar{\varepsilon}_{L_2,\bar{N}}(avg)$ | 4.4779 | 4.4796 | 0.5401 | 0.54 |
| *approach* | median (IQR) | median (IQR) | median (IQR) | median (IQR) |
| fc-neural-1 | 0.2938 (0.070) | 0.3431 (0.279) | **0.1328 (0.067)** | 0.3172 (0.222) |
| fc-neural-3 | 0.2451 (0.201) | **0.2567 0.207** | 0.1787 (0.156) | 0.3800 (0.399) |
| fc-neural-6 | **0.2141 (0.141)** | 0.2688 (0.207) | 0.1393 (0.125) | 0.3272 (0.313) |
| kernel | 0.3516 (0.070) | 0.3713 (0.077) | 0.8009 (0.095) | 0.7961 (0.087) |
| neighbor | 0.3406 (0.066) | 0.3602 (0.075) | 0.4502 (0.107) | 0.5109 (0.128) |
| RBF | 0.3668 (0.234) | 0.5162 (0.112) | 0.2017 (0.080) | 0.6378 (0.312) |
| MARS | 0.4300 (0.708) | 0.4259 (0.393) | 0.5875 (0.288) | 0.6816 (0.328) |
| proj-neural | 0.236 (0.075) | 0.2822 (0.132) | 0.1433 (0.048) | **0.2488 (0.171)** |

Table 2: Median and IQR of the scaled empirical $L_2$ error of estimates for $m_3$ and $m_4$ for sample size $n = 100$.

**Definition 2** *Let $N \in \mathbb{N}_0$. A function $\sigma : \mathbb{R} \to [0,1]$ is called* ***N-admissible****, if it is nondecreasing and Lipschitz continuous and if, in addition, the following three conditions are satisfied:*

*(i) The function $\sigma$ is $N+1$ times continuously differentiable with bounded derivatives.*

*(ii) A point $t_\sigma \in \mathbb{R}$ exists, where all derivatives up to order $N$ of $\sigma$ are nonzero.*

*(iii) If $y > 0$, the relation $|\sigma(y) - 1| \leq \frac{1}{y}$ holds. If $y < 0$, the relation $|\sigma(y)| \leq \frac{1}{|y|}$ holds.*

It is easy to see that the logistic squasher (3) is $N$–admissible for any $N \in \mathbb{N}$ (cf., e.g., Bauer and Kohler (2019)).

**Lemma 1** *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a function, let $R, a > 0$.*
**a)** *Assume that $\sigma$ is two times continuously differentiable and let $t_{\sigma,id} \in \mathbb{R}$ be such that $\sigma'(t_{\sigma,id}) \neq 0$. Then*

$$f_{id}(x) = \frac{R}{\sigma'(t_{\sigma,id})} \cdot \left(\sigma\left(\frac{x}{R} + t_{\sigma,id}\right) - \sigma(t_{\sigma,id})\right)$$

*satisfies for any $x \in [-a,a]$:*

$$|f_{id}(x) - x| \leq \frac{\|\sigma''\|_\infty \cdot a^2}{2 \cdot |\sigma'(t_{\sigma,id})|} \cdot \frac{1}{R}.$$

**b)** *Assume that $\sigma$ is three times continuously differentiable and let $t_{\sigma,sq} \in \mathbb{R}$ be such that $\sigma''(t_{\sigma,sq}) \neq 0$. Then*

$$f_{sq}(x) = \frac{R^2}{\sigma''(t_{\sigma,sq})} \cdot \left(\sigma\left(\frac{2x}{R} + t_{\sigma,sq}\right) - 2 \cdot \sigma\left(\frac{x}{R} + t_{\sigma,sq}\right) + \sigma(t_{\sigma,sq})\right)$$

*satisfies for any $x \in [-a,a]$:*

$$|f_{sq}(x) - x^2| \leq \frac{5 \cdot \|\sigma'''\|_\infty \cdot a^3}{3 \cdot |\sigma''(t_{\sigma,sq})|} \cdot \frac{1}{R}.$$

**Proof.** The result follows in a straightforward way from the proof of Theorem 2 in Scarselli and Tsoi (1998), cf. Lemma 1 in Kohler, Krzyżak and Langer (2019). □
**Remark 3.** In case of the logistic squasher it is easy to see that with the choice $t_{\sigma,id} = 0$ the network $f_{id}$ in Lemma 1 is given by (9).

**Lemma 2** *Let $\sigma : \mathbb{R} \to [0,1]$ be 2-admissible according to Definition 2. Then for any $R > 0$ and any $a > 0$ the neural network*

$$f_{mult}(x,y) \quad = \quad \frac{R^2}{4 \cdot \sigma''(t_\sigma)} \cdot \left(\sigma\left(\frac{2 \cdot (x+y)}{R} + t_\sigma\right) - 2 \cdot \sigma\left(\frac{x+y}{R} + t_\sigma\right)\right.$$

$$-\sigma\left(\frac{2\cdot(x-y)}{R}+t_\sigma\right)+2\cdot\sigma\left(\frac{x-y}{R}+t_\sigma\right)\Bigg)$$

*satisfies for any $x \in [-a,a]$:*

$$|f_{mult}(x,y)-x\cdot y| \leq \frac{20\cdot\|\sigma'''\|_\infty\cdot a^3}{3\cdot|\sigma''(t_\sigma)|}\cdot\frac{1}{R}.$$

**Proof.** See Lemma 2 in Kohler, Krzyżak and Langer (2019). □

**Remark 4.** In case of the logistic squasher it is easy to see that with the choice $t_\sigma = 1$ the network $f_{mult}$ in Lemma 2 is given by (10).

**Lemma 3** *Let $\sigma : \mathbb{R} \to [0,1]$ be 2-admissible according to Definition 2. Let $f_{mult}$ be the neural network from Lemma 2 and let $f_{id}$ be the network from Lemma 1. Assume*

$$a \geq 1 \quad and \quad R \geq \frac{\|\sigma''\|_\infty\cdot a}{2\cdot|\sigma'(t_{\sigma.id})|}. \tag{19}$$

*Then the neural network*

$$\begin{aligned} f_{ReLU}(x) &= f_{mult}\left(f_{id}(x),\sigma\left(R\cdot x\right)\right) \\ &= \sum_{k=1}^{4} d_k\cdot\sigma\left(\sum_{i=1}^{2} b_{k,i}\cdot\sigma(a_i\cdot x+t_\sigma)+b_{k,3}\cdot\sigma(a_3\cdot x)+t_\sigma\right) \end{aligned}$$

*satisfies*

$$|f_{ReLU}(x)-\max\{x,0\}| \leq 56\cdot\frac{\max\{\|\sigma''\|_\infty,\|\sigma'''\|_\infty,1\}}{\min\{2\cdot|\sigma'(t_{\sigma.id})|,|\sigma''(t_\sigma)|,1\}}\cdot a^3\cdot\frac{1}{R}$$

*for all $x \in [-a,a]$.*

**Proof.** See Lemma 3 in Kohler, Krzyżak and Langer (2019). □

**Lemma 4** *Let $M \in \mathbb{N}$ and let $\sigma : \mathbb{R} \to [0,1]$ be 2-admissible according to Definition 2. Let $a > 0$ and*

$$R \geq \frac{\|\sigma''\|_\infty\cdot(M+1)}{2\cdot|\sigma'(t_{\sigma.id})|},$$

*let $y \in [-a,a]$ and let $f_{ReLU}$ be the neural network of Lemma 3. Then the network*

$$\begin{aligned} f_{hat,y}(x) &= f_{ReLU}\left(\frac{M}{2a}\cdot(x-y)+1\right)-2\cdot f_{ReLU}\left(\frac{M}{2a}\cdot(x-y)\right) \\ &\quad +f_{ReLU}\left(\frac{M}{2a}\cdot(x-y)-1\right) \end{aligned}$$

*satisfies*

$$\left|f_{hat,y}(x)-\left(1-\frac{M}{2a}\cdot|x-y|\right)_+\right| \leq 1792\cdot\frac{\max\{\|\sigma''\|_\infty,\|\sigma'''\|_\infty,1\}}{\min\{2\cdot|\sigma'(t_{\sigma.id})|,|\sigma''(t_\sigma)|,1\}}\cdot M^3\cdot\frac{1}{R}$$

*for all $x \in [-a,a]$.*

**Proof.** Since

$$(1-\frac{M}{2a}\cdot|x|)_+=\max\{\frac{M}{2a}\cdot x+1,0\}-2\cdot\max\{\frac{M}{2a}\cdot x,0\}+\max\{\frac{M}{2a}\cdot x-1,0\}\quad(x\in\mathbb{R})$$

the result is an easy consequence of Lemma 3 (applied with $M+1$ instead of $a$). □

**Lemma 5** *Let $M\in\mathbb{N}$ and let $\sigma:\mathbb{R}\to[0,1]$ be 2-admissible according to Definition 2. Let $a\geq 1$ and*

$$R \;\geq\; \max\Bigg\{\frac{\|\sigma''\|_\infty\cdot(M+1)}{2\cdot|\sigma'(t_{\sigma,id})|},\frac{9\cdot\|\sigma''\|_\infty\cdot a}{|\sigma'(t_{\sigma,id})|}, \frac{20\cdot\|\sigma'''\|_\infty}{3\cdot|\sigma''(t_\sigma)|}\cdot 3^{3\cdot 3^s}\cdot a^{3\cdot 2^s},1792\cdot\frac{\max\{\|\sigma''\|_\infty,\|\sigma'''\|_\infty,1\}}{\min\{2\cdot|\sigma'(t_{\sigma,id})|,|\sigma''(t_\sigma)|,1\}}\cdot M^3\Bigg\} \tag{20}$$

*and let $y\in[-a,a]^d$. Let $N\in\mathbb{N}$ and let $j_1,\dots,j_d\in\mathbb{N}_0$ such that $j_1+\dots+j_d\leq N$, and set $s=\lceil\log_2(N+d)\rceil$. Let $f_{id}$, $f_{mult}$ and $f_{hat,z}$ (for $z\in\mathbb{R}$) be the neural networks defined in Lemma 1, Lemma 2 and Lemma 4, resp. Define the network $f_{net,j_1,\dots,j_d,y}$ by*

$$f_{net,j_1,\dots,j_d,y}(x)=f_1^{(0)}(x),$$

*where $f_1^{(0)}$ is defined by backward recursion as follows:*

$$f_k^{(l)}(x)=f_{mult}\left(f_{2k-1}^{(l+1)}(x),f_{2k}^{(l+1)}(x)\right)$$

*for $k\in\{1,2,\dots,2^l\}$ and $l\in\{0,\dots,s-1\}$, and*

$$f_k^{(s)}(x)=f_{id}(f_{id}(x^{(l)}-y^{(l)}))$$

*for $j_1+j_2+\dots+j_{l-1}+1\leq k\leq j_1+j_2+\dots+j_l$ and $l=1,\dots,d$,*

$$f_{j_1+j_2+\dots+j_d+k}^{(s)}(x)=f_{hat,y^{(k)}}(x^{(k)})$$

*for $k=1,\dots,d$, and*

$$f_k^{(s)}(x)=1$$

*for $k=j_1+j_2+\dots+j_d+d+1,j_1+j_2+\dots+j_d+d+2,\dots,2^s$. Then we have for any $x\in[-a,a]^d$:*

$$\left|f_{net,y}(x)-(x^{(1)}-y^{(1)})^{j_1}\cdots(x^{(d)}-y^{(d)})^{j_d}\prod_{j=1}^d(1-\frac{M}{2a}\cdot|x^{(j)}-y^{(j)}|)_+\right|$$
$$\leq c_{12}\cdot 3^{3\cdot 3^s}\cdot a^{3\cdot 2^s}\cdot M^3\cdot\frac{1}{R}.$$

**Proof.** The result follows from Lemma 1, Lemma 2 and Lemma 4 in a straightforward but technical way using an induction. A complete proof can be found in the Supplement. □

**Remark 5.** The result can be analogously stated for our estimate in the context of the projection pursuit model. The corresponding statement and a complete proof can be found in the Supplement.

## 5.2. Approximation of a projection pursuit model by piecewise polynomials

**Lemma 6** *Let $p = q + s$ for some $q \in \mathbb{N}_0$ and $s \in (0, 1]$. Let $C > 0$, $r \in \mathbb{N}$, $g_l : \mathbb{R} \to \mathbb{R}$ $(p, C)$-smooth functions $(l = 1, \dots, r)$ and $\mathbf{a}_l \in \mathbb{R}^d$ $(l = 1, \dots, r)$. Set*

$$m(x) = \sum_{l=1}^{r} g_l(\mathbf{a}_l^T x) \quad (x \in \mathbb{R}^d).$$

*For $\mathbf{b}_l \in \mathbb{R}^d$ $(l = 1, \dots, r)$ set*

$$g(x) = \sum_{l=1}^{r} \sum_{j=0}^{q} \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j,$$

*where $g_l^{(j)}$ denotes the $j$–th derivative of $g_l$. Then we have for any $x \in \mathbb{R}^d$*

$$|m(x) - g(x)| \leq \frac{r \cdot d^p \cdot C}{q!} \cdot \|x\|_\infty^p \cdot \|\mathbf{a}_l - \mathbf{b}_l\|_\infty^p.$$

**Proof.** By the proof of Lemma 11.1 in Györfi et al. (2002) we have for any $z \in \mathbb{R}$

$$\left| g_l(u) - \sum_{j=0}^{q} \frac{g_l^{(j)}(z)}{j!} \cdot (u - z)^j \right| \leq \frac{1}{q!} \cdot C \cdot |u - z|^p \quad (u \in \mathbb{R}).$$

Applying this with $u = \mathbf{a}_l^T x$ and $z = \mathbf{b}_l^T x$ we get

$$\begin{aligned}
&|m(x) - g(x)| \\
&\leq \sum_{l=1}^{r} \left| g_l(\mathbf{a}_l^T x) - \sum_{j=0}^{q} \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot (\mathbf{a}_l^T x - \mathbf{b}_l^T x)^j \right| \\
&\leq \sum_{l=1}^{r} \frac{1}{q!} \cdot C \cdot |\mathbf{a}_l^T x - \mathbf{b}_l^T x|^p \\
&\leq \frac{r \cdot d^p \cdot C}{q!} \cdot \|x\|_\infty^p \cdot \|\mathbf{a}_l - \mathbf{b}_l\|_\infty^p.
\end{aligned}$$

□

**Lemma 7** *Let $p = q + s$ for some $q \in \mathbb{N}_0$ and $s \in (0, 1]$. Let $C > 0$, $r \in \mathbb{N}$, $g_l : \mathbb{R} \to \mathbb{R}$ $(p, C)$-smooth functions $(l = 1, \dots, r)$ and $\mathbf{a}_l \in \mathbb{R}^d$ with $\|\mathbf{a}_l\| = 1$ and $\mathbf{b}_l \in [-1, 1]^d$ $(l = 1, \dots, r)$. Let $A \geq 1$, $M \in \mathbb{N}$, set*

$$u_i = -\sqrt{d} \cdot A + i \cdot \frac{2 \cdot \sqrt{d} \cdot A}{M} \quad (i = 0, \dots, M)$$

*and* $\{i_1, \dots, i_{M+1}\} = \{0, \dots, M\}$*. Then there exist polynomials* $p_{i_k,l} : \mathbb{R}^d \to \mathbb{R}$ *of total degree* $q$*, which depend on* $\mathbf{a}_l$ *and* $\mathbf{b}_l$ *and where all coefficients are bounded in absolute value by*

$$(q+1) \cdot 2^p \cdot d^{3p/2} \cdot A^p \cdot \max_{l \in \{1,\dots,r\}, j \in \{0,\dots,q\}} \|g_l^{(j)}\|_\infty, \cdot (\max\{\max_{l=1,\dots,r} \|\mathbf{a}_l - \mathbf{b}_l\|_\infty, 1\})^p,$$

*such that we have for all* $x \in [-A, A]^d$

$$\left| \sum_{l=1}^r \sum_{j=0}^q \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j - \sum_{l=1}^r \sum_{k=1}^{M+1} p_{i_k,l}(x) \cdot \left(1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}_l^T x - u_{i_k}|\right)_+ \right|$$
$$\leq r \cdot 2^p \cdot (p+1) \cdot C \cdot d^{3p/2} \cdot A^{2p} \cdot \left( \max\left\{ \frac{1}{M}, \max_{l=1,\dots,r} \|\mathbf{a}_l - \mathbf{b}_l\|_\infty \right\} \right)^p.$$

**Proof.** Let $p_{l,j,i_k}$ be the Taylor polynomial of $g_l^{(j)}$ of degree $q-j$ around $u_{i_k}$. Because of the $(p-j, C)$-smoothness of $g_l^{(j)}$ Lemma 11.1 in Györfi et al. (2002) implies that we have

$$\left| g_l^{(j)}(\mathbf{b}_l^T x) - p_{l,j,i_k}(\mathbf{b}_l^T x) \right| \leq \frac{1}{(q-j)!} \cdot C \cdot |\mathbf{b}_l^T x - u_{i_k}|^{(p-j)}.$$

From this we can conclude for $x \in [-A, A]^d$

$$\left| \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j - \frac{p_{l,j,i_k}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j \right|$$
$$\leq \frac{1}{(q-j)!} \cdot C \cdot d^j \cdot A^j \cdot \left( \max\left\{ |\mathbf{b}_l^T x - u_{i_k}|, \|\mathbf{a}_l - \mathbf{b}_l\|_\infty \right\} \right)^p.$$

Using

$$\sum_{k=1}^{M+1} \left(1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}_l^T x - u_{i_k}|\right)_+ = 1$$

for $x \in [-A, A]^d$, this in turn implies for $x \in [-A, A]^d$

$$\left| \sum_{j=0}^q \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j \right.$$
$$\left. - \sum_{j=0}^q \sum_{k=1}^{M+1} \frac{p_{l,j,i_k}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j \cdot \left(1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}_l^T x - u_{i_k}|\right)_+ \right|$$
$$\leq \sum_{j=0}^q \sum_{k=1}^{M+1} \left| \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j - \frac{p_{l,j,i_k}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j \right|$$
$$\cdot \left(1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}_l^T x - u_{i_k}|\right)_+$$

$$\leq \sum_{j=0}^{q} \max_{\substack{i_k \in \{0,\dots,M\}, \\ |b_l^T x - u_{i_k}| \leq 2 \cdot \sqrt{d} \cdot A/M}} \left| \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j - \frac{p_{l,j,i_k}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j \right|$$

$$\leq (q+1) \cdot C \cdot d^q \cdot A^q \left( \max \left\{ \frac{2 \cdot \sqrt{d} \cdot A}{M}, \|\mathbf{a}_l - \mathbf{b}_l\|_\infty \right\} \right)^p .$$

With

$$p_{i_k,l}(x) = \sum_{j=0}^{q} \frac{p_{l,j,i_k}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j$$

we get the assertion. □

## 5.3. Auxiliary results

**Lemma 8** *Let $\beta_n = c_6 \cdot \log(n)$ for some suitably large constant $c_6 > 0$. Assume that the distribution of $(X,Y)$ satisfies (14) for some constant $c_4 > 0$ and that the regression function $m$ is bounded in absolute value. Let $\mathcal{F}_n$ be a set of functions $f : \mathbb{R}^d \to \mathbb{R}$ and assume that the estimate $m_n$ satisfies*

$$m_n = T_{\beta_n} \tilde{m}_n$$

*and*

$$\tilde{m}_n(\cdot) = \tilde{m}_n(\cdot, (X_1,Y_1), \dots, (X_n,Y_n)) \in \mathcal{F}_n$$

*and*

$$\frac{1}{n} \sum_{i=1}^{n} |Y_i - \tilde{m}_n(X_i)|^2 \leq \min_{l \in \Theta_n} \left( \frac{1}{n} \sum_{i=1}^{n} |Y_i - g_{n,l}(X_i)|^2 + pen_n(g_{n,l}) \right)$$

*for some nonempty parameter set $\Theta_n$, some random functions $g_{n,l} : \mathbb{R}^d \to \mathbb{R}$ and some deterministic penalty terms $pen_n(g_{n,l}) \geq 0$, where the random function $g_{n,l} : \mathbb{R}^d \to \mathbb{R}$ depend only on random variables*

$$\mathbf{b}_1^{(1)}, \dots, \mathbf{b}_r^{(1)}, \dots, \mathbf{b}_1^{(I_n)}, \dots, \mathbf{b}_r^{(I_n)},$$

*which are independent of $(X_1,Y_1)$, $(X_2,Y_2)$, ...*

*Then $m_n$ satisfies*

$$\begin{aligned} &\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \\ &\leq \frac{c_{13} \cdot (\log n)^2 \cdot \left( \log \left( \sup_{x_1^n \in (supp(X))^n} \mathcal{N}_1 \left( \frac{1}{n \cdot \beta_n}, \mathcal{F}_n, x_1^n \right) \right) + 1 \right)}{n} \\ &\quad + 2 \cdot \mathbf{E} \left( \min_{l \in \Theta_n} \int |g_{n,l}(x) - m(x)|^2 \mathbf{P}_X(dx) + pen_n(g_{n,l}) \right) \end{aligned}$$

*for $n > 1$ and some constant $c_{13} > 0$, which does not depend on $n$.*

**Proof.** This lemma follows in a straightforward way from the proof of Theorem 1 in Bagirov et al. (2009). A complete version of the proof is given in the Supplement. □

In order to bound the covering number $\mathcal{N}_1\left(\frac{1}{n\cdot\beta_n}, \mathcal{F}_n, x_1^n\right)$ we will use the following lemma.

**Lemma 9** *Let $a > 0$ and let $d, N, J_n \in \mathbb{N}$ be such that $J_n \leq n^{c_{14}}$ and set $\beta_n = c_6 \cdot \log n$. Let $\sigma$ be 2–admissible according to Definition 2. Let $\mathcal{F}$ be the set of all functions defined by (4), (5) and (6) where $k_1 = k_2 = \dots = k_L = 24 \cdot (N + d)$ and the weights are bounded in absolute value by $c_{15} \cdot n^{c_{16}}$. Set*

$$\mathcal{F}^{(J_n)} = \left\{ \sum_{j=1}^{J_n} a_j \cdot f_j \, : \, f_j \in \mathcal{F} \quad \textit{and} \quad \sum_{j=1}^{J_n} a_j^2 \leq c_{17} \cdot n^{c_{18}} \right\}.$$

*Then we have for $n > 1$*

$$\log\left( supp_{x_1^n \in [-a,a]^{d\cdot n}} \mathcal{N}_1\left(\frac{1}{n\cdot\beta_n}, \mathcal{F}^{(J_n)}, x_1^n\right)\right) \leq c_{19} \cdot \log n \cdot J_n$$

*for some constant $c_{19}$ which depends only on $L$, $N$, $a$ and $d$.*

**Proof.** Since the networks in $\mathcal{F}^{(J_n)}$ are linear combinations of $J_n$ fully connected neural networks with $L$ hidden layers, a bounded number of neurons in each hidden layers and all weights bounded by a polynomial in $n$, the result follows by combining Lemma 16.6 in Györfi et al. (2002) with Lemma 7 in the Supplement of Bauer et al. (2019). □

## 5.4. Proof of Theorem 1

Since $supp(\mathbf{P}_X)$ is bounded and $m$ is $(p, C)$–smooth, we conclude that $m$ is bounded in absolute value, and we can assume without loss of generality that $supp(X) \subseteq [-a_n, a_n]^d$ and $\|m\|_\infty \leq \beta_n$.

Let $\mathcal{F}$ be the set of all functions defined by (4), (5) and (6) where $L = s + 2 = \lceil \log_2(N + d) \rceil + 2$, where $k_1 = k_2 = \dots = k_L = 24 \cdot (N + d)$ and where the weights are bounded in absolute value by $n^{c_{20}}$. Set

$$\mathcal{F}^{(J_n)} = \left\{ \sum_{j=1}^{J_n} a_j \cdot f_j \, : \, f_j \in \mathcal{F} \quad \text{and} \quad \sum_{j=1}^{J_n} a_j^2 \leq c_{21} \cdot n \right\}$$

for $c_{21}$ chosen below, where

$$J_n = (M_n + 1)^d \cdot |\{(j_1, \dots, j_d) \, : \, j_1, \dots, j_d \in \{0, \dots, N\}, \, j_1 + \dots + j_d \leq N\}|$$

Then $J_n \leq (M_n + 1)^d \cdot (N + 1)^d$.

Let

$$g_n(x) = \sum_{k=1}^{(M_n+1)^d} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\} \\ j_1+\dots+j_d \le q}} \frac{1}{j_1! \cdots j_d!} \cdot \frac{\partial^{j_1+\dots+j_d} m}{\partial^{j_1} x^{(1)} \dots \partial^{j_d} x^{(d)}}(x_{\mathbf{i}_k}) \cdot f_{net,j_1,\dots,j_d,\mathbf{i}_k}(x).$$

Because of the $(p, C)$–smoothness of $m$ we know that

$$\max_{k \in \{1,\dots,(M_n+1)^d, j_1,\dots,j_d \in \{0,\dots,q\}, j_1+\dots+j_d \le q} \left| \frac{\partial^{j_1+\dots+j_d} m}{\partial^{j_1} x^{(1)} \dots \partial^{j_d} x^{(d)}}(x_{\mathbf{i}_k}) \right| < \infty. \tag{21}$$

Set

$$\begin{aligned} c_{21} \quad = \quad & \max\left\{ \frac{1+\mathbf{E}\{Y^2\}}{c_3}, (N+1)^d \cdot \max\left\{ \left| \frac{1}{j_1! \cdots j_d!} \cdot \frac{\partial^{j_1+\dots+j_d} m}{\partial^{j_1} x^{(1)} \dots \partial^{j_d} x^{(d)}}(x_{\mathbf{i}_k}) \right|^2 \quad : \right.\right. \\ & \left.\left. j_1, \dots, j_d \in \{0, \dots, q\},\, j_1 + \dots + j_d \le q \right\}\right\} \end{aligned} \tag{22}$$

and let $A_n$ be the event that

$$\frac{1}{n} \sum_{i=1}^{n} Y_i^2 \le 1 + \mathbf{E}\{Y^2\} \tag{23}$$

holds. Then

$$\mathbf{P}(A_n^c) \le \frac{\mathbf{Var}\{Y^2\}}{n} \le \frac{c_{22}}{n}$$

by Chebychev inequality.

Set $\hat{m}_n = T_{\beta_n} \tilde{m}_n = m_n$ in case that $A_n$ holds and set $\hat{m}_n = T_{\beta_n} g_n$ otherwise. Then

$$\begin{aligned} & \mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \\ & \le 4\beta_n^2 \cdot \mathbf{P}\{A_n^c\} + \mathbf{E}\{\int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \cdot 1_{A_n}\} \\ & \le \frac{4 \cdot c_{22} \cdot \beta_n^2}{n} + \mathbf{E} \int |\hat{m}_n(x) - m(x)|^2 \mathbf{P}_X(dx). \end{aligned}$$

The definition of the estimate $\tilde{m}_n$ implies

$$\tilde{m}_n(x) = \sum_{j=1}^{J_n} \hat{a}_j \cdot f_j$$

for some $f_j \in \mathcal{F}$ and some $\hat{a}_j$ satisfying

$$\sum_{j=1}^{J_n} \hat{a}_j^2 \le \frac{1}{n} \sum_{i=1}^{n} Y_i^2 \cdot \frac{n}{c_3}.$$

Hence on $A_n$ we have

$$\sum_{j=1}^{J_n} \hat{a}_j^2 \leq \frac{1+\mathbf{E}Y^2}{c_3} \cdot n,$$

and consequently we can assume w.l.o.g. that $m_n$ satisfies $m_n = T_{\beta_n} \bar{m}_n$ for some $\bar{m}_n \in \mathcal{F}^{(J_n)}$. And since

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} |Y_i - \tilde{m}_n(X_i)|^2 \\
&\leq \frac{1}{n}\sum_{i=1}^{n} |Y_i - \tilde{m}_n(X_i)|^2 + \frac{c_3}{n} \cdot \sum_{k=1}^{(M_n+1)^d} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^2_{\mathbf{i}_k, j_1,\dots,j_d} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} |Y_i - g_n(X_i)|^2 \\
&\quad + \frac{c_3}{n} \cdot \sum_{k=1}^{(M_n+1)^d} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\} \\ j_1+\dots+j_d \leq q}} \left| \frac{1}{j_1! \cdots j_d!} \cdot \frac{\partial^{j_1+\dots+j_d} m}{\partial^{j_1} x^{(1)} \dots \partial^{j_d} x^{(d)}}(x_{\mathbf{i}_k}) \right|^2
\end{aligned}$$

(by definition of $\tilde{m}_n$) and (21), we also have

$$\frac{1}{n}\sum_{i=1}^{n} |Y_i - \bar{m}_n(X_i)|^2 \leq \frac{1}{n}\sum_{i=1}^{n} |Y_i - g_n(X_i)|^2 + c_{23} \cdot \frac{(M_n+1)^d}{n}.$$

Set

$$\begin{aligned}
P_n(x) \quad = \quad & \sum_{k=1}^{(M_n+1)^d} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,q\} \\ j_1+\dots+j_d \leq q}} \frac{1}{j_1! \cdots j_d!} \cdot \frac{\partial^{j_1+\dots+j_d} m}{\partial^{j_1} x^{(1)} \dots \partial^{j_d} x^{(d)}}(x_{\mathbf{i}_k}) \\
& \cdot (x^{(1)} - x^{(1)}_{\mathbf{i}_k})^{j_1} \cdots (x^{(d)} - x^{(d)}_{\mathbf{i}_k})^{j_d} \cdot \prod_{j=1}^{d} (1 - \frac{M_n}{2a} \cdot |x^{(j)} - x^{(j)}_{\mathbf{i}_k}|)_+ .
\end{aligned}$$

Application of Lemma 8 (with $|\Theta_n| = 1$ and $g_{n,1} = g_n$ deterministic) yields

$$\begin{aligned}
&\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \\
&\leq \frac{c_{23} \cdot (\log n)^2 \cdot \left( \log \left( \sup_{x_1^n \in supp(X)^n} \mathcal{N}_1 \left( \frac{1}{n \cdot \beta_n}, \mathcal{F}^{(J_n)}, x_1^n \right) \right) + 1 \right)}{n} \\
&\quad + 2 \cdot \int |g_n(x) - m(x)|^2 \mathbf{P}_X(dx) + 2 \cdot c_{21} \cdot \frac{(M_n+1)^d}{n}.
\end{aligned}$$

By Lemma 9 we know that

$$\frac{c_{23} \cdot \log(n)^2 \cdot \left( \log \left( \sup_{x_1^n \in supp(X)^n} \mathcal{N}_1 \left( \frac{1}{n \cdot \beta_n}, \mathcal{F}^{(J_n)}, x_1^n \right) \right) + 1 \right)}{n}$$

$$\leq c_{24} \cdot \frac{(\log n)^3 \cdot (N+1)^d \cdot (M_n+1)^d}{n}.$$

Furthermore we have

$$\int |g_n(x)-m(x)|^2 \mathbf{P}_X(dx) \leq 2 \cdot \sup_{x \in [-a_n,a_n]^d} |g_n(x)-P_n(x)|^2 + 2 \cdot \sup_{x \in [-a_n,a_n]^d} |P_n(x)-m(x)|^2.$$

By Lemma 5 we know

$$\begin{aligned} \sup_{x \in [-a_n,a_n]^d} |g_n(x)-P_n(x)| &\leq (M_n+1)^d \cdot (q+1)^d \cdot c_{25} \cdot a_n^{6(N+d)} \cdot M_n^3 \frac{1}{R_n} \\ &\leq (M_n+1)^d \cdot (q+1)^d \cdot c_{25} \cdot (\log n) \cdot \frac{M_n^3}{R_n}, \end{aligned}$$

and Lemma 5 in Schmidt–Hieber (2019) implies

$$\sup_{x \in [-a_n,a_n]^d} |P_n(x)-m(x)| \leq c_{26} \cdot \frac{a_n^p}{M_n^p} \leq c_{26} \cdot (\log n) \cdot \frac{1}{M_n^p}.$$

Plugging in the values for $R_n$ and $M_n$ we get the assertion. □

## 5.5. Proof of Theorem 2

W.l.o.g. we assume $supp(X) \subseteq [-A_n, A_n]^d$.

Define the estimate $\bar{m}_n$ exactly like $m_n$ except that for given directions $\mathbf{b}_l$ $(l = 1, \dots, r)$ we define the neural network estimate $\tilde{m}_n(x)$ by

$$\tilde{m}_n(x) = \sum_{l=1,\dots,r} \sum_{k=1}^{M_n+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a_{i_k,j_1,\dots,j_d,\mathbf{b}_l} \cdot f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l}(x),$$

where the coefficients $a_{k,j_1,\dots,j_d,\mathbf{b}_l}$ are chosen from the set

$$\left\{ (a_{k,j_1,\dots,j_d,\mathbf{b}_l})_{k,j_1,\dots,j_d,l} : \sum_{k,j_1,\dots,j_d,l} a_{k,j_1,\dots,j_d,\mathbf{b}_l}^2 \leq c_{27} \cdot n^2 \right\}$$

by minimizing

$$\frac{1}{n} \sum_{i=1}^n |Y_i - \tilde{m}_n(X_i)|^2 + \frac{c_3}{n} \cdot \sum_{l=1}^r \sum_{k=0}^K \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a_{k,j_1,\dots,j_d,\mathbf{b}_l}^2$$

for some constant $c_3 > 0$. Then $\bar{m}_n$ satisfies

$$\bar{m}_n \in \left\{ T_{\beta_n} f \,:\, f \in \mathcal{F}^{(J_n)} \right\},$$

where $\mathcal{F}^{(J_n)}$ (with $J_n = r \cdot (M_n + 1) \cdot \binom{N+d}{d}$) is the function space defined in Lemma 9. On the event

$$B_n = \{|Y_i| \leq \sqrt{n} \, : \, i = 1, \dots, n\}$$

we know by (17) that we have $m_n = \bar{m}_n$ (provided $c_{27} \geq 1/c_3$). Hence

$$\int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \leq \int |\bar{m}_n(x) - m(x)|^2 \mathbf{P}_X(dx) + 4\beta_n^2 \cdot 1_{B_n^c}.$$

By Markov inequality we know

$$\mathbf{P}\{B_n^c\} \leq n \cdot \mathbf{P}\{|Y| > \sqrt{n}\} \leq \frac{n \cdot \mathbf{E}\{e^{c_3 \cdot Y^2}\}}{\exp(c_3 \cdot n)},$$

therefore (14) implies that it suffices to show the assertion under the additional assumption

$$\tilde{m}_n(\cdot, (X_1, Y_1), \dots, (X_n, Y_n)) \in \mathcal{F}^{(J_n)}. \tag{24}$$

By Lemma 7 we know that for each $i \in \{1, \dots, I_n\}$ there exist coefficients $a^{(i)}_{k,j_1,\dots,j_d,l} \in [-c_{28} \cdot A_n^p, c_{28} \cdot A_n^p]$, which depend on $\mathbf{a}_l$ and on $\mathbf{b}_l^{(i)}$, but which are independent of $(X_1, Y_1)$, $\dots$, $(X_n, Y_n)$, such that we have for all $x \in [-A_n, A_n]^d$

$$\begin{aligned}
&\Bigg| \sum_{l=1}^{r} \sum_{j=0}^{q} \frac{g_l^{(j)}((\mathbf{b}_l^{(i)})^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l^{(i)})^T x)^j \\
&\qquad - \sum_{l=0}^{r} \sum_{k=1}^{M_n+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(i)}_{i_k,j_1,\dots,j_d,l} \cdot (x^{(1)})^{j_1} \cdots (x^{(d)})^{j_d} \\
&\qquad\qquad \cdot \left(1 - \frac{M_n}{2 \cdot \sqrt{d} \cdot A_n} \cdot |(\mathbf{b}_l^{(i)})^T x - u_{i_k}|\right)_+ \Bigg| \\
&\leq r \cdot 2^p \cdot (p+1) \cdot C \cdot A_n^{2p} \cdot \left( \max\left\{ \frac{1}{M_n}, \max_{l=1,\dots,r} \|\mathbf{a}_l - \mathbf{b}_l^{(i)}\|_\infty \right\} \right)^p .
\end{aligned} \tag{25}$$

From the definition of the estimate we get

$$\begin{aligned}
&\frac{1}{n} \sum_{i=1}^{n} |Y_i - \tilde{m}_n(X_i)|^2 \\
&\leq \min_{t=1,\dots,I_n} \Bigg\{ \frac{1}{n} \sum_{i=1}^{n} |Y_i - \sum_{l=1,\dots,r} \sum_{k=1}^{M_n+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(t)}_{i_k,j_1,\dots,j_d,l} \cdot f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l^{(t)}}(X_i)|^2 \\
&\qquad + \frac{c_3}{n} \cdot \sum_{l=1}^{r} \sum_{k=1}^{M_n+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} (a^{(t)}_{i_k,j_1,\dots,j_d,l})^2 \Bigg\}
\end{aligned}$$

$$\leq \min_{t=1,\dots,I_n} \Bigg\{ \frac{1}{n} \sum_{i=1}^{n} |Y_i - \sum_{l=1,\dots,r} \sum_{k=1}^{M_n+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(t)}_{i_k,j_1,\dots,j_d,l} \cdot f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l^{(t)}}(X_i)|^2$$

$$+c_{29} \cdot A_n^{2p} \cdot r \cdot \binom{N+d}{d} \cdot \frac{M_n}{n} \Bigg\}.$$

Hence, application of Lemma 8 and Lemma 9 (together with (24)) yields

$$\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx)$$

$$\leq c_{30} \cdot \frac{(\log n)^3 \cdot M_n}{n}$$

$$+2 \cdot \mathbf{E} \Bigg( \min_{t=1,\dots,I_n} \int \Big| \sum_{l=1,\dots,r} \sum_{k=1}^{M_n+1} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(t)}_{i_k,j_1,\dots,j_d,l} \cdot f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l^{(t)}}(x)$$

$$-m(x) \Big|^2 \mathbf{P}_X(dx) \Bigg) + c_{31} \cdot (\log n) \cdot n^{-\frac{2p}{2p+1}}.$$

Because of $(a+b+c)^2 \leq 3a^2 + 3b^2 + 3c^2$ $(a, b, c \in \mathbb{R})$ we have

$$\int | \sum_{l=1,\dots,r} \sum_{k=1}^{M_n} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(t)}_{i_k,j_1,\dots,j_d,l} \cdot f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l^{(t)}}(x) - m(x)|^2 \mathbf{P}_X(dx)$$

$$\leq 3 \cdot \int \Bigg| \sum_{l=1,\dots,r} \sum_{k=1}^{M_n} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(t)}_{i_k,j_1,\dots,j_d,l} \cdot f_{net,j_1,\dots,j_d,i_k,\mathbf{b}_l^{(t)}}(x)$$

$$-\sum_{l=1}^{r} \sum_{j=0}^{q} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(t)}_{i_k,j_1,\dots,j_d,l} \cdot (x^{(1)})^{j_1} \cdots (x^{(d)})^{j_d}$$

$$\cdot \left(1 - \frac{M_n}{2 \cdot \sqrt{d} \cdot A_n} \cdot |(\mathbf{b}_l^{(t)})^T x - u_{i_k}|\right)_+ \Bigg|^2 \mathbf{P}_X(dx)$$

$$+3 \cdot \int | \sum_{l=1}^{r} \sum_{j=0}^{q} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \leq N}} a^{(t)}_{i_k,j_1,\dots,j_d,l} \cdot (x^{(1)})^{j_1} \cdots (x^{(d)})^{j_d}$$

$$\cdot \left(1 - \frac{M_n}{2 \cdot \sqrt{d} \cdot A_n} \cdot |(\mathbf{b}_l^{(t)})^T x - u_{i_k}|\right)_+$$

$$-\sum_{l=1}^{r} \sum_{j=0}^{q} \frac{g_l^{(j)}((\mathbf{b}_l^{(t)})^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l^{(t)})^T x)^j |^2 \mathbf{P}_X(dx)$$

$$+3 \cdot \int |\sum_{l=1}^{r} \sum_{j=0}^{q} \frac{g_l^{(j)}(\mathbf{b}_l^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l^{(t)})^T x)^j - m(x)|^2 \mathbf{P}_X(dx).$$

Application of Lemma 5 implies for all $x \in [-A_n, A_n]^d$

$$\Bigg| \sum_{l=1,\dots,r} \sum_{k=0}^{M_n} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \le N}} a_{k,j_1,\dots,j_d,l}^{(t)} \cdot f_{net,k,j_1,\dots,j_d,\mathbf{b}_l^{(t)}}(x)$$

$$- \sum_{l=1}^{r} \sum_{j=0}^{q} \sum_{\substack{j_1,\dots,j_d \in \{0,\dots,N\} \\ j_1+\dots+j_d \le N}} a_{k,j_1,\dots,j_d,l}^{(t)} \cdot (x^{(1)})^{j_1} \cdots (x^{(d)})^{j_d}$$

$$\cdot \left(1 - \frac{K}{2 \cdot \sqrt{d} \cdot A} \cdot |(\mathbf{b}_l^{(t)})^T x - u_k|\right)_+ \Bigg|^2$$

$$\le r^2 \cdot (M_n + 1)^2 \cdot (N+d)^{2d} \cdot c_{28}^2 \cdot A_n^{2p} \cdot c_{12}^2 \cdot 3^{6 \cdot 3^s} \cdot A_n^{6 \cdot 2^s} \cdot M_n^6 \cdot \frac{1}{R_n^2} \le c_{32} \cdot \frac{(\log n)^2}{n}.$$

By Lemma 6 we have for all $x \in [-A_n, A_n]^d$

$$|\sum_{l=1}^{r} \sum_{j=0}^{q} \frac{g_l^{(j)}((\mathbf{b}_l^{(t)})^T x)}{j!} \cdot ((\mathbf{a}_l - \mathbf{b}_l)^T x)^j - m(x)|^2 \le c_{33} \cdot A_n^{2p} \cdot \|\mathbf{a}_l - \mathbf{b}_l^{(t)}\|_\infty^{2p}.$$

Using this together with (25) we see that it remains to show

$$\mathbf{E}\left\{ \min_{i=1,\dots,I_n} \max_{s=1,\dots,r} \|\mathbf{b}_s^{(i)} - \mathbf{a}_s\|_\infty^{2p} \right\} \le c_{34} \cdot (\log n)^2 \cdot n^{-\frac{2p}{2p+1}}.$$

By the random choice of the $\mathbf{b}_l^{(i)}$ we know for any $t \in (0,1]$

$$\begin{aligned} \mathbf{P}\left\{ \min_{i=1,\dots,I_n} \max_{l=1,\dots,r} \|\mathbf{b}_l^{(i)} - \mathbf{a}_l\|_\infty > t \right\} &= \prod_{i=1}^{I_n} (1 - \mathbf{P}\left\{ \max_{l=1,\dots,r} \|\mathbf{b}_l^{(i)} - \mathbf{a}_l\|_\infty \le t \right\}) \\ &\le \left(1 - \left(\frac{t}{2}\right)^{r \cdot d}\right)^{I_n} \end{aligned}$$

from which we conclude

$$\begin{aligned} &\mathbf{E}\left\{ \min_{i=1,\dots,I_n} \max_{l=1,\dots,r} \|\mathbf{b}_l^{(i)} - \mathbf{a}_l\|_\infty^{2p} \right\} \\ &\le \left(\frac{\log n}{n}\right)^{\frac{2p}{2p+1}} + 2^{2p} \cdot \mathbf{P}\left\{ \min_{i=1,\dots,I_n} \max_{l=1,\dots,r} \|\mathbf{b}_l^{(i)} - \mathbf{a}_l\|_\infty > \left(\frac{\log n}{n}\right)^{\frac{1}{2p+1}} \right\} \\ &\le \left(\frac{\log n}{n}\right)^{\frac{2p}{2p+1}} + 2^{2p} \cdot \left(1 - \cdot \frac{1}{2^{r \cdot d}} \left(\frac{\log n}{n}\right)^{\frac{r \cdot d}{2p+1}}\right)^{I_n} \end{aligned}$$

$$
\begin{aligned}
&\leq c_{35} \cdot \exp\left(-I_n \cdot \frac{1}{2^{r \cdot d}} \cdot \left(\frac{\log n}{n}\right)^{\frac{r \cdot d}{2p+1}}\right) \\
&= c_{35} \cdot \exp\left(-\frac{c_9}{2^{r \cdot d}} \cdot (\log n)^2\right) \\
&\leq c_{35} \cdot \left(\frac{\log n}{n}\right)^{\frac{2p}{2p+1}}
\end{aligned}
$$

where the last inequality follows from

$$
I_n \geq c_9 \cdot (\log n)^2 \cdot \left(\frac{n}{\log n}\right)^{\frac{r \cdot d}{2p+1}}.
$$

Putting together the above results we get the assertion. □

## References

[1] Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep kearning via over-parameterization. *Proceedings of the 36th International Conference on Machine Learning (PMLR 2019)*, **97**, pp. 242-252. Long Beach, California.

[2] Anthony, M., and Bartlett, P. L. (1999). *Neural Networks and Learning: Theoretical Foundations.* Cambridge University Press, Cambridge, UK.

[3] Arora, S., Cohen, N., Golowich, N., and Hu, W. (2018). A convergence analysis of gradient descent for deep linear neural networks. *International Conference on Learning Representations (ICLR 2019).* New Orleans, Louisiana.

[4] Bagirov, A. M., Clausen, C., and Kohler, M. (2009). Estimation of a regression function by maxima of minima of linear functions. *IEEE Transactions on Information Theory*, **55**, pp. 833-845.

[5] Bauer, B., Heimrich, F., Kohler, M., and Krzyżak, A. (2019). On estimation of surrogate models for high-dimensional computer experiments. *Annals of the Institute of Statistical Mathematics* **71**, pp. 107–136.

[6] Bauer, B., and Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Annals of Statistics* **47**, pp. 2261-2285.

[7] Braun, A., Kohler, M., and Walk, H. (2019). On the rate of convergence of a neural network regression estimate learned by gradient descent. Submitted for publication.

[8] Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. (2015) The loss surface of multilayer networks. International Con- ference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. Proceeding of Machine Learning Research, volume 38, pp. 192-204.

[9] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, New York, US.

[10] Devroye, L., and Wagner, T. J. (1980). Distribution-free consistency results in nonparametric discrimination and regression function estimation. *Annals of Statistics*, **8**, pp. 231-239.

[11] Du, S., and Lee, J. (2018). On the power of over-parametrization in neural networks with quadratic activation. *Proceedings of the 35th International Conference on Machine Learning (PMLR 2018)*, **80**, pp. 1329-1338. Stockholm, Sweden.

[12] Du, S., Lee, J., Tian, Y., Póczos, B., and Singh, A. (2018). Gradient descent learns one-hidden-layer CNN: don't be afraid of spurious local minima. *Proceedings of the 35th International Conference on Machine Learning (PMLR 2018)*, **80**, pp. 1339-1348. Stockholm, Sweden.

[13] Friedman, J. H., and Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association*, **76**, pp. 817-823.

[14] Györfi, L., Kohler, M., Krzyżak, A., and Walk, H. (2002). *A Distribution–Free Theory of Nonparametric Regression.* Springer.

[15] Härdle, W. and Stoker, T. M. (1989). Investigating smooth multiple regression by the method of average derivatives. *Journal of the American Statistical Association*, **84**, pp. 986-995.

[16] Härdle, W., Hall, P. and Ichimura, H. (1993). Optimal smoothing in single-index models. *Annals of Statistics*, **21**, pp. 157-178.

[17] Haykin, S. O. (2008). *Neural Networks and Learning Machines.* 3rd ed. Prentice-Hall, New York, US.

[18] Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, California, US.

[19] Horowitz, J. L., and Mammen, E. (2007). Rate-optimal estimation for a general class of nonparametric regression models with unknown link functions. *Annals of Statistics*, **35**, pp. 2589-2619.

[20] Huber, P. J. (1985). Projection pursuit. *Annals of Statistics*, **13**, pp. 435-475.

[21] Imaizumi, M., and Fukamizu, K. (2019). Deep neural networks learn non-smooth functions effectively. *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019).* Naha, Okinawa, Japan.

[22] Kawaguchi, K. (2016). Deep learning without poor local minima. *30th Conference on Neural Information Processing Systems (NIPS 2016).* Barcelona, Spain.

[23] Kawaguchi, K, and Huang, J. (2019). Gradient descent finds global minima for generalizable deep neural networks of practical sizes. *arXiv: 1908.02419v1.*

[24] Kohler, M., and Krzyżak, A. (2017). Nonparametric regression based on hierarchical interaction models. *IEEE Transaction on Information Theory*, **63**, pp. 1620-1630.

[25] Kohler, M., Krzyżak, A., and Langer, S. (2019). Deep learning and MARS: a connection. Submitted for publication.

[26] Kohler, M., and Langer, S. (2019). On the rate of convergence of fully connected very deep neural network regression estimates. Submitted for publication.

[27] Kong, E. and Xia, Y. (2007). Variable selection for the single-index model *Biometrika*, **94**, pp. 217-229.

[28] Lepski, O and Serdyukova, O. (2014). Adaptive estimation under single-index constraint in a regression model. *Annals of Statistics*, **42**, pp. 1-28.

[29] Liang, S., Sun, R., Lee, J., and Srikant, R. (2018). Adding one neuron can eliminate all bad local minima. *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)*, pp. 4355 - 4365. Montréal, Canada.

[30] Ripley, B. D. (2008). *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge, UK.

[31] Scarselli, F., and Tsoi, A. C. (1998). Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results. *Neural Networks*, **11**, pp. 15-37.

[32] Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks*, **61**, pp. 85-117.

[33] Schmidt-Hieber, J. (2019). Nonparametric regression using deep neural networks with ReLU activation function. To appear in *Annals of Statistics* (with discussion). *arXiv:1708.06633v2.*

[34] Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, **10**, pp. 1040-1053.

[35] Stone, C. J. (1985). Additive regression and other nonparametric models. *Annals of Statistics*, **13**, pp. 689-705.

[36] Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation. *Annals of Statistics*, **22**, pp. 118-184.

[37] Yu, Y., and Ruppert, D. (2002). Penalized Spline Estimation for Partially Linear Single-Index Models. *Journal of the American Statistical Association*, **97**, pp. 1042-1054.

# A. Supplementary material

## A.1. Computation of the linear neural network estimate

The estimate in Subsection 2.2 is given by

$$\tilde{m}_n(x) = \sum_{j=1}^{J} a_j \cdot B_j(x) \tag{26}$$

where $\mathbf{a} = (a_j)_{j=1,\dots,J} \in \mathbb{R}^J$ minimizes

$$\begin{aligned} &\frac{1}{n}(\mathbf{Y} - \mathbf{B}a)^T(\mathbf{Y} - \mathbf{B}a) + \frac{c_3}{n} \cdot \mathbf{a}^T\mathbf{a} \\ &= \frac{1}{n}(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{B}a) + \mathbf{a}^T\left(\frac{1}{n}\mathbf{B}^T\mathbf{B} + \frac{c_3}{n} \cdot \mathbf{1}\right)\mathbf{a}. \end{aligned}$$

Since the matrix

$$\mathbf{A} = \frac{1}{n}\mathbf{B}^T\mathbf{B} + \frac{c_3}{n} \cdot \mathbf{1}$$

is positive definite, its inverse matrix $\mathbf{A}^{-1}$ exists and it is easy to see that we have

$$\begin{aligned} &\frac{1}{n} \cdot (\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{B}a) + \mathbf{a}^T\left(\frac{1}{n}\mathbf{B}^T\mathbf{B} + \frac{c_3}{n} \cdot \mathbf{1}\right)\mathbf{a} \\ &= (\mathbf{a} - \frac{1}{n} \cdot \mathbf{A}^{-1}\mathbf{B}^T\mathbf{Y})^T\mathbf{A}(\mathbf{a} - \frac{1}{n} \cdot \mathbf{A}^{-1}\mathbf{B}^T\mathbf{Y}) + \frac{1}{n}\mathbf{Y}^T\mathbf{Y} - \frac{1}{n^2} \cdot \mathbf{Y}^T\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\mathbf{Y}. \end{aligned}$$

The last expression is minimal for $\mathbf{a} = \frac{1}{n} \cdot \mathbf{A}^{-1}\mathbf{B}^T\mathbf{Y}$, which proves that the vector of coefficients of our estimate (26) is the unique solution of the linear equation system (13).

## A.2. Proof of Lemma 5

Define $g_1^{(0)}$ by backward recursion:

$$g_k^{(s)}(x) = x^{(l)} - y^{(l)}$$

for $j_1 + j_2 + \dots + j_{l-1} + 1 \leq k \leq j_1 + j_2 + \dots + j_l$ and $l = 1, \dots, d$,

$$g_{j_1+j_2+\dots+j_d+k}^{(s)}(x) = \left(1 - \frac{M}{2a} \cdot |x^{(k)} - y^{(k)}|\right)_+$$

for $k = 1, \dots, d$, and

$$g_k^{(s)}(x) = 1$$

for $k = j_1 + j_2 + \dots + j_d + d + 1, j_1 + j_2 + \dots + j_d + d + 2, \dots, 2^s$, and

$$g_k^{(l)}(x) = g_{2k-1}^{(l+1)}(x) \cdot g_{2k}^{(l+1)}(x)$$

for $k \in \{1, 2, \dots, 2^l\}$ and $l \in \{0, \dots, s-1\}$.

Then we have for any $l \in \{0, \dots, s\}$, $k \in \{1, \dots, 2^l\}$ and $x \in [-a, a]$

$$|g_k^{(l)}(x)| \leq (2a)^{2^{s-l}}.$$

By Lemma 2 the network $f_{mult}$ satisfies for any $l \in \{0, \dots, s\}$ and $x, y \in [-3^{3^{s-l}} \cdot a^{2^{s-l}}, 3^{3^{s-l}} \cdot a^{2^{s-l}}]$

$$|f_{mult}(x, y) - x \cdot y| \leq \frac{20 \cdot \|\sigma'''\|_\infty}{3 \cdot |\sigma''(t_\sigma)|} \cdot 3^{3 \cdot 3^{s-l}} \cdot a^{3 \cdot 2^{s-l}} \cdot \frac{1}{R}.$$

Furthermore we have by Lemma 1 and Lemma 4 for any $x \in [-3a, 3a]$

$$|f_{id}(x) - x| \leq \frac{9 \cdot \|\sigma''\|_\infty \cdot a^2}{2 \cdot |\sigma'(t_{\sigma,id})|} \cdot \frac{1}{R} \tag{27}$$

and for any $x \in [-a, a]^d$

$$\begin{aligned} &\left| f_{hat,y}(x) - \left(1 - \frac{M}{2a} \cdot |x - y|\right)_+ \right| \\ &\leq 1792 \cdot \frac{\max\{\|\sigma''\|_\infty, \|\sigma'''\|_\infty, 1\}}{\min\{2 \cdot |\sigma'(t_{\sigma.id})|, |\sigma''(t_\sigma)|, 1\}} \cdot M^3 \cdot \frac{1}{R}. \end{aligned} \tag{28}$$

From this and (20) we can recursively conclude

$$|f_k^{(l)}(x)| \leq 3^{3^{s-l}} \cdot a^{2^{s-l}}$$

for $k \in \{1, \dots, 2^l\}$ and $l \in \{0, \dots, s\}$.

In order to prove the assertion of Lemma 5 we show in the sequel

$$|f_k^{(l)}(x) - g_k^{(l)}(x)| \leq c_{36} \cdot 3^{3 \cdot 3^{s-l}} \cdot a^{3 \cdot 2^{s-l}} \cdot M^3 \cdot \frac{1}{R}$$

for $k \in \{1, \dots, 2^l\}$ and $l \in \{0, \dots, s\}$, where

$$c_{36} = \max\left\{ \frac{20 \cdot \|\sigma'''\|_\infty}{3 \cdot |\sigma''(t_\sigma)|}, \frac{9 \cdot \|\sigma''\|_\infty}{|\sigma'(t_{\sigma,id})|}, 1792 \cdot \frac{\max\{\|\sigma''\|_\infty, \|\sigma'''\|_\infty, 1\}}{\min\{2 \cdot |\sigma'(t_{\sigma.id})|, |\sigma''(t_\sigma)|, 1\}} \right\}$$

For $s = l$ this is a consequence of (27), and (28). For $l \in \{0, 1, \dots, s-1\}$ we can conclude via induction

$$\begin{aligned} &|f_k^{(l)}(x) - g_k^{(l)}(x)| \\ &\leq |f_{mult}(f_{2k-1}^{(l+1)}(x), f_{2k}^{(l+1)}(x)) - f_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x)| \\ &\quad + |f_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x) - g_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x)| \\ &\quad + |g_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x) - g_{2k-1}^{(l+1)}(x) \cdot g_{2k}^{(l+1)}(x)| \\ &\leq c_{36} \cdot 3^{3 \cdot 3^{s-l-1}} \cdot a^{3 \cdot 2^{s-l-1}} \cdot \frac{1}{R} + 3^{3^{s-l-1}} \cdot a^{2^{s-l-1}} \cdot 2 \cdot c_{36} \cdot 3^{3 \cdot 3^{s-l-1}} \cdot a^{3 \cdot 2^{s-l-1}} \cdot M^3 \cdot \frac{1}{R} \\ &\leq c_{36} \cdot \left(3^{3^{s-l}} + 2 \cdot 3^{4 \cdot 3^{s-l-1}}\right) \cdot a^{3 \cdot 2^{s-l}} \cdot M^3 \cdot \frac{1}{R} \\ &\leq c_{36} \cdot 3^{3 \cdot 3^{s-l}} \cdot a^{3 \cdot 2^{s-l}} \cdot M^3 \cdot \frac{1}{R}. \end{aligned}$$

□

## A.3. Lemma 5 in the context of projection pursuit

**Lemma 10** *Let $M \in \mathbb{N}$ and let $\sigma : \mathbb{R} \to [0,1]$ be 2-admissible according to Definition 2. Let $A \geq 1$, $\mathbf{b} \in \mathbb{R}^d$ with $\|\mathbf{b}\| \leq 1$ and*

$$R \geq \max\left\{ \frac{\|\sigma''\|_\infty \cdot (M+1)}{2 \cdot |\sigma'(t_{\sigma,id})|}, \frac{9 \cdot \|\sigma''\|_\infty \cdot A}{|\sigma'(t_{\sigma,id})|}, \right. \tag{29}$$

$$\left. \frac{20 \cdot \|\sigma'''\|_\infty}{3 \cdot |\sigma''(t_\sigma)|} \cdot 3^{3 \cdot 3^s} \cdot A^{3 \cdot 2^s}, 1792 \cdot \frac{\max\{\|\sigma''\|_\infty, \|\sigma'''\|_\infty, 1\}}{\min\{2 \cdot |\sigma'(t_{\sigma,id})|, |\sigma''(t_\sigma)|, 1\}} \cdot d^{3/2} \cdot M^3 \right\}$$

*and let $y \in [-A, A]$. Let $N \in \mathbb{N}$ and let $j_1, \dots, j_d \in \mathbb{N}_0$ such that $j_1 + \dots + j_d \leq N$, and set $s = \lceil \log_2(N+1) \rceil$. Let $f_{id}$, $f_{mult}$ and $\bar{f}_{hat,z}$ (for $z \in \mathbb{R}$) be the neural networks defined in Subsection 3.2. (So in particular $\bar{f}_{hat,z}$ is the neural network from Lemma 4 with $y = z$ and $a = \sqrt{d} \cdot A$.) Define the network $f_{net,j_1,\dots,j_d,y}$ by*

$$f_{net,j_1,\dots,j_d,y}(x) = f_1^{(0)}(x),$$

*where $f_1^{(0)}$ is defined by backward recursion as follows:*

$$f_k^{(l)}(x) = f_{mult}\left(f_{2k-1}^{(l+1)}(x), f_{2k}^{(l+1)}(x)\right)$$

*for $k \in \{1, 2, \dots, 2^l\}$ and $l \in \{0, \dots, s-1\}$, and*

$$f_k^{(s)}(x) = f_{id}(f_{id}(x^{(l)}))$$

*for $j_1 + j_2 + \dots + j_{l-1} + 1 \leq k \leq j_1 + j_2 + \dots + j_l$ and $l = 1, \dots, d$,*

$$f_{j_1+j_2+\dots+j_d+1}^{(s)}(x) = \bar{f}_{hat,y}(\mathbf{b}^T x),$$

*and*

$$f_k^{(s)}(x) = 1$$

*for $k = j_1 + j_2 + \dots + j_d + 2, j_1 + j_2 + \dots + j_d + 3, \dots, 2^s$. Then we have for any $x \in [-A, A]^d$:*

$$\left| f_{net,y}(x) - (x^{(1)})^{j_1} \cdots (x^{(d)})^{j_d} \cdot (1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}^T x - y|)_+ \right|$$
$$\leq c_{37} \cdot 3^{3 \cdot 3^s} \cdot A^{3 \cdot 2^s} \cdot M^3 \cdot \frac{1}{R}.$$

**Proof.** Define $g_1^{(0)}$ by backward recursion:

$$g_k^{(s)}(x) = x^{(l)}$$

for $j_1 + j_2 + \dots + j_{l-1} + 1 \leq k \leq j_1 + j_2 + \dots + j_l$ and $l = 1, \dots, d$,

$$g_{j_1+j_2+\dots+j_d+1}^{(s)}(x) = \left(1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}^T x - y|\right)_+,$$

and

$$g_k^{(s)}(x) = 1$$

for $k = j_1 + j_2 + \dots + j_d + 2, j_1 + j_2 + \dots + j_d + 3, \dots, 2^s$, and

$$g_k^{(l)}(x) = g_{2k-1}^{(l+1)}(x) \cdot g_{2k}^{(l+1)}(x)$$

for $k \in \{1, 2, \dots, 2^l\}$ and $l \in \{0, \dots, s-1\}$.

Then we have for any $x \in [-A, A]^d$

$$|g_k^{(l)}(x)| \leq A^{2^{s-l}}.$$

By Lemma 2 the network $f_{mult}$ satisfies for any $l \in \{0, \dots, s\}$ and $x, y \in [-3^{3^{s-l}} \cdot A^{2^{s-l}}, 3^{3^{s-l}} \cdot A^{2^{s-l}}]$

$$|f_{mult}(x, y) - x \cdot y| \leq \frac{20 \cdot \|\sigma'''\|_\infty}{3 \cdot |\sigma''(t_\sigma)|} \cdot 3^{3 \cdot 3^{s-l}} \cdot A^{3 \cdot 2^{s-l}} \cdot \frac{1}{R}.$$

Furthermore we have by Lemma 1 and Lemma 4 for any $x \in [-3A, 3A]$

$$|f_{id}(x) - x| \leq \frac{9 \cdot \|\sigma''\|_\infty \cdot A^2}{2 \cdot |\sigma'(t_{\sigma,id})|} \cdot \frac{1}{R} \tag{30}$$

and for any $x \in [-A, A]^d$

$$\begin{aligned} &\left| \bar{f}_{hat,y}(x) - \left(1 - \frac{M}{2 \cdot \sqrt{d} \cdot A} \cdot |\mathbf{b}^T x - y|\right)_+ \right| \\ &\leq 1792 \cdot \frac{\max\{\|\sigma''\|_\infty, \|\sigma'''\|_\infty, 1\}}{\min\{2 \cdot |\sigma'(t_{\sigma.id})|, |\sigma''(t_\sigma)|, 1\}} \cdot M^3 \cdot \frac{1}{R}. \end{aligned} \tag{31}$$

From this and (29) we can recursively conclude

$$|f_k^{(l)}(x)| \leq 3^{3^{s-l}} \cdot A^{2^{s-l}}$$

for $k \in \{1, \dots, 2^l\}$ and $l \in \{0, \dots, s\}$.

In order to prove the assertion of Lemma 5 we show in the sequel

$$|f_k^{(l)}(x) - g_k^{(l)}(x)| \leq c_{37} \cdot 3^{3 \cdot 3^{s-l}} \cdot A^{3 \cdot 2^{s-l}} \cdot M^3 \cdot \frac{1}{R}$$

for $k \in \{1, \dots, 2^l\}$ and $l \in \{0, \dots, s\}$, where

$$c_{37} = \max\left\{ \frac{20 \cdot \|\sigma'''\|_\infty}{3 \cdot |\sigma''(t_\sigma)|}, \frac{9 \cdot \|\sigma''\|_\infty}{|\sigma'(t_{\sigma,id})|}, 1792 \cdot \frac{\max\{\|\sigma''\|_\infty, \|\sigma'''\|_\infty, 1\}}{\min\{2 \cdot |\sigma'(t_{\sigma.id})|, |\sigma''(t_\sigma)|, 1\}} \right\}.$$

For $s = l$ this is a consequence of (30) and (31). For $l \in \{0, 1, \dots, s-1\}$ we can conclude via induction

$$|f_k^{(l)}(x) - g_k^{(l)}(x)|$$

$$
\begin{aligned}
&\leq |f_{mult}(f_{2k-1}^{(l+1)}(x), f_{2k}^{(l+1)}(x)) - f_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x)| \\
&\quad +|f_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x) - g_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x)| \\
&\quad +|g_{2k-1}^{(l+1)}(x) \cdot f_{2k}^{(l+1)}(x) - g_{2k-1}^{(l+1)}(x) \cdot g_{2k}^{(l+1)}(x)| \\
&\leq c_{37} \cdot 3^{3 \cdot 3^{s-l-1}} \cdot A^{3 \cdot 2^{s-l-1}} \cdot \frac{1}{R} + 3^{3^{s-l-1}} \cdot A^{2^{s-l-1}} \cdot 2 \cdot c_{37} \cdot 3^{3 \cdot 3^{s-l-1}} \cdot A^{3 \cdot 2^{s-l-1}} \cdot M^3 \cdot \frac{1}{R} \\
&\leq c_{37} \cdot \left(3^{3^{s-l}} + 2 \cdot 3^{4 \cdot 3^{s-l-1}}\right) \cdot A^{3 \cdot 2^{s-l}} \cdot M^3 \cdot \frac{1}{R} \\
&\leq c_{37} \cdot 3^{3 \cdot 3^{s-l}} \cdot A^{3 \cdot 2^{s-l}} \cdot M^3 \cdot \frac{1}{R}.
\end{aligned}
$$

□

## A.4. Proof of Lemma 8

In the proof we use the following error decomposition:

$$
\begin{aligned}
&\int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \\
&\quad = \Big[\mathbf{E}\Big\{|m_n(X) - Y|^2 | \mathcal{D}_n\Big\} - \mathbf{E}\Big\{|m(X) - Y|^2\Big\} \\
&\qquad\quad - \Big(\mathbf{E}\Big\{|m_n(X) - T_{\beta_n}Y|^2 | \mathcal{D}_n\Big\} - \mathbf{E}\Big\{|m_{\beta_n}(X) - T_{\beta_n}Y|^2\Big\}\Big)\Big] \\
&\qquad + \Bigg[\mathbf{E}\Big\{|m_n(X) - T_{\beta_n}Y|^2 | \mathcal{D}_n\Big\} - \mathbf{E}\Big\{|m_{\beta_n}(X) - T_{\beta_n}Y|^2\Big\} \\
&\qquad\quad -2 \cdot \frac{1}{n} \sum_{i=1}^{n} \Big(|m_n(X_i) - T_{\beta_n}Y_i|^2 - |m_{\beta_n}(X_i) - T_{\beta_n}Y_i|^2\Big)\Bigg] \\
&\qquad + \Bigg[2 \cdot \frac{1}{n} \sum_{i=1}^{n} |m_n(X_i) - T_{\beta_n}Y_i|^2 - 2 \cdot \frac{1}{n} \sum_{i=1}^{n} |m_{\beta_n}(X_i) - T_{\beta_n}Y_i|^2 \\
&\qquad\quad - \left(2 \cdot \frac{1}{n} \sum_{i=1}^{n} |m_n(X_i) - Y_i|^2 - 2 \cdot \frac{1}{n} \sum_{i=1}^{n} |m(X_i) - Y_i|^2\right)\Bigg] \\
&\qquad + \left[2\left(\frac{1}{n} \sum_{i=1}^{n} |m_n(X_i) - Y_i|^2 - \frac{1}{n} \sum_{i=1}^{n} |m(X_i) - Y_i|^2\right)\right] \\
&\quad = \sum_{i=1}^{4} T_{i,n},
\end{aligned}
$$

where $T_{\beta_n}Y$ is the truncated version of $Y$ and $m_{\beta_n}$ is the regression function of $T_{\beta_n}Y$, i.e.,

$$
m_{\beta_n}(x) = \mathbf{E}\Big\{T_{\beta_n}Y | X = x\Big\}.
$$

We start with bounding $T_{1,n}$. By using $a^2 - b^2 = (a-b)(a+b)$ we get

$$
T_{1,n} \;=\; \mathbf{E}\Big\{|m_n(X) - Y|^2 - |m_n(X) - T_{\beta_n}Y|^2 \Big| \mathcal{D}_n\Big\}
$$

$$
\begin{aligned}
&-\mathbf{E}\Big\{|m(X)-Y|^2-|m_{\beta_n}(X)-T_{\beta_n}Y|^2\Big\}\\
=\;&\mathbf{E}\Big\{(T_{\beta_n}Y-Y)(2m_n(X)-Y-T_{\beta_n}Y)\Big|\mathcal{D}_n\Big\}\\
&-\mathbf{E}\Big\{\Big((m(X)-m_{\beta_n}(X))+(T_{\beta_n}Y-Y)\Big)\Big(m(X)+m_{\beta_n}(X)-Y-T_{\beta_n}Y\Big)\Big\}\\
=\;&T_{5,n}+T_{6,n}.
\end{aligned}
$$

With the Cauchy-Schwarz inequality and

$$
I_{\{|Y|>\beta_n\}}\leq\frac{\exp(c_4/2\cdot|Y|^2)}{\exp(c_4/2\cdot\beta_n^2)} \tag{32}
$$

we conclude

$$
\begin{aligned}
|T_{5,n}| &\leq \sqrt{\mathbf{E}\big\{|T_{\beta_n}Y-Y|^2\big\}}\cdot\sqrt{\mathbf{E}\big\{|2m_n(X)-Y-T_{\beta_n}Y|^2\big|\mathcal{D}_n\big\}}\\
&\leq \sqrt{\mathbf{E}\big\{|Y|^2\cdot I_{\{|Y|>\beta_n\}}\big\}}\cdot\sqrt{\mathbf{E}\big\{2\cdot|2m_n(X)-T_{\beta_n}Y|^2+2\cdot|Y|^2\big|\mathcal{D}_n\big\}}\\
&\leq \sqrt{\mathbf{E}\bigg\{|Y|^2\cdot\frac{\exp(c_4/2\cdot|Y|^2)}{\exp(c_4/2\cdot\beta_n^2)}\bigg\}}\\
&\qquad\cdot\sqrt{\mathbf{E}\big\{2\cdot|2m_n(X)-T_{\beta_n}Y|^2\big|\mathcal{D}_n\big\}+2\mathbf{E}\big\{|Y|^2\big\}}\\
&\leq \sqrt{\mathbf{E}\Big\{|Y|^2\cdot\exp(c_4/2\cdot|Y|^2)\Big\}}\cdot\exp\left(-\frac{c_4\cdot\beta_n^2}{4}\right)\cdot\sqrt{2(3\beta_n)^2+2\mathbf{E}\big\{|Y|^2\big\}}.
\end{aligned}
$$

With $x\leq\exp(x)$ for $x\in\mathbb{R}$ we get

$$
|Y|^2\leq\frac{2}{c_4}\cdot\exp\Big(\frac{c_4}{2}\cdot|Y|^2\Big)
$$

and hence $\mathbf{E}\Big\{|Y|^2\cdot\exp(c_4/2\cdot|Y|^2)\Big\}$ is bounded by

$$
\mathbf{E}\left(\frac{2}{c_4}\cdot\exp\big(c_4/2\cdot|Y|^2\big)\cdot\exp(c_4/2\cdot|Y|^2)\right)\leq\mathbf{E}\left(\frac{2}{c_4}\cdot\exp\big(c_4\cdot|Y|^2\big)\right)\leq c_{38}
$$

which is less than infinity by the assumptions of the lemma. Furthermore the third term is bounded by $\sqrt{18\beta_n^2+c_{39}}$ because

$$
\mathbf{E}(|Y|^2)\leq\mathbf{E}(1/c_4\cdot\exp(c_4\cdot|Y|^2)\leq c_{39}<\infty, \tag{33}
$$

which follows again as above. With the setting $\beta_n=c_6\cdot\log(n)$ it follows for some constants $c_{40},c_{41}>0$ that

$$
|T_{5,n}|\;\leq\;\sqrt{c_{38}}\cdot\exp\big(-c_{40}\cdot\log(n)^2\big)\cdot\sqrt{(18\cdot c_6^2\cdot(\log n)^2+c_{39})}\leq c_{41}\cdot\frac{\log(n)}{n}.
$$

By the Cauchy-Schwarz inequality we get

$$\begin{aligned} T_{6,n} \quad \leq \quad & \sqrt{2 \cdot \mathbf{E}\Big\{ |(m(X) - m_{\beta_n}(X))|^2 \Big\} + 2 \cdot \mathbf{E}\Big\{ |(T_{\beta_n}Y - Y)|^2 \Big\}} \\ & \cdot \sqrt{\mathbf{E}\Big\{ \Big| m(X) + m_{\beta_n}(X) - Y - T_{\beta_n}Y \Big|^2 \Big\}}, \end{aligned}$$

where we can bound the second factor on the right-hand side in the above inequality in the same way we have bounded the second factor in $T_{5,n}$, because by assumption $||m||_\infty$ is bounded and furthermore $m_{\beta_n}$ is bounded by $\beta_n$. Thus we get for some constant $c_{42} > 0$

$$\sqrt{\mathbf{E}\Big\{ \Big| m(X) + m_{\beta_n}(X) - Y - T_{\beta_n}Y \Big|^2 \Big\}} \leq c_{42} \cdot \log(n).$$

Next we consider the first term. By Jensen's inequality it follows that

$$\mathbf{E}\Big\{ |m(X) - m_{\beta_n}(X)|^2 \Big\} \quad \leq \quad \mathbf{E}\left\{ \mathbf{E}\Big( |Y - T_{\beta_n}Y|^2 \Big| X \Big) \right\} = \mathbf{E}\Big\{ |Y - T_{\beta_n}Y|^2 \Big\}.$$

Hence we get

$$T_{6,n} \quad \leq \quad \sqrt{4 \cdot \mathbf{E}\left\{ |Y - T_{\beta_n}Y|^2 \right\}} \cdot c_{42} \cdot \log(n)$$

and therefore with the calculations from $T_{5,n}$ it follows that $T_{6,n} \leq c_{43} \cdot \log(n)/n$ for some constant $c_{43} > 0$. Altogether we get

$$T_{1,n} \leq c_{44} \cdot \frac{\log(n)}{n}$$

for some constant $c_{44} > 0$.
Next we consider $T_{2,n}$ and conclude for $t > 0$

$$\begin{aligned} \mathbf{P}\{T_{2,n} > t\} \leq \mathbf{P} \Bigg\{ & \exists f \in T_{\beta_n, supp(X)} \mathcal{F}_n : \mathbf{E}\left( \left| \frac{f(X)}{\beta_n} - \frac{T_{\beta_n}Y}{\beta_n} \right|^2 \right) - \mathbf{E}\left( \left| \frac{m_{\beta_n}(X)}{\beta_n} - \frac{T_{\beta_n}Y}{\beta_n} \right|^2 \right) \\ & - \frac{1}{n} \sum_{i=1}^{n} \left( \left| \frac{f(X_i)}{\beta_n} - \frac{T_{\beta_n}Y_i}{\beta_n} \right|^2 - \left| \frac{m_{\beta_n}(X_i)}{\beta_n} - \frac{T_{\beta_n}Y_i}{\beta_n} \right|^2 \right) \\ & > \frac{1}{2} \left( \frac{t}{\beta_n^2} + \mathbf{E}\left( \left| \frac{f(X)}{\beta_n} - \frac{T_{\beta_n}Y}{\beta_n} \right|^2 \right) - \mathbf{E}\left( \left| \frac{m_{\beta_n}(X)}{\beta_n} - \frac{T_{\beta_n}Y}{\beta_n} \right|^2 \right) \right) \Bigg\}, \end{aligned}$$

where $T_{\beta_n, supp(X)} \mathcal{F}_n$ is defined as $\left\{ T_{\beta_n} f \cdot 1_{supp(X)} : f \in \mathcal{F}_n \right\}$. Theorem 11.4 in Györfi et al. (2002) and the relation

$$\mathcal{N}_1 \left( \delta, \left\{ \frac{1}{\beta_n} g : g \in \mathcal{G} \right\}, x_1^n \right) \leq \mathcal{N}_1 \left( \delta \cdot \beta_n, \mathcal{G}, x_1^n \right)$$

for an arbitrary function space $\mathcal{G}$ and $\delta > 0$ lead to

$$\mathbf{P}\{T_{2,n} > t\} \leq 14 \cdot \sup_{x_1^n \in supp(X)^n} \mathcal{N}_1\left(\frac{t}{80 \cdot \beta_n}, \mathcal{F}_n, x_1^n\right) \cdot \exp\left(-\frac{n}{5136 \cdot \beta_n^2} \cdot t\right).$$

Since the covering number is decreasing in $t$, we can conclude for $\varepsilon_n \geq \frac{80}{n}$

$$\begin{aligned}\mathbf{E}(T_{2,n}) &\leq \varepsilon_n + \int_{\varepsilon_n}^{\infty} \mathbf{P}\{T_{2,n} > t\} dt \\ &\leq \varepsilon_n + 14 \cdot \sup_{x_1^n \in supp(X)^n} \mathcal{N}_1\left(\frac{1}{n \cdot \beta_n}, \mathcal{F}_n, x_1^n\right) \cdot \exp\left(-\frac{n}{5136 \cdot \beta_n^2} \cdot \varepsilon_n\right) \cdot \frac{5136 \cdot \beta_n^2}{n}.\end{aligned}$$

Choosing

$$\varepsilon_n = \frac{5136 \cdot \beta_n^2}{n} \cdot \log\left(14 \cdot \sup_{x_1^n \in supp(X)^n} \mathcal{N}_1\left(\frac{1}{n \cdot \beta_n}, \mathcal{F}_n, x_1^n\right)\right)$$

(which satisfies the necessary condition $\varepsilon_n \geq \frac{80}{n}$ if the constant $c_6$ in the definition of $\beta_n$ is not too small) minimizes the right-hand side and implies

$$\mathbf{E}(T_{2,n}) \leq \frac{c_{45} \cdot \log(n)^2 \cdot \log\left(\sup_{x_1^n \in supp(X)^n} \mathcal{N}_1\left(\frac{1}{n \cdot \beta_n}, \mathcal{F}_n, x_1^n\right)\right)}{n}.$$

By bounding $T_{3,n}$ similarly to $T_{1,n}$ we get

$$\mathbf{E}(T_{3,n}) \quad \leq \quad c_{46} \cdot \frac{\log(n)}{n}$$

for some large enough constant $c_{46} > 0$ and hence we get in total

$$\mathbf{E}\left(\sum_{i=1}^{3} T_{i,n}\right) \quad \leq \quad \frac{c_{47} \cdot \log(n)^2 \cdot \left(\log\left(\sup_{x_1^n \in supp(X)^n} \mathcal{N}_1\left(\frac{1}{n \cdot \beta_n}, \mathcal{F}_n, x_1^n\right)\right) + 1\right)}{n}$$

for some sufficient large constant $c_{47} > 0$.

We finish the proof by bounding $T_{4,n}$. Let $A_n$ be the event, that there exists $i \in \{1, ..., n\}$ such that $|Y_i| > \beta_n$ and let $I_{A_n}$ be the indicator function of $A_n$. Then we get

$$\begin{aligned}\mathbf{E}(T_{4,n}) \quad &\leq \quad 2 \cdot \mathbf{E}\left(\frac{1}{n}\sum_{i=1}^{n} |m_n(X_i) - Y_i|^2 \cdot I_{A_n}\right) \\ &\quad\quad +2 \cdot \mathbf{E}\left(\frac{1}{n}\sum_{i=1}^{n} |m_n(X_i) - Y_i|^2 \cdot I_{A_n^c} - \frac{1}{n}\sum_{i=1}^{n} |m(X_i) - Y_i|^2\right) \\ &= \quad 2 \cdot \mathbf{E}\left(|m_n(X_1) - Y_1|^2 \cdot I_{A_n}\right) \\ &\quad\quad +2 \cdot \mathbf{E}\left(\frac{1}{n}\sum_{i=1}^{n} |m_n(X_i) - Y_i|^2 \cdot I_{A_n^c} - \frac{1}{n}\sum_{i=1}^{n} |m(X_i) - Y_i|^2\right)\end{aligned}$$

$$= \quad T_{7,n} + T_{8,n}.$$

By the Cauchy-Schwarz inequality we get for $T_{7,n}$

$$\begin{aligned}
\frac{1}{2} \cdot T_{7,n} &\leq \sqrt{\mathbf{E}\left( (|m_n(X_1) - Y_1|^2)^2 \right)} \cdot \sqrt{\mathbf{P}(A_n)} \\
&\leq \sqrt{\mathbf{E}\left( (2|m_n(X_1)|^2 + 2|Y_1|^2)^2 \right)} \cdot \sqrt{n \cdot \mathbf{P}\{|Y_1| > \beta_n\}} \\
&\leq \sqrt{\mathbf{E}\left(8|m_n(X_1)|^4 + 8|Y_1|^4\right)} \cdot \sqrt{n \cdot \frac{\mathbf{E}\left(\exp(c_4 \cdot |Y_1|^2)\right)}{\exp(c_4 \cdot \beta_n^2)}},
\end{aligned}$$

where the last inequality follows as in the proof of inequality (32). Using $x \leq \exp(x)$ for $x \in \mathbb{R}$ we get

$$\begin{aligned}
\mathbf{E}\left(|Y|^4\right) &= \mathbf{E}\left(|Y|^2 \cdot |Y|^2\right) \leq \mathbf{E}\left( \frac{2}{c_4} \cdot \exp\left(\frac{c_4}{2} \cdot |Y|^2\right) \cdot \frac{2}{c_4} \cdot \exp\left(\frac{c_4}{2} \cdot |Y|^2\right)\right) \\
&= \frac{4}{c_4^2} \cdot \mathbf{E}\left(\exp\left(c_4 \cdot |Y|^2\right)\right),
\end{aligned}$$

which is finite by assumption (14) of the lemma. Furthermore $||m_n||_\infty$ is bounded by $\beta_n$ and therefore the first factor is bounded by

$$c_{48} \cdot \beta_n^2 = c_{49} \cdot (\log n)^2$$

for some constant $c_{49} > 0$. The second factor is bounded by $1/n$, because by the assumptions of the lemma $\mathbf{E}\left(\exp\left(c_4 \cdot |Y_1|^2\right)\right)$ is bounded by some constant $c_{50} < \infty$ and hence we get

$$\sqrt{n \cdot \frac{\mathbf{E}\left(\exp(c_4 \cdot |Y_1|^2)\right)}{\exp(c_4 \cdot \beta_n^2)}} \quad \leq \quad \sqrt{n} \cdot \frac{\sqrt{c_{49}}}{\sqrt{\exp(c_4 \cdot \beta_n^2)}} \leq \frac{\sqrt{n} \cdot \sqrt{c_{50}}}{\exp((c_4 \cdot c_6^2 \cdot (\log n)^2)/2)}.$$

Since $\exp(-c \cdot \log(n)^2) = O(n^{-2})$ for any $c > 0$, we get altogether

$$T_{7,n} \quad \leq \quad c_{51} \cdot \frac{(\log n)^2 \sqrt{n}}{n^2} \leq c_{52} \cdot \frac{(\log n)^2}{n}.$$

With the definition of $A_n^c$ and $\tilde{m}_n$ defined as in the assumptions of this lemma we conclude

$$\begin{aligned}
T_{8,n} &\leq 2 \cdot \mathbf{E}\left( \frac{1}{n} \sum_{i=1}^n |\tilde{m}_n(X_i) - Y_i|^2 \cdot I_{A_n^c} - \frac{1}{n} \sum_{i=1}^n |m(X_i) - Y_i|^2 \right) \\
&\leq 2 \cdot \mathbf{E}\left( \frac{1}{n} \sum_{i=1}^n |\tilde{m}_n(X_i) - Y_i|^2 - \frac{1}{n} \sum_{i=1}^n |m(X_i) - Y_i|^2 \right) \\
&\leq 2 \cdot \mathbf{E}\left( \min_{l \in \Theta_n} \frac{1}{n} \sum_{i=1}^n |g_n(X_i) - Y_i|^2 + pen_n(g_{n,l}) - \frac{1}{n} \sum_{i=1}^n |m(X_i) - Y_i|^2 \right)
\end{aligned}$$

$$
\begin{aligned}
&\leq \quad 2 \cdot \mathbf{E}\Bigg( \min_{l \in \Theta_n} \mathbf{E}\Bigg( \frac{1}{n} \sum_{i=1}^{n} |g_{n,l}(X_i) - Y_i|^2 + pen_n(g_{n,l}) \\
&\qquad\qquad - \frac{1}{n} \sum_{i=1}^{n} |m(X_i) - Y_i|^2 \Bigg| \mathbf{b}_1^{(1)}, \dots, \mathbf{b}_r^{(1)}, \dots, \mathbf{b}_1^{(I_n)}, \dots, \mathbf{b}_r^{(I_n)} \Bigg) \Bigg) \\
&\leq \quad 2 \cdot \mathbf{E}\left( \min_{l \in \Theta_n} \int |g_{n,l}(x) - m(x)|^2 \mathbf{P}_X(dx) + pen_n(g_{n,l}) \right)
\end{aligned}
$$

because $|T_\beta z - y| \leq |z - y|$ holds for $|y| \leq \beta$. Hence

$$
\mathbf{E}(T_{4,n}) \quad \leq \quad c_{53} \cdot \frac{(\log n)^2}{n} + 2 \cdot \mathbf{E}\left( \min_{l \in \Theta_n} \int |g_{n,l}(x) - m(x)|^2 \mathbf{P}_X(dx) + pen_n(g_{n,l}) \right)
$$

holds. Thus the proof of Lemma 8 is complete. □