*The Annals of Statistics*
2020, Vol. 48, No. 4, 1875–1897
https://doi.org/10.1214/19-AOS1875


# NONPARAMETRIC REGRESSION USING DEEP NEURAL NETWORKS WITH RELU ACTIVATION FUNCTION[1]

BY JOHANNES SCHMIDT-HIEBER

*Department of Applied Mathematics, University of Twente, a.j.schmidt-hieber@utwente.nl*

Consider the multivariate nonparametric regression model. It is shown that estimators based on sparsely connected deep neural networks with ReLU activation function and properly chosen network architecture achieve the minimax rates of convergence (up to $\log n$-factors) under a general composition assumption on the regression function. The framework includes many well-studied structural constraints such as (generalized) additive models. While there is a lot of flexibility in the network architecture, the tuning parameter is the sparsity of the network. Specifically, we consider large networks with number of potential network parameters exceeding the sample size. The analysis gives some insights into why multilayer feedforward neural networks perform well in practice. Interestingly, for ReLU activation function the depth (number of layers) of the neural network architectures plays an important role, and our theory suggests that for nonparametric regression, scaling the network depth with the sample size is natural. It is also shown that under the composition assumption wavelet estimators can only achieve suboptimal rates.

**1. Introduction.** In the nonparametric regression model with random covariates in the unit hypercube, we observe $n$ i.i.d. vectors $\mathbf{X}_i \in [0,1]^d$ and $n$ responses $Y_i \in \mathbb{R}$ from the model

$$Y_i = f_0(\mathbf{X}_i) + \varepsilon_i, \quad i = 1, \dots, n. \tag{1}$$

The noise variables $\varepsilon_i$ are assumed to be i.i.d. standard, normal and independent of $(\mathbf{X}_i)_i$. The statistical problem is to recover the unknown function $f_0 : [0,1]^d \to \mathbb{R}$ from the sample $(\mathbf{X}_i, Y_i)_i$. Various methods exist that allow to estimate the regression function nonparametrically, including kernel smoothing, series estimators/wavelets and splines; cf. [15, 50, 51]. In this work we consider fitting a multilayer feedforward artificial neural network to the data. It is shown that the estimator achieves nearly optimal convergence rates under various constraints on the regression function.

Multilayer (or deep) neural networks have been successfully trained recently to achieve impressive results for complicated tasks such as object detection on images and speech recognition. Deep learning is now considered to be the state-of-the art for these tasks. But there is a lack of mathematical understanding. One problem is that fitting a neural network to data is highly nonlinear in the parameters. Moreover, the function class is nonconvex, and different regularization methods are combined in practice.

This article is inspired by the idea to build a statistical theory that provides some understanding of these procedures. As the full method is too complex to be theoretically tractable, we need to make some selection of important characteristics that we believe are crucial for the success of the procedure.

Received May 2018; revised March 2019.

[1]Discussed in 10.1214/19-AOS1910, 10.1214/19-AOS1911, 10.1214/19-AOS1912, 10.1214/19-AOS1915; rejoinder at 10.1214/19-AOS1931.

*MSC2020 subject classifications.* 62G08.

*Key words and phrases.* Nonparametric regression, multilayer neural networks, ReLU activation function, minimax estimation risk, additive models, wavelets.

To fit a neural network, an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ needs to be chosen. Traditionally, sigmoidal activation functions (differentiable functions that are bounded and monotonically increasing) were employed. For deep neural networks, however, there is a computational advantage using the nonsigmoidal rectified linear unit (ReLU) $\sigma(x) = \max(x, 0) = (x)_+$. In terms of statistical performance, the ReLU outperforms sigmoidal activation functions for classification problems [13, 38], but for regression this remains unclear; see [6], Supplementary Material B. Whereas earlier statistical work focuses mainly on shallow networks with sigmoidal activation functions, we provide statistical theory specifically for deep ReLU networks.

The statistical analysis for the ReLU activation function is quite different from earlier approaches, and we discuss this in more detail in the overview on related literature in Section 6. Viewed as a nonparametric method, ReLU networks have some surprising properties. To explain this, notice that deep networks with ReLU activation produce functions that are piecewise linear in the input. Nonparametric methods which are based on piecewise linear approximations are typically not able to capture higher-order smoothness in the signal and are rate-optimal only up to smoothness index two. Interestingly, ReLU activation combined with a deep network architecture achieves near minimax rates for arbitrary smoothness of the regression function.

The number of hidden layers of state-of-the-art network architectures has been growing over the past years; cf. [48]. There are versions of the recently developed deep network ResNet which are based on 152 layers; cf. [18]. Our analysis indicates that for the ReLU activation function the network depth should be scaled with the sample size. This suggests that, for larger samples, additional hidden layers should be added.

Recent deep architectures include more network parameters than training samples. The well-known AlexNet [28], for instance, is based on 60 million network parameters using only 1.2 million samples. We account for high-dimensional parameter spaces in our analysis by assuming that the number of potential network parameters is much larger than the sample size. For noisy data generated from the nonparametric regression model, overfitting does not lead to good generalization errors and incorporating regularization or sparsity in the estimator becomes essential. In the deep networks literature, one option is to make the network thinner assuming that only few parameters are nonzero (or active); cf. [14], Section 7.10. Our analysis shows that the number of nonzero parameters plays the role of the effective model dimension and, as is common in nonparametric regression, needs to be chosen carefully.

Existing statistical theory often requires that the size of the network parameters tends to infinity as the sample size increases. In practice, estimated network weights are, however, rather small. We can incorporate small parameters in our theory, proving that it is sufficient to consider neural networks with all network parameters bounded in absolute value by one.

Multilayer neural networks are typically applied to high-dimensional input. Without additional structure in the signal besides smoothness, nonparametric estimation rates are then slow because of the well-known curse of dimensionality. This means that no statistical procedure can do well regarding pointwise reconstruction of the signal. Multilayer neural networks are believed to be able to adapt to many different structures in the signal, therefore avoiding the curse of dimensionality and achieving faster rates in many situations. In this work we stick to the regression setup and show that deep ReLU networks can indeed attain faster rates under a hierarchical composition assumption on the regression function which includes (generalized) additive models and the composition models considered in [3, 6, 21, 22, 26].

Parts of the success of multilayer neural networks can be explained by the fast algorithms that are available to estimate the network weights from data. These iterative algorithms are based on minimization of some empirical loss function using stochastic gradient descent. Because of the nonconvex function space, gradient descent methods might get stuck in a saddle

point or converge to one of the potentially many local minima. Choromanska et al. [9] derive a heuristic argument and shows that the risk of most of the local minima is not much larger than the risk of the global minimum. Despite the huge number of variations of the stochastic gradient descent, the common objective of all approaches is to reduce the empirical loss. In our framework we associate to any network reconstruction method a parameter quantifying the expected discrepancy between the achieved empirical risk and the global minimum of the energy landscape. The main theorem then states that a network estimator is minimax rate optimal (up to log factors) if and only if the method almost minimizes the empirical risk.

We also show that wavelet series estimators are unable to adapt to the underlying structure under the composition assumption on the regression function. By deriving lower bounds, it is shown that the rates are suboptimal by a polynomial factor in the sample size $n$. This provides an example of a function class for which fitting a neural network outperforms wavelet series estimators.

Our setting deviates in two aspects from the computer science literature on deep learning. First, we consider regression and not classification. Second, we restrict ourselves in this article to multilayer feedforward artificial neural networks, while most of the many recent deep learning applications have been obtained using specific types of networks such as convolutional or recurrent neural networks.

The article is structured as follows. Section 2 introduces multilayer feedforward artificial neural networks and discusses mathematical modeling. This section also contains the definition of the network classes. The considered function classes for the regression function and the main result can be found in Section 3. Specific structural constraints, such as additive models, are discussed in Section 4. In Section 5 it is shown that wavelet estimators can only achieve suboptimal rates under the composition assumption. We give an overview of relevant related literature in Section 6. The proof of the main result together with additional discussion can be found in Section 7.

*Notation*: Vectors are denoted by bold letters, for example, $\mathbf{x} := (x_1, \ldots, x_d)^\top$. As usual, we define $|\mathbf{x}|_p := (\sum_{i=1}^d |x_i|^p)^{1/p}$, $|\mathbf{x}|_\infty := \max_i |x_i|$, $|\mathbf{x}|_0 := \sum_i \mathbf{1}(x_i \neq 0)$ and write $\|f\|_p := \|f\|_{L^p(D)}$ for the $L^p$-norm on $D$, whenever there is no ambiguity of the domain $D$. For two sequences, $(a_n)_n$ and $(b_n)_n$, we write $a_n \lesssim b_n$ if there exists a constant $C$ such that $a_n \leq C b_n$ for all $n$. Moreover, $a_n \asymp b_n$ means that $a_n \lesssim b_n$ and $b_n \lesssim a_n$. We denote by $\log_2$ the logarithm with respect to the basis two and write $\lceil x \rceil$ for the smallest integer $\geq x$.

**2. Mathematical definition of multilayer neural networks.** *Definitions*: Fitting a multilayer neural network requires the choice of an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ and the network architecture. Motivated by the importance in deep learning, we study the rectifier linear unit (ReLU) activation function

$$\sigma(x) = \max(x, 0).$$

For $\mathbf{v} = (v_1, \ldots, v_r) \in \mathbb{R}^r$, define the shifted activation function $\sigma_{\mathbf{v}} : \mathbb{R}^r \to \mathbb{R}^r$ as

$$\sigma_{\mathbf{v}} \begin{pmatrix} y_1 \\ \vdots \\ y_r \end{pmatrix} = \begin{pmatrix} \sigma(y_1 - v_1) \\ \vdots \\ \sigma(y_r - v_r) \end{pmatrix}.$$

The network architecture $(L, \mathbf{p})$ consists of a positive integer $L$, called the *number of hidden layers* or *depth*, and a *width vector* $\mathbf{p} = (p_0, \ldots, p_{L+1}) \in \mathbb{N}^{L+2}$. A neural network with network architecture $(L, \mathbf{p})$ is then any function of the form

$$f : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{L+1}}, \quad \mathbf{x} \mapsto f(\mathbf{x}) = W_L \sigma_{\mathbf{v}_L} W_{L-1} \sigma_{\mathbf{v}_{L-1}} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}, \tag{2}$$

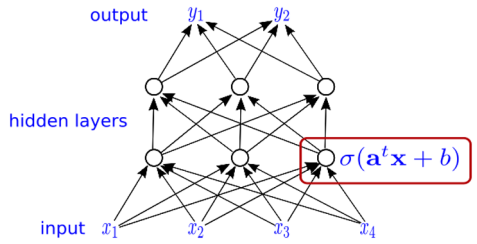


FIG. 1. *Representation as a direct graph of a network with two hidden layers* $L = 2$ *and width vector* $\mathbf{p} = (4, 3, 3, 2)$.

where $W_i$ is a $p_{i+1} \times p_i$ weight matrix and $\mathbf{v}_i \in \mathbb{R}^{p_i}$ is a shift vector. Network functions are therefore built by alternating matrix-vector multiplications with the action of the nonlinear activation function $\sigma$. In (2), it is also possible to omit the shift vectors by considering the input $(\mathbf{x}, 1)$ and enlarging the weight matrices by one row and one column with appropriate entries. For our analysis it is, however, more convenient to work with representation (2). To fit networks to data generated from the $d$-variate nonparametric regression model we must have $p_0 = d$ and $p_{L+1} = 1$.

In computer science, neural networks are more commonly introduced via their representation as directed acyclic graphs; cf. Figure 1. Using this equivalent definition, the nodes in the graph (also called *units*) are arranged in layers. The input layer is the first layer and the output layer the last layer. The layers that lie in between are called hidden layers. The number of hidden layers corresponds to $L$, and the number of units in each layer generates the width vector $\mathbf{p}$. Each node/unit in the graph representation stands for a scalar product of the incoming signal with a weight vector which is then shifted and applied to the activation function.

*Mathematical modeling of deep network characteristics*: Given a network function $f(\mathbf{x}) = W_L\sigma_{\mathbf{v}_L}W_{L-1}\sigma_{\mathbf{v}_{L-1}}\cdots W_1\sigma_{\mathbf{v}_1}W_0\mathbf{x}$, the network parameters are the entries of the matrices $(W_j)_{j=0,\ldots,L}$ and vectors $(\mathbf{v}_j)_{j=1,\ldots,L}$. These parameters need to be estimated/learned from the data.

The aim of this article is to consider a framework that incorporates essential features of modern deep network architectures. In particular, we allow for large depth $L$ and a large number of potential network parameters. For the main result, no upper bound on the number of network parameters is needed. Thus, we consider high-dimensional settings with more parameters than training data.

Another characteristic of trained networks is that the size of the learned network parameters is typically not very large. Common network initialization methods initialize the weight matrices $W_j$ by a (nearly) orthogonal random matrix if two successive layers have the same width; cf. [14], Section 8.4. In practice, the trained network weights are typically not far from the initialized weights. Since in an orthogonal matrix all entries are bounded in absolute value by one, the trained network weights will not be large.

Existing theoretical results, however, often require that the size of the network parameters tends to infinity. If large parameters are allowed, one can, for instance, easily approximate step functions by ReLU networks. To be more in line with what is observed in practice, we consider networks with all parameters bounded by one. This constraint can be easily built into the deep learning algorithm by projecting the network parameters in each iteration onto the interval $[-1, 1]$.

If $\|W_j\|_\infty$ denotes the maximum-entry norm of $W_j$, the space of network functions with given network architecture and network parameters bounded by one is

$$\mathcal{F}(L, \mathbf{p}) := \Big\{ f \text{ of the form (2)} : \max_{j=0,\ldots,L} \|W_j\|_\infty \vee |\mathbf{v}_j|_\infty \le 1 \Big\}, \tag{3}$$

with the convention that $\mathbf{v}_0$ is a vector with all components equal to zero.

In deep learning, sparsity of the neural network is enforced through regularization or specific forms of networks. Dropout for instance sets randomly units to zero and has the effect that each unit will be active only for a small fraction of the data; cf. [44], Section 7.2. In our notation this means that each entry of the vectors $\sigma_{\mathbf{v}_k} W_{k-1} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}$, $k = 1, \ldots, L$ is zero over a large range of the input space $\mathbf{x} \in [0,1]^d$. Convolutional neural networks filter the input over local neighborhoods. Rewritten in the form (2) this essentially means that the $W_i$ are banded Toeplitz matrices. All network parameters corresponding to higher off-diagonal entries are thus set to zero.

In this work we model the network sparsity assuming that there are only few nonzero/active network parameters. If $\|W_j\|_0$ denotes the number of nonzero entries of $W_j$ and $\||f|_\infty\|_\infty$ stands for the sup-norm of the function $\mathbf{x} \mapsto |f(\mathbf{x})|_\infty$, then the $s$-sparse networks are given by

$$\begin{aligned} &\mathcal{F}(L, \mathbf{p}, s) := \mathcal{F}(L, \mathbf{p}, s, F) \\ &:= \left\{ f \in \mathcal{F}(L, \mathbf{p}) : \sum_{j=0}^{L} \|W_j\|_0 + |\mathbf{v}_j|_0 \le s, \left\||f|_\infty\right\|_\infty \le F \right\}. \end{aligned} \tag{4}$$

The upper bound on the uniform norm of $f$ is most of the time dispensable and, therefore, omitted in the notation. We consider cases where the number of network parameters $s$ is small compared to the total number of parameters in the network.

In deep learning, it is common to apply variations of stochastic gradient descent combined with other techniques such as dropout to the loss induced by the log-likelihood (see Section 6.2.1.1 in [14]). For nonparametric regression with normal errors, this coincides with the least-squares loss (in machine learning terms this is the cross entropy for this model; cf. [14], p. 129). The common objective of all reconstruction methods is to find networks $f$ with small empirical risk $\frac{1}{n}\sum_{i=1}^n (Y_i - f(\mathbf{X}_i))^2$. For any estimator $\widehat{f}_n$ that returns a network in the class $\mathcal{F}(L, \mathbf{p}, s, F)$, we define the corresponding quantity

$$\begin{aligned} &\Delta_n(\widehat{f}_n, f_0) \\ &:= E_{f_0}\left[ \frac{1}{n}\sum_{i=1}^n \big(Y_i - \widehat{f}_n(\mathbf{X}_i)\big)^2 - \inf_{f \in \mathcal{F}(L,\mathbf{p},s,F)} \frac{1}{n}\sum_{i=1}^n \big(Y_i - f(\mathbf{X}_i)\big)^2 \right]. \end{aligned} \tag{5}$$

The sequence $\Delta_n(\widehat{f}_n, f_0)$ measures the difference between the expected empirical risk of $\widehat{f}_n$ and the global minimum over all networks in the class. The subscript $f_0$ in $E_{f_0}$ indicates that the expectation is taken with respect to a sample generated from the nonparametric regression model with regression function $f_0$. Notice that $\Delta_n(\widehat{f}_n, f_0) \ge 0$ and $\Delta_n(\widehat{f}_n, f_0) = 0$ if $\widehat{f}_n$ is an empirical risk minimizer.

To evaluate the statistical performance of an estimator $\widehat{f}_n$, we derive bounds for the prediction error

$$R(\widehat{f}_n, f_0) := E_{f_0}\big[\big(\widehat{f}_n(\mathbf{X}) - f_0(\mathbf{X})\big)^2\big],$$

with $\mathbf{X} \stackrel{\mathcal{D}}{=} \mathbf{X}_1$ being independent of the sample $(\mathbf{X}_i, Y_i)_i$.

The term $\Delta_n(\widehat{f}_n, f_0)$ can be related via empirical process theory to constant $\times$ $(R(\widehat{f}_n, f_0) - R(\widehat{f}_n^{\mathrm{ERM}}, f_0)) +$ remainder, with $\widehat{f}_n^{\mathrm{ERM}}$ an empirical risk minimizer. Therefore, $\Delta_n(\widehat{f}_n, f_0)$ is the key quantity that together with the minimax estimation rate sharply determines the convergence rate of $\widehat{f}_n$ (up to $\log n$-factors). Determining the decay of $\Delta_n(\widehat{f}_n, f_0)$ in $n$ for commonly employed methods, such as stochastic gradient descent, is an interesting problem in its own. We only sketch a possible proof strategy here. Because of the potentially many local minima and saddle points of the loss surface or energy landscape, gradient descent

based methods have only a small chance to reach the global minimum without getting stuck in a local minimum first. By making a link to spherical spin glasses, Choromanska et al. [9] provide a heuristic suggesting that the loss of any local minima lies in a band that is lower bounded by the loss of the global minimum. The width of the band depends on the width of the network. If the heuristic argument can be made rigorous, then the width of the band provides an upper bound for $\Delta_n(\widehat{f}_n, f_0)$ for all methods that converge to a local minimum. This would allow us then to study deep learning without an explicit analysis of the algorithm. For more on the energy landscape, see [31].

**3. Main results.** The theoretical performance of neural networks depends on the underlying function class. The classical approach in nonparametric statistics is to assume that the regression function is $\beta$-smooth. The minimax estimation rate for the prediction error is then $n^{-2\beta/(2\beta+d)}$. Since the input dimension $d$ in neural network applications is very large, these rates are extremely slow. The huge sample sizes often encountered in deep learning applications are by far not sufficient to compensate the slow rates.

We therefore consider a function class that is natural for neural networks and exhibits some low-dimensional structure that leads to input dimension free exponents in the estimation rates. We assume that the regression function $f_0$ is a composition of several functions, that is,

$$f_0 = g_q \circ g_{q-1} \circ \cdots \circ g_1 \circ g_0 \tag{6}$$

with $g_i : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}$. Denote by $g_i = (g_{ij})^\top_{j=1,\ldots,d_{i+1}}$ the components of $g_i$, and let $t_i$ be the maximal number of variables on which each of the $g_{ij}$ depends on. Thus, each $g_{ij}$ is a $t_i$-variate function. As an example consider the function $f_0(x_1, x_2, x_3) = g_{11}(g_{01}(x_1, x_3), g_{02}(x_1, x_2))$ for which $d_0 = 3$, $t_0 = 2$, $d_1 = t_1 = 2$, $d_2 = 1$. We always must have $t_i \le d_i$ and for specific constraints, such as additive models, $t_i$ might be much smaller than $d_i$. The single components $g_0, \ldots, g_q$ and the pairs $(\beta_i, t_i)$ are obviously not identifiable. As we are only interested in estimation of $f_0$, this causes no problems. Among all possible representations, one should always pick one that leads to the fastest estimation rate in Theorem 1 below.

In the $d$-variate regression model (1), $f_0 : [0,1]^d \to \mathbb{R}$ and, thus, $d_0 = d$, $a_0 = 0$, $b_0 = 1$ and $d_{q+1} = 1$. One should keep in mind that (6) is an assumption on the regression function that can be made independently of whether neural networks are used to fit the data or not. In particular, the number of layers $L$ in the network has not to be the same as $q$.

It is conceivable that for many of the problems for which neural networks perform well, a hidden hierarchical input-output relationship of the form (6) is present with small values $t_i$; cf. [35, 40]. Slightly more specific function spaces, which alternate between summations and composition of functions, have been considered in [6, 21]. We provide below an example of a function class that can be decomposed in the form (6) but is not contained in these spaces.

Recall that a function has Hölder smoothness index $\beta$ if all partial derivatives up to order $\lfloor\beta\rfloor$ exist and are bounded, and the partial derivatives of order $\lfloor\beta\rfloor$ are $\beta - \lfloor\beta\rfloor$ Hölder, where $\lfloor\beta\rfloor$ denotes the largest integer strictly smaller than $\beta$. The ball of $\beta$-Hölder functions with radius $K$ is then defined as

$$\mathcal{C}_r^\beta(D, K) = \Bigg\{ f : D \subset \mathbb{R}^r \to \mathbb{R} : \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|<\beta} \|\partial^{\boldsymbol{\alpha}} f\|_\infty + \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=\lfloor\beta\rfloor} \sup_{\substack{\mathbf{x},\mathbf{y}\in D \\ \mathbf{x}\neq\mathbf{y}}} \frac{|\partial^{\boldsymbol{\alpha}} f(\mathbf{x}) - \partial^{\boldsymbol{\alpha}} f(\mathbf{y})|}{|\mathbf{x}-\mathbf{y}|_\infty^{\beta-\lfloor\beta\rfloor}} \le K \Bigg\},$$

where we used multi-index notation, that is, $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \ldots \partial^{\alpha_r}$ with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_r) \in \mathbb{N}^r$ and $|\boldsymbol{\alpha}| := |\boldsymbol{\alpha}|_1$.

We assume that each of the functions $g_{ij}$ has Hölder smoothness $\beta_i$. Since $g_{ij}$ is also $t_i$-variate, $g_{ij} \in \mathcal{C}_{t_i}^{\beta_i}([a_i, b_i]^{t_i}, K_i)$, and the underlying function space becomes

$$\mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K) := \big\{ f = g_q \circ \ldots \circ g_0 : g_i = (g_{ij})_j : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}},$$
$$g_{ij} \in \mathcal{C}_{t_i}^{\beta_i}([a_i, b_i]^{t_i}, K), \text{ for some } |a_i|, |b_i| \le K \big\},$$

with $\mathbf{d} := (d_0, \ldots, d_{q+1})$, $\mathbf{t} := (t_0, \ldots, t_q)$, $\boldsymbol{\beta} := (\beta_0, \ldots, \beta_q)$.

For estimation rates in the nonparametric regression model, the crucial quantity is the smoothness of $f$. Imposing smoothness on the functions $g_i$, we must then find the induced smoothness on $f$. If, for instance, $q = 1$, $\beta_0, \beta_1 \le 1$, $d_0 = d_1 = t_0 = t_1 = 1$, then $f = g_1 \circ g_0$ and $f$ has smoothness $\beta_0\beta_1$; cf. [22, 41]. We should then be able to achieve at least the convergence rate $n^{-2\beta_0\beta_1/(2\beta_0\beta_1+1)}$. For $\beta_1 > 1$, the rate changes. Below we see that the convergence of the network estimator is described by the effective smoothness indices

$$\beta_i^* := \beta_i \prod_{\ell=i+1}^{q} (\beta_\ell \wedge 1)$$

via the rate

$$\phi_n := \max_{i=0,\ldots,q} n^{-\frac{2\beta_i^*}{2\beta_i^*+t_i}}. \tag{7}$$

Recall the definition of $\Delta_n(\widehat{f}_n, f_0)$ in (5). We can now state the main result.

THEOREM 1. *Consider the $d$-variate nonparametric regression model (1) for composite regression function (6) in the class $\mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)$. Let $\widehat{f}_n$ be an estimator taking values in the network class $\mathcal{F}(L, (p_i)_{i=0,\ldots,L+1}, s, F)$ satisfying*:

(i) $F \ge \max(K, 1)$,
(ii) $\sum_{i=0}^{q} \log_2(4t_i \vee 4\beta_i) \log_2 n \le L \lesssim n\phi_n$,
(iii) $n\phi_n \lesssim \min_{i=1,\ldots,L} p_i$,
(iv) $s \asymp n\phi_n \log n$.

*There exist constants $C, C'$ only depending on $q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, F$, such that if*

$$\Delta_n(\widehat{f}_n, f_0) \le C\phi_n L \log^2 n,$$

*then*

$$R(\widehat{f}_n, f_0) \le C'\phi_n L \log^2 n, \tag{8}$$

*and if $\Delta_n(\widehat{f}_n, f_0) \ge C\phi_n L \log^2 n$, then*

$$\frac{1}{C'}\Delta_n(\widehat{f}_n, f_0) \le R(\widehat{f}_n, f_0) \le C'\Delta_n(\widehat{f}_n, f_0). \tag{9}$$

In order to minimize the rate $\phi_n L \log^2 n$, the best choice is to choose $L$ of the order of $\log_2 n$. The rate in the regime $\Delta_n(\widehat{f}_n, f_0) \le C\phi_n \log^3 n$ becomes then

$$R(\widehat{f}_n, f_0) \le C'\phi_n \log^3 n.$$

The convergence rate in Theorem 1 depends on $\phi_n$ and $\Delta_n(\widehat{f}_n, f_0)$. Below we show that $\phi_n$ is a lower bound for the minimax estimation risk over this class. Recall that the term $\Delta_n(\widehat{f}_n, f_0)$ is large if $\widehat{f}_n$ has a large empirical risk compared to an empirical risk minimizer. Having this term in the convergence rate is unavoidable as it also appears in the lower bound in (9). Since for any empirical risk minimizer the $\Delta_n$-term is zero by definition, we have the following direct consequence of the main theorem:

COROLLARY 1. *Let* $\widetilde{f}_n \in \arg\min_{f\in\mathcal{F}(L,\mathbf{p},s,F)} \sum_{i=1}^n (Y_i - f(\mathbf{X}_i))^2$ *be an empirical risk minimizer. Under the same conditions as for Theorem* 1, *there exists a constant* $C'$, *only depending on* $q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, F$, *such that*

$$R(\widetilde{f}_n, f_0) \le C'\phi_n L \log^2 n. \tag{10}$$

Condition (i) in Theorem 1 is very mild and only states that the network functions should have at least the same supremum norm as the regression function. From the other assumptions in Theorem 1, it becomes clear that there is a lot of flexibility in picking a good network architecture as long as the number of active parameters $s$ is taken to be of the right order. Interestingly, to choose a network depth $L$, it is sufficient to have an upper bound on the $t_i \le d_i$ and the smoothness indices $\beta_i$. The network width can be chosen independent of the smoothness indices by taking, for instance, $n \lesssim \min_i p_i$. One might wonder whether for an empirical risk minimizer the sparsity $s$ can be made adaptive by minimizing a penalized least squares problem with sparsity inducing penalty on the network weights. It is conceivable that a complexity penalty of the form $\lambda s$ will lead to adaptation if the regularization parameter $\lambda$ is chosen of the correct order. From a practical point of view, it is more interesting to study $\ell^1/\ell^2$-weight decay. As this requires much more machinery, the question will be moved to future work.

The number of network parameters in a fully connected network is of the order $\sum_{i=0}^L p_i p_{i+1}$. This shows that Theorem 1 requires sparse networks. More specifically, the network has at least $\sum_{i=1}^L p_i - s$ completely inactive nodes, meaning that all incoming signal is zero. The choice $s \asymp n\phi_n \log n$ in condition (iv) balances the squared bias and the variance. From the proof of the theorem, convergence rates can also be derived if $s$ is chosen of a different order.

For convenience, Theorem 1 is stated without explicit constants. The proofs, however, are nonasymptotic, although we did not make an attempt to minimize the constants. It is well known that deep learning outperforms other methods only for large sample sizes. This indicates that the method might be able to adapt to underlying structure in the signal and, therefore, to achieve fast convergence rates but with large constants or remainder terms which spoil the results for small samples.

The proof of the risk bounds in Theorem 1 is based on the following oracle-type inequality:

THEOREM 2. *Consider the* $d$*-variate nonparametric regression model* (1) *with unknown regression function* $f_0$, *satisfying* $\|f_0\|_\infty \le F$ *for some* $F \ge 1$. *Let* $\widehat{f}_n$ *be any estimator taking values in the class* $\mathcal{F}(L, \mathbf{p}, s, F)$, *and let* $\Delta_n(\widehat{f}_n, f_0)$ *be the quantity defined in* (5). *For any* $\varepsilon \in (0, 1]$, *there exists a constant* $C_\varepsilon$, *only depending on* $\varepsilon$, *such that with*

$$\tau_{\varepsilon,n} := C_\varepsilon F^2 \frac{(s+1)\log(n(s+1)^L p_0 p_{L+1})}{n},$$

$$(1-\varepsilon)^2\Delta_n(\widehat{f}_n, f_0) - \tau_{\varepsilon,n} \le R(\widehat{f}_n, f_0)$$
$$\le (1+\varepsilon)^2\Big(\inf_{f\in\mathcal{F}(L,\mathbf{p},s,F)} \|f - f_0\|_\infty^2 + \Delta_n(\widehat{f}_n, f_0)\Big) + \tau_{\varepsilon,n}.$$

One consequence of the oracle inequality is that the upper bounds on the risk become worse if the number of layers increases. In practice, it also has been observed that too many layers lead to a degradation of the performance; cf. [18], [17], Section 4.4 and [45], Section 4. Residual networks can overcome this problem. But they are not of the form (2) and will need to be analyzed separately.

One may wonder whether there is anything special about ReLU networks compared to other activation functions. A close inspection of the proof shows that two specific properties of the ReLU function are used.

One of the advantages of deep ReLU networks is the projection property

$$\sigma \circ \sigma = \sigma \tag{11}$$

that we can use to pass a signal without change through several layers in the network. This is important since the approximation theory is based on the construction of smaller networks for simpler tasks that might not all have the same network depth. To combine these subnetworks we need to synchronize the network depths by adding hidden layers that do not change the output. This can be easily realized by choosing the weight matrices in the network to be the identity (assuming equal network width in successive layers) and using (11); see also (18). This property is not only a theoretical tool. To pass an outcome without change to a deeper layer is also often helpful in practice and realized by so-called skip connections, in which case they do not need to be learned from the data. A specific instance are residual networks with ReLU activation function [18] that are successfully applied in practice. The difference to standard feedforward networks is that if all networks parameters are set to zero in a residual network, the network becomes essentially the identity map. For other activation functions it is much harder to approximate the identity.

Another advantage of the ReLU activation is that all network parameters can be taken to be bounded in absolute value by one. If all network parameters are initialized by a value in $[-1, 1]$, this means that each network parameter only need to be varied by at most two during training. It is unclear whether other results in the literature for non-ReLU activation functions hold for bounded network parameters. An important step is the approximation of the square function $x \mapsto x^2$. For any twice differentiable and nonlinear activation function, the classical approach to approximate the square function by a network is to use rescaled second order differences $(\sigma(t + 2xh) - 2\sigma(t + xh) + \sigma(xh))/(h^2\sigma''(t)) \to x^2$ for $h \to 0$ and a $t$ with $\sigma''(t) \neq 0$. To achieve a sufficiently good approximation, we need to let $h$ tend to zero with the sample size, making some of the network parameters necessarily very large.

The $\log^2 n$-factor in the convergence rate $\phi_n L \log^2 n$ is likely an artifact of the proof. Next, we show that $\phi_n$ is a lower bound for the minimax estimation risk over the class $\mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)$ in the interesting regime $t_i \leq \min(d_0, \ldots, d_{i-1})$ for all $i$. This means that no dimensions are added on deeper abstraction levels in the composition of functions. In particular, it avoids that $t_i$ is larger than the input dimension $d_0$. Outside of this regime, it is hard to determine the minimax rate, and in some cases it is even possible to find another representation of $f$ as a composition of functions which yields a faster convergence rate.

THEOREM 3. *Consider the nonparametric regression model* (1) *with* $\mathbf{X}_i$ *drawn from a distribution with Lebesgue density on* $[0, 1]^d$ *which is lower and upper bounded by positive constants. For any nonnegative integer* $q$, *any dimension vectors* $\mathbf{d}$ *and* $\mathbf{t}$ *satisfying* $t_i \leq \min(d_0, \ldots, d_{i-1})$ *for all* $i$, *any smoothness vector* $\boldsymbol{\beta}$ *and all sufficiently large constants* $K > 0$, *there exists a positive constant* $c$ *such that*

$$\inf_{\widehat{f}_n} \sup_{f_0 \in \mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)} R(\widehat{f}_n, f_0) \geq c\phi_n,$$

*where the* inf *is taken over all estimators* $\widehat{f}_n$.

The proof is deferred to Section 7. To illustrate the main ideas, we provide a sketch here. For simplicity, assume that $t_i = d_i = 1$ for all $i$. In this case the functions $g_i$ are univariate and real valued. Define $i^* \in \arg\min_{i=0,\ldots,q} \beta_i^*/(2\beta_i^* + 1)$ as an index for which the estimation

rate is obtained. For any $\alpha > 0$, $x^\alpha$ has Hölder smoothness $\alpha$, and for $\alpha = 1$, the function is infinitely often differentiable and has finite Hölder norm for all smoothness indices. Set $g_\ell(x) = x$ for $\ell < i^*$ and $g_\ell(x) = x^{\beta_\ell \wedge 1}$ for $\ell > i^*$. Then,

$$f_0(x) = g_q \circ g_{q-1} \circ \cdots \circ g_1 \circ g_0(x) = \big(g_{i^*}(x)\big)^{\prod_{\ell=i^*+1}^q \beta_\ell \wedge 1}.$$

Assuming for the moment uniform random design, the Kullback–Leibler divergence is $\mathrm{KL}(P_f, P_g) = \frac{n}{2}\|f - g\|_2^2$. Take a kernel function $K$, and consider $\widetilde{g}(x) = h^{\beta_{i^*}} K(x/h)$. Under standard assumptions on $K$, $\widetilde{g}$ has Hölder smoothness index $\beta_{i^*}$. Now, we can generate two hypotheses $f_{00}(x) = 0$ and $f_{01}(x) = (h^{\beta_{i^*}} K(x/h))^{\prod_{\ell=i^*+1}^q \beta_\ell \wedge 1}$ by taking $g_{i^*}(x) = 0$ and $g_{i^*}(x) = \widetilde{g}(x)$. Therefore, $|f_{00}(0) - f_{01}(0)| \gtrsim h^{\beta_{i^*}^*}$ assuming that $K(0) > 0$. For the Kullback–Leibler divergence, we find $\mathrm{KL}(P_{f_{00}}, P_{f_{01}}) \lesssim nh^{2\beta_{i^*}^*+1}$. Using Theorem 2.2(iii) in [50], this shows that the pointwise rate of convergence is $n^{-2\beta_{i^*}^*/(2\beta_{i^*}^*+1)} = \max_{i=0,\ldots,q} n^{-2\beta_i^*/(2\beta_i^*+1)}$. This matches with the upper bound since $t_i = 1$ for all $i$. For lower bounds on the prediction error, we generalize the argument to a multiple testing problem.

The $L^2$-minimax rate coincides in most regimes with the sup-norm rate obtained in Section 4.1 of [22] for composition of two functions. But unlike the classical nonparametric regression model, the minimax estimation rates for $L^2$-loss and sup-norm loss differ for some setups by a polynomial power in the sample size $n$.

There are several recent results in approximation theory that provide lower bounds on the number of required network weights $s$ such that all functions in a function class can be approximated by a $s$-sparse network up to some prescribed error; cf., for instance, [7]. Results of this flavor can also be quite easily derived by combining the minimax lower bound with the oracle inequality. The argument is that if the same approximation rates would hold for networks with less parameters, then we would obtain rates that are faster than the minimax rates which clearly is a contradiction. This provides a new statistical route to establish approximation theoretic properties.

LEMMA 1. *Given $\beta, K > 0$, $d \in \mathbb{N}$, there exist constants $c_1, c_2$ only depending on $\beta, K, d$, such that if*

$$s \le c_1 \frac{\varepsilon^{-d/\beta}}{L \log(1/\varepsilon)}$$

*for some $\varepsilon \le c_2$, then for any width vector $\mathbf{p}$ with $p_0 = d$ and $p_{L+1} = 1$,*

$$\sup_{f_0 \in \mathcal{C}_d^\beta([0,1]^d, K)} \inf_{f \in \mathcal{F}(L,\mathbf{p},s)} \|f - f_0\|_\infty \ge \varepsilon.$$

A more refined argument using Lemma 4 instead of Theorem 2 yields also lower bounds for $L^2$.

**4. Examples of specific structural constraints.** In this section we discuss several well-studied special cases of compositional constraints on the regression function.

*Additive models*: In an additive model the regression function has the form

$$f_0(x_1, \ldots, x_d) = \sum_{i=1}^d f_i(x_i).$$

This can be written as a composition of functions

$$f_0 = g_1 \circ g_0 \tag{12}$$

with $g_0(\mathbf{x}) = (f_1(x_1), \ldots, f_d(x_d))^\top$ and $g_1(\mathbf{y}) = \sum_{j=1}^d y_j$. Consequently, $g_0 : [0,1]^d \to \mathbb{R}^d$ and $g_1 : \mathbb{R}^d \to \mathbb{R}$ and, thus, $d_0 = d$, $t_0 = 1$, $d_1 = t_1 = d$, $d_2 = 1$. Equation (12) decomposes

the original function into one function where each component only depends on one variable only and another function that depends on all variables but is infinitely smooth. For both types of functions, fast rates can be obtained that do not suffer from the curse of dimensionality. This explains then the fast rate that can be obtained for additive models.

Suppose that $f_i \in \mathcal{C}_1^\beta([0,1], K)$ for $i = 1, \dots, d$. Then, $f : [0,1]^d \xrightarrow{g_0} [-K, K]^d \xrightarrow{g_1} [-Kd, Kd]$. Since for any $\gamma > 1$, $g_1 \in \mathcal{C}_d^\gamma([-K, K]^d, (K+1)d)$,

$$f_0 \in \mathcal{G}\big(1, (d, d, 1), (1, d), \big(\beta, (\beta \vee 2)d\big), (K+1)d\big).$$

For network architectures $\mathcal{F}(L, \mathbf{p}, s, F)$ satisfying $F \geq (K+1)d$, $2\log_2(4(\beta \vee 2)d)\log n \leq L \lesssim \log n$, $n^{1/(2\beta+1)} \lesssim \min_i p_i$ and $s \asymp n^{1/(2\beta+1)} \log n$, we thus obtain by Theorem 1,

$$R(\widehat{f}_n, f_0) \lesssim n^{-\frac{2\beta}{2\beta+1}} \log^3 n + \Delta(\widehat{f}_n, f_0).$$

This coincides up to the $\log^3 n$-factor with the minimax estimation rate.

*Generalized additive models*: Suppose the regression function is of the form

$$f_0(x_1, \dots, x_d) = h\left(\sum_{i=1}^d f_i(x_i)\right)$$

for some unknown link function $h : \mathbb{R} \to \mathbb{R}$. This can be written as composition of three functions $f_0 = g_2 \circ g_1 \circ g_0$ with $g_0$ and $g_1$ as before and $g_2 = h$. If $f_i \in \mathcal{C}_1^\beta([0,1], K)$ and $h \in \mathcal{C}_1^\gamma(\mathbb{R}, K)$, then $f_0 : [0,1]^d \xrightarrow{g_0} [-K, K]^d \xrightarrow{g_1} [-Kd, Kd] \xrightarrow{g_2} [-K, K]$. Arguing as for additive models,

$$f_0 \in \mathcal{G}\big(2, (d, d, 1, 1), (1, d, 1), \big(\beta, (\beta \vee 2)d, \gamma\big), (K+1)d\big).$$

For network architectures satisfying the assumptions of Theorem 1, the bound on the estimation rate becomes

$$R(\widehat{f}_n, f_0) \lesssim \big(n^{-\frac{2\beta(\gamma\wedge 1)}{2\beta(\gamma\wedge 1)+1}} + n^{-\frac{2\gamma}{2\gamma+1}}\big)\log^3 n + \Delta(\widehat{f}_n, f_0). \tag{13}$$

Theorem 3 shows that $n^{-2\beta(\gamma\wedge 1)/(2\beta(\gamma\wedge 1)+1)} + n^{-2\gamma/(2\gamma+1)}$ is also a lower bound. Let us also remark that for the special case $\beta = \gamma \geq 2$ and $\beta, \gamma$ integers, Theorem 2.1 of [21] establishes the estimation rate $n^{-2\beta/(2\beta+1)}$.

*Sparse tensor decomposition*: Assume that the regression function $f_0$ has the form

$$f_0(\mathbf{x}) = \sum_{\ell=1}^N a_\ell \prod_{i=1}^d f_{i\ell}(x_i) \tag{14}$$

for fixed $N$, real coefficients $a_\ell$ and univariate functions $f_{i\ell}$. Especially, if $N = 1$, this is the same as imposing a product structure on the regression function $f_0(\mathbf{x}) = \prod_{i=1}^d f_i(x_i)$. The function class spanned by such sparse tensor decomposition can be best explained by making a link to series estimators. Series estimators are based on the idea that the unknown function is close to a linear combination of few basis functions, where the approximation error depends on the smoothness of the signal. This means that any $L^2$-function can be approximated by $f_0(\mathbf{x}) \approx \sum_{\ell=1}^N a_\ell \prod_{i=1}^d \phi_{i\ell}(x_i)$ for suitable coefficients $a_\ell$ and functions $\phi_{i\ell}$.

Whereas series estimators require the choice of a basis, for neural networks to achieve fast rates it is enough that (14) holds. The functions $f_{i\ell}$ can be unknown and do not need to be orthogonal.

We can rewrite (14) as a composition of functions $f_0 = g_2 \circ g_1 \circ g_0$ with $g_0(\mathbf{x}) = (f_{i\ell}(x_i))_{i,\ell}$, $g_1 = (g_{1j})_{j=1,\dots,N}$ performing the $N$ multiplications $\prod_{i=1}^d$ and $g_2(\mathbf{y}) = \sum_{\ell=1}^N a_\ell y_\ell$. Observe that $t_0 = 1$ and $t_1 = d$. Assume that $f_{i\ell} \in \mathcal{C}_1^\beta([0,1], K)$ for $K \geq 1$ and $\max_\ell |a_\ell| \leq 1$. Because of $g_{1,j} \in \mathcal{C}_d^\gamma([-K, K]^d, 2^d K^d)$ for all $\gamma \geq d+1$ and $g_2 \in$

$\mathcal{C}_N^{\gamma'}([-2^dK^d, 2^dK^d]^N, N(2^dK^d+1))$ for $\gamma' > 1$, we have $[0,1]^d \xrightarrow{g_0} [-K,K]^{Nd} \xrightarrow{g_1} [-2^dK^d, 2^dK^d]^N \xrightarrow{g_2} [-N(2^dK^d+1), N(2^dK^d+1)]$ and

$$f_0 \in \mathcal{G}\big(2, (d, Nd, N, 1), (1, d, Nd), \big(\beta, \beta d \vee (d+1), N\beta+1\big), N\big(2^dK^d+1\big)\big).$$

For networks with architectures satisfying $3\log_2(4(\beta+1)(d+1)N)\log_2 n \le L \lesssim \log n$, $n^{1/(2\beta+1)} \lesssim \min_i p_i$ and $s \asymp n^{1/(2\beta+1)}\log n$, Theorem 1 yields the rate

$$R(\widehat{f}_n, f_0) \lesssim n^{-\frac{2\beta}{2\beta+1}}\log^3 n + \Delta(\widehat{f}_n, f_0),$$

and the exponent in the rate does not depend on the input dimension $d$.

**5. Suboptimality of wavelet series estimators.** As argued before, the composition assumption in (6) is very natural and generalizes many structural constraints such as additive models. In this section we show that wavelet series estimators are unable to take advantage from the underlying composition structure in the regression function and achieve in some setups much slower convergence rates.

More specifically, we consider general additive models of the form $f_0(\mathbf{x}) = h(x_1 + \cdots + x_d)$ with $h \in \mathcal{C}_1^\alpha([0,d], K)$. This can also be viewed as a special instance of the single index model, where the aim is not to estimate $h$ but $f_0$. Using (13), the prediction error of neural network reconstructions with small empirical risk and depth $L \asymp \log n$ is then bounded by $n^{-2\alpha/(2\alpha+1)}\log^3 n$. The lower bound below shows that wavelet series estimators cannot converge faster than with the rate $n^{-2\alpha/(2\alpha+d)}$. This rate can be much slower if $d$ is large. Wavelet series estimators thus suffer in this case from the curse of dimensionality while neural networks achieve fast rates.

Consider a compact wavelet basis of $L^2(\mathbb{R})$ restricted to $L^2[0,1]$, say $(\psi_{\boldsymbol{\lambda}}, \boldsymbol{\lambda} \in \boldsymbol{\Lambda})$; cf. [10]. Here, $\boldsymbol{\Lambda} = \{(j,k) : j = -1, 0, 1, \ldots; k \in I_j\}$ with $k$ ranging over the index set $I_j$, and $\psi_{-1,k} := \phi(\cdot - k)$ are the shifted scaling functions. Then, for any function $f \in L^2[0,1]^d$,

$$f(\mathbf{x}) = \sum_{(\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_d)\in\boldsymbol{\Lambda}\times\cdots\times\boldsymbol{\Lambda}} d_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f)\prod_{r=1}^d \psi_{\boldsymbol{\lambda}_r}(x_r),$$

with convergence in $L^2[0,1]$ and

$$d_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f) := \int f(\mathbf{x})\prod_{r=1}^d \psi_{\boldsymbol{\lambda}_r}(x_r)\,d\mathbf{x}$$

the wavelet coefficients.

To construct a counterexample, it is enough to consider the nonparametric regression model $Y_i = f_0(\mathbf{X}_i) + \varepsilon_i$, $i = 1, \ldots, n$ with uniform design $\mathbf{X}_i := (U_{i,1}, \ldots, U_{i,d}) \sim \mathrm{Unif}[0,1]^d$. The empirical wavelet coefficients are

$$\widehat{d}_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0) = \frac{1}{n}\sum_{i=1}^n Y_i \prod_{r=1}^d \psi_{\boldsymbol{\lambda}_r}(U_{i,r}).$$

Because of $E[\widehat{d}_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0)] = d_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0)$, this gives unbiased estimators for the wavelet coefficients. By the law of total variance,

$$\begin{aligned}\mathrm{Var}(\widehat{d}_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0)) &= \frac{1}{n}\mathrm{Var}\left(Y_1\prod_{r=1}^d \psi_{\boldsymbol{\lambda}_r}(U_{1,r})\right)\\ &\ge \frac{1}{n}E\left[\mathrm{Var}\left(Y_1\prod_{r=1}^d \psi_{\boldsymbol{\lambda}_r}(U_{1,r})|U_{1,1},,\ldots,U_{1,d}\right)\right]\\ &= \frac{1}{n}.\end{aligned}$$

For the lower bounds we may assume that the smoothness indices are known. For estimation we can truncate the series expansion on a resolution level that balances squared bias and variance of the total estimator. More generally, we study estimators of the form

$$\widehat{f}_n(\mathbf{x}) = \sum_{(\boldsymbol{\lambda}_1,\dots,\boldsymbol{\lambda}_d)\in I} \widehat{d}_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0) \prod_{r=1}^{d} \psi_{\boldsymbol{\lambda}_r}(x_r) \tag{15}$$

for an arbitrary subset $I \subset \Lambda \times \cdots \times \Lambda$. Using that, the design is uniform,

$$\begin{aligned} R(\widehat{f}_n, f_0) &= \sum_{(\boldsymbol{\lambda}_1,\dots,\boldsymbol{\lambda}_d)\in I} E\big[\big(\widehat{d}_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0) - d_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0)\big)^2\big] + \sum_{(\boldsymbol{\lambda}_1,\dots,\boldsymbol{\lambda}_d)\in I^c} d_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0)^2 \\ &\geq \sum_{(\boldsymbol{\lambda}_1,\dots,\boldsymbol{\lambda}_d)\in\Lambda\times\cdots\times\Lambda} \frac{1}{n} \wedge d_{\boldsymbol{\lambda}_1\cdots\boldsymbol{\lambda}_d}(f_0)^2. \end{aligned} \tag{16}$$

By construction, $\psi \in L^2(\mathbb{R})$ has compact support, We can, therefore, without loss of generality assume that $\psi$ is zero outside of $[0, 2^q]$ for some integer $q > 0$.

LEMMA 2. *Let $q$ be as above and set $\nu := \lceil \log_2 d \rceil + 1$. For any $0 < \alpha \leq 1$ and any $K > 0$, there exists a nonzero constant $c(\psi, d)$ only depending on $d$ and properties of the wavelet function $\psi$ such that, for any $j$, we can find a function $f_{j,\alpha}(\mathbf{x}) = h_{j,\alpha}(x_1 + \cdots + x_d)$ with $h_{j,\alpha} \in \mathcal{C}_1^\alpha([0, d], K)$ satisfying*

$$d_{(j,2^{q+\nu}p_1)\cdots(j,2^{q+\nu}p_d)}(f_{j,\alpha}) = c(\psi, d) K 2^{-\frac{j}{2}(2\alpha+d)}$$

*for all $p_1, \dots, p_d \in \{0, 1, \dots, 2^{j-q-\nu} - 1\}$.*

THEOREM 4. *If $\widehat{f}_n$ denotes the wavelet estimator (15) for a compactly supported wavelet $\psi$ and an arbitrary index set $I$, then, for any $0 < \alpha \leq 1$ and any Hölder radius $K > 0$,*

$$\sup_{f_0(\mathbf{x}) = h(\sum_{r=1}^d x_r), h\in\mathcal{C}_1^\alpha([0,d],K)} R(\widehat{f}_n, f_0) \gtrsim n^{-\frac{2\alpha}{2\alpha+d}}.$$

A close inspection of the proof shows that the theorem even holds for $0 < \alpha \leq r$ with $r$ the smallest positive integer for which $\int x^r \psi(x)\, dx \neq 0$.

**6. A brief summary of related statistical theory for neural networks.** This section is intended as a condensed overview on related literature summarizing main proving strategies for bounds on the statistical risk. An extended summary of the work until the late 90s is given in [39]. To control the stochastic error of neural networks, bounds on the covering entropy and VC dimension can be found in the monograph [1]. A challenging part in the analysis of neural networks is the approximation theory for multivariate functions. We first recall results for shallow neural networks, that is, neural networks with one hidden layer.

*Shallow neural networks*: A shallow network with one output unit and width vector $(d, m, 1)$ can be written as

$$f_m(\mathbf{x}) = \sum_{j=1}^{m} c_j \sigma\big(\mathbf{w}_j^\top \mathbf{x} + v_j\big), \quad \mathbf{w}_j \in \mathbb{R}^d, v_j, c_j \in \mathbb{R}. \tag{17}$$

The universal approximation theorem states that a neural network with one hidden layer can approximate any continuous function $f$ arbitrarily well with respect to the uniform norm provided there are enough hidden units; cf. [11, 19, 20, 29, 46]. If $f$ has a derivative $f'$, then the derivative of the neural network also approximates $f'$. The number of required hidden units might be, however, extremely large; cf. [37] and [36]. There are several proofs for the

universal approximation theorem based on the Fourier transform, the Radon transform and the Hahn–Banach theorem [42].

The proofs can be sharpened in order to obtain rates of convergence. In [33] the convergence rate $n^{-2\beta/(2\beta+d+5)}$ is derived. Compared with the minimax estimation rate, this is suboptimal by a polynomial factor. The reason for the loss of performance with this approach is that rewriting the function as a network requires too many parameters.

In [4, 5, 23, 24] a similar strategy is used to derive the rate $C_f(d\frac{\log n}{n})^{1/2}$ for the squared $L^2$-risk, where $C_f := \int |\omega|_1 |\mathcal{F}f(\omega)|\,d\omega$ and $\mathcal{F}f$ denotes the Fourier transform of $f$. If $C_f < \infty$ and $d$ is fixed, the rate is always $n^{-1/2}$ up to logarithmic factors. Since $\sum_i \|\partial_i f\|_\infty \le C_f$, this means that $C_f < \infty$ can only hold if $f$ has Hölder smoothness at least one. This rate is difficult to compare with the standard nonparametric rates except for the special case $d = 1$, where the rate is suboptimal compared with the minimax rate $n^{-2/(2+d)}$ for $d$-variate functions with smoothness one. More interestingly, the rate $C_f(d\frac{\log n}{n})^{1/2}$ shows that neural networks can converge fast if the underlying function satisfies some additional structural constraint. The same rate can also be obtained by a Fourier series estimator; see [8], Section 1.7. In a similar fashion, Bach [2] studies abstract function spaces on which shallow networks achieve fast convergence rates.

*Results for multilayer neural networks*: In [34] it is shown how to approximate a polytope by a neural network with two hidden layers. Based on this result, [25] uses two-layer neural networks with sigmoidal activation function and achieves the nonparametric rate $n^{-2\beta/(2\beta+d)}$ up to $\log n$-factors for $\beta \le 1$. This is extended in [26] to a composition assumption and further generalized to $\beta > 1$ in the recent article [6]. Unfortunately, the result requires that the activation function is at least as smooth as the signal (cf. Theorem 1 in [6]) and, therefore, rules out the ReLU activation function.

The activation function $\sigma(x) = x^2$ is not of practical relevance but has some interesting theory. Indeed, with one hidden layer we can generate quadratic polynomials and with $L$ hidden layers polynomials of degree $2^L$. For this activation function the role of the network depth is the polynomial degree, and we can use standard results to approximate functions in common function classes. A natural generalization is the class of activation functions satisfying $\lim_{x\to-\infty} x^{-k}\sigma(x) = 0$ and $\lim_{x\to+\infty} x^{-k}\sigma(x) = 1$.

If the growth is at least quadratic ($k \ge 2$), the approximation theory has been derived in [34] for deep networks with a number of layers scaling with $\log d$. The same class has also been considered recently in [7]. For the approximations to work, the assumption $k \ge 2$ is crucial, and the same approach does not generalize to the ReLU activation function, which satisfies the growth condition with $k = 1$, and always produces functions that are piecewise linear in the input.

Approximation theory for the ReLU activation function has been only recently developed in [30, 47, 49, 52]. The key observation is that there are specific deep networks with few units which approximate the square function well. In particular, the function approximation presented in [52] is essential for our approach, and we use a similar strategy to construct networks that are close to a given function. We are, however, interested in a somehow different question. Instead of deriving existence of a network architecture with good approximation properties, we show that for any network architecture satisfying the conditions of Theorem 1, good approximation rates are obtainable. An additional difficulty in our approach is the boundedness of the network parameters.

## 7. Proofs.

7.1. *Embedding properties of network function classes.* For the approximation of a function by a network, we first construct smaller networks computing simpler objects. Let

$\mathbf{p} = (p_0, \dots, p_{L+1})$ and $\mathbf{p}' = (p'_0, \dots, p'_{L+1})$. To combine networks, we make frequent use of the following rules.

*Enlarging*: $\mathcal{F}(L, \mathbf{p}, s) \subseteq \mathcal{F}(L, \mathbf{q}, s')$ whenever $\mathbf{p} \le \mathbf{q}$ componentwise and $s \le s'$.

*Composition*: Suppose that $f \in \mathcal{F}(L, \mathbf{p})$ and $g \in \mathcal{F}(L', \mathbf{p}')$ with $p_{L+1} = p'_0$. For a vector $\mathbf{v} \in \mathbb{R}^{p_{L+1}}$, we define the composed network $g \circ \sigma_{\mathbf{v}}(f)$ which is in the space $\mathcal{F}(L + L' + 1, (\mathbf{p}, p'_1, \dots, p'_{L'+1}))$. In most of the cases that we consider, the output of the first network is nonnegative, and the shift vector $\mathbf{v}$ will be taken to be zero.

*Additional layers/depth synchronization*: To synchronize the number of hidden layers for two networks, we can add additional layers with identity weight matrix, such that

$$\mathcal{F}(L, \mathbf{p}, s) \subset \mathcal{F}\big(L + q, (\underbrace{p_0, \dots, p_0}_{q \text{ times}}, \mathbf{p}), s + q p_0\big). \tag{18}$$

*Parallelization*: Suppose that $f, g$ are two networks with the same number of hidden layers and the same input dimension, that is, $f \in \mathcal{F}(L, \mathbf{p})$ and $g \in \mathcal{F}(L, \mathbf{p}')$ with $p_0 = p'_0$. The parallelized network $(f, g)$ computes $f$ and $g$ simultaneously in a joint network in the class $\mathcal{F}(L, (p_0, p_1 + p'_1, \dots, p_{L+1} + p'_{L+1}))$.

*Removal of inactive nodes*: We have

$$\mathcal{F}(L, \mathbf{p}, s) = \mathcal{F}\big(L, (p_0, p_1 \wedge s, p_2 \wedge s, \dots, p_L \wedge s, p_{L+1}), s\big). \tag{19}$$

To see this, let $f(\mathbf{x}) = W_L \sigma_{\mathbf{v}_L} W_{L-1} \dots \sigma_{\mathbf{v}_1} W_0 \mathbf{x} \in \mathcal{F}(L, \mathbf{p}, s)$. If all entries of the $j$th column of $W_i$ are zero, then we can remove this column together with the $j$th row in $W_{i-1}$ and the $j$th entry of $\mathbf{v}_i$ without changing the function. This shows then that $f \in \mathcal{F}(L, (p_0, \dots, p_{i-1}, p_i - 1, p_{i+1}, \dots, p_{L+1}), s)$. Because there are $s$ active parameters, we can iterate this procedure at least $p_i - s$ times for any $i = 1, \dots, L$. This proves $f \in \mathcal{F}(L, (p_0, p_1 \wedge s, p_2 \wedge s, \dots, p_L \wedge s, p_{L+1}), s)$.

We frequently make use of the fact that, for a fully connected network in $\mathcal{F}(L, \mathbf{p})$, there are $\sum_{\ell=0}^{L} p_\ell p_{\ell+1}$ weight matrix parameters and $\sum_{\ell=1}^{L} p_\ell$ network parameters coming from the shift vectors. The total number of parameters is thus

$$\sum_{\ell=0}^{L} (p_\ell + 1) p_{\ell+1} - p_{L+1}. \tag{20}$$

THEOREM 5. *For any function $f \in \mathcal{C}_r^\beta([0,1]^r, K)$ and any integers $m \ge 1$ and $N \ge (\beta+1)^r \vee (K+1)e^r$, there exists a network*

$$\widetilde{f} \in \mathcal{F}\big(L, (r, 6(r + \lceil\beta\rceil)N, \dots, 6(r + \lceil\beta\rceil)N, 1), s, \infty\big)$$

*with depth*

$$L = 8 + (m+5)\big(1 + \lceil \log_2(r \vee \beta) \rceil\big)$$

*and number of parameters*

$$s \le 141(r + \beta + 1)^{3+r} N(m+6),$$

*such that*

$$\|\widetilde{f} - f\|_{L^\infty([0,1]^r)} \le (2K+1)\big(1 + r^2 + \beta^2\big)6^r N 2^{-m} + K 3^\beta N^{-\frac{\beta}{r}}.$$

The proof of the theorem is given in the Supplementary Material [43]. The idea is to first build networks that, for given input $(x, y)$, approximately compute the product $xy$. We then split the input space into small hyper-cubes and construct a network that approximates a local Taylor expansion on each of these hypercubes.

Based on Theorem 5, we can now construct a network that approximates $f = g_q \circ \cdots \circ g_0$. In a first step, we show that $f$ can always be written as composition of functions defined on hypercubes $[0,1]^{t_i}$. As in the previous theorem, let $g_{ij} \in \mathcal{C}_{t_i}^{\beta_i}([a_i,b_i]^{t_i}, K_i)$, and assume that $K_i \geq 1$. For $i = 1, \ldots, q-1$, define

$$h_0 := \frac{g_0}{2K_0} + \frac{1}{2}, \qquad h_i := \frac{g_i(2K_{i-1} \cdot - K_{i-1})}{2K_i} + \frac{1}{2}, \qquad h_q = g_q(2K_{q-1} \cdot - K_{q-1}).$$

Here, $2K_{i-1}\mathbf{x} - K_{i-1}$ means that we transform the entries by $2K_{i-1}x_j - K_{i-1}$ for all $j$. Clearly,

$$f = g_q \circ \cdots \circ g_0 = h_q \circ \cdots \circ h_0. \tag{21}$$

Using the definition of the Hölder balls $C_r^{\beta}(D,K)$, it follows that $h_{0j}$ takes values in $[0,1]$, $h_{0j} \in \mathcal{C}_{t_0}^{\beta_0}([0,1]^{t_0}, 1)$, $h_{ij} \in \mathcal{C}_{t_i}^{\beta_i}([0,1]^{t_i}, (2K_{i-1})^{\beta_i})$ for $i = 1, \ldots, q-1$ and $h_{qj} \in \mathcal{C}_{t_q}^{\beta_q}([0,1]^{t_q}, K_q(2K_{q-1})^{\beta_q})$. Without loss of generality, we can always assume that the radii of the Hölder balls are at least one, that is, $K_i \geq 1$.

LEMMA 3. *Let $h_{ij}$ be as above with $K_i \geq 1$. Then, for any functions $\widetilde{h}_i = (\widetilde{h}_{ij})_j^\top$ with $\widetilde{h}_{ij} : [0,1]^{t_i} \to [0,1]$,*

$$\begin{aligned}&\|h_q \circ \cdots \circ h_0 - \widetilde{h}_q \circ \cdots \circ \widetilde{h}_0\|_{L^\infty[0,1]^d} \\ &\quad \leq K_q \prod_{\ell=0}^{q-1} (2K_\ell)^{\beta_{\ell+1}} \sum_{i=0}^{q} \big\| |h_i - \widetilde{h}_i|_\infty \big\|_{L^\infty[0,1]^{d_i}}^{\prod_{\ell=i+1}^q \beta_\ell \wedge 1}.\end{aligned}$$

PROOF. Define $H_i = h_i \circ \cdots \circ h_0$ and $\widetilde{H}_i = \widetilde{h}_i \circ \cdots \circ \widetilde{h}_0$. If $Q_i$ is an upper bound for the Hölder seminorm of $h_{ij}$, $j = 1, \ldots, d_{i+1}$, we find, using triangle inequality,

$$\begin{aligned}&\big|H_i(\mathbf{x}) - \widetilde{H}_i(\mathbf{x})\big|_\infty \\ &\quad \leq \big|h_i \circ H_{i-1}(\mathbf{x}) - h_i \circ \widetilde{H}_{i-1}(\mathbf{x})\big|_\infty + \big|h_i \circ \widetilde{H}_{i-1}(\mathbf{x}) - \widetilde{h}_i \circ \widetilde{H}_{i-1}(\mathbf{x})\big|_\infty \\ &\quad \leq Q_i \big|H_{i-1}(\mathbf{x}) - \widetilde{H}_{i-1}(\mathbf{x})\big|_\infty^{\beta_i \wedge 1} + \big\| |h_i - \widetilde{h}_i|_\infty \big\|_{L^\infty[0,1]^{d_i}}.\end{aligned}$$

Together with the inequality $(y+z)^\alpha \leq y^\alpha + z^\alpha$, which holds for all $y, z \geq 0$ and all $\alpha \in [0,1]$, the result follows. □

PROOF OF THEOREM 1. It is enough to prove the result for all sufficiently large $n$. Throughout the proof $C'$ is a constant only depending on $(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, F)$ that may change from line to line. Combining Theorem 2 with the assumed bounds on the depth $L$ and the network sparsity $s$, it follows for $n \geq 3$:

$$\begin{aligned}&\frac{1}{4}\Delta_n(\widehat{f}_n, f_0) - C'\phi_n L \log^2 n \\ &\quad \leq R(\widehat{f}, f_0) \leq 4 \inf_{f^* \in \mathcal{F}(L,\mathbf{p},s,F)} \|f^* - f_0\|_\infty^2 + 4\Delta_n(\widehat{f}_n, f_0) + C'\phi_n L \log^2 n,\end{aligned} \tag{22}$$

where we used $\varepsilon = 1/2$ for the lower bound and $\varepsilon = 1$ for the upper bound. Taking $C = 8C'$, we find that $\frac{1}{8}\Delta_n(\widehat{f}_n, f_0) \leq R(\widehat{f}, f_0)$ whenever $\Delta_n(\widehat{f}_n, f_0) \geq C\phi_n L \log^2 n$. This proves the lower bound in (9).

To derive the upper bounds in (8) and (9), we need to bound the approximation error. To do this, we rewrite the regression function $f_0$ as in (21), that is, $f_0 = h_q \circ \ldots h_0$ with $h_i = (h_{ij})_j^\top$, $h_{ij}$ defined on $[0,1]^{t_i}$, and for any $i < q$, $h_{ij}$ mapping to $[0,1]$.

We apply Theorem 5 to each function $h_{ij}$ separately. Take $m = \lceil \log_2 n \rceil$, and let $L_i' := 8 + (\lceil \log_2 n \rceil + 5)(1 + \lceil \log_2(t_i \vee \beta_i) \rceil)$. This means there exists a network $\widetilde{h}_{ij} \in \mathcal{F}(L_i', (t_i, 6(t_i + \lceil \beta_i \rceil)N, \dots, 6(t_i + \lceil \beta_i \rceil)N, 1), s_i)$ with $s_i \le 141(t_i + \beta_i + 1)^{3+t_i} N(\lceil \log_2 n \rceil + 6)$, such that

$$\|\widetilde{h}_{ij} - h_{ij}\|_{L^\infty([0,1]^{t_i})} \le (2Q_i + 1)\big(1 + t_i^2 + \beta_i^2\big)6^{t_i} N n^{-1} + Q_i 3^{\beta_i} N^{-\frac{\beta_i}{t_i}}, \tag{23}$$

where $Q_i$ is any upper bound of the Hölder norms of $h_{ij}$. If $i < q$, then we apply to the output the two additional layers $1 - (1 - x)_+$. This requires four additional parameters. Call the resulting network $h_{ij}^* \in \mathcal{F}(L_i' + 2, (t_i, 6(t_i + \lceil \beta_i \rceil)N, \dots, 6(t_i + \lceil \beta_i \rceil)N, 1), s_i + 4)$, and observe that $\sigma(h_{ij}^*) = (\widetilde{h}_{ij}(x) \vee 0) \wedge 1$. Since $h_{ij}(\mathbf{x}) \in [0, 1]$, we must have

$$\big\|\sigma\big(h_{ij}^*\big) - h_{ij}\big\|_{L^\infty([0,1]^r)} \le \|\widetilde{h}_{ij} - h_{ij}\|_{L^\infty([0,1]^r)}. \tag{24}$$

If the networks $h_{ij}^*$ are computed in parallel, $h_i^* = (h_{ij}^*)_{j=1,\dots,d_{i+1}}$ lies in the class

$$\mathcal{F}\big(L_i' + 2, (d_i, 6r_i N, \dots, 6r_i N, d_{i+1}), d_{i+1}(s_i + 4)\big),$$

with $r_i := \max_i d_{i+1}(t_i + \lceil \beta_i \rceil)$. Finally, we construct the composite network $f^* = \widetilde{h}_{q1} \circ \sigma(h_{q-1}^*) \circ \cdots \circ \sigma(h_0^*)$, which by the composition rule in Section 7.1 can be realized in the class

$$\mathcal{F}\left(E, (d, 6r_i N, \dots, 6r_i N, 1), \sum_{i=0}^{q} d_{i+1}(s_i + 4)\right), \tag{25}$$

with $E := 3(q-1) + \sum_{i=0}^q L_i'$. Observe that there is an $A_n$ that is bounded in $n$ such that $E = A_n + \log_2 n(\sum_{i=0}^q \lceil \log_2(t_i \vee \beta_i) \rceil + 1)$. Using that $\lceil x \rceil < x + 1$, $E \le \sum_{i=0}^q (\log_2(4) + \log_2(t_i \vee \beta_i)) \log_2 n \le L$ for all sufficiently large $n$. By (18) and for sufficiently large $n$, the space (25) can be embedded into $\mathcal{F}(L, \mathbf{p}, s)$ with $L, \mathbf{p}, s$ satisfying the assumptions of the theorem by choosing $N = \lceil c \max_{i=0,\dots,q} n^{\frac{t_i}{2\beta_i^* + t_i}} \rceil$ for a sufficiently small constant $c > 0$ only depending on $q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}$. Combining Lemma 3 with (23) and (24),

$$\inf_{f^* \in \mathcal{F}(L,\mathbf{p},s)} \big\|f^* - f_0\big\|_\infty^2 \le C' \max_{i=0,\dots,q} N^{-\frac{2\beta_i^*}{t_i}} \le C' \max_{i=0,\dots,q} c^{-\frac{2\beta_i^*}{t_i}} n^{-\frac{2\beta_i^*}{2\beta_i^* + t_i}}. \tag{26}$$

For the approximation error in (22) we need to find a network function that is bounded in sup-norm by $F$. By the previous inequality there exists a sequence $(\widetilde{f}_n)_n$ such that for all sufficiently large $n$, $\widetilde{f}_n \in \mathcal{F}(L, \mathbf{p}, s)$ and $\|\widetilde{f}_n - f_0\|_\infty^2 \le 2C \max_{i=0,\dots,q} c^{-2\beta_i^*/t_i} n^{-(2\beta_i^*)/(2\beta_i^* + t_i)}$. Define $f_n^* = \widetilde{f}_n(\|f_0\|_\infty / \|\widetilde{f}_n\|_\infty \wedge 1)$. Then, $\|f_n^*\|_\infty \le \|f_0\|_\infty = \|g_q\|_\infty \le K \le F$, where the last inequality follows from assumption $(i)$. Moreover, $f_n^* \in \mathcal{F}(L, \mathbf{p}, s, F)$. Writing $f_n^* - f_0 = (f_n^* - \widetilde{f}_n) + (\widetilde{f}_n - f_0)$, we obtain $\|f_n^* - f_0\|_\infty \le 2\|\widetilde{f}_n - f_0\|_\infty$. This shows that (26) also holds (with constants multiplied by 8) if the infimum is taken over the smaller space $\mathcal{F}(L, \mathbf{p}, s, F)$. Together with (22), the upper bounds in (8) and (9) follow for any constant $C$. This completes the proof. □

7.2. *Proof of Theorem 2.* Several oracle inequalities for the least-squares estimator are known; cf. [12, 15, 16, 27, 32]. The common assumption of bounded response variables is, however, violated in the nonparametric regression model with Gaussian measurement noise. Additionally, we provide also a lower bound of the risk and give a proof that can be easily generalized to arbitrary noise distributions. Let $\mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_\infty)$ be the covering number, that is, the minimal number of $\|\cdot\|_\infty$-balls with radius $\delta$ that covers $\mathcal{F}$ (the centers do not need to be in $\mathcal{F}$).

LEMMA 4. *Consider the $d$-variate nonparametric regression model (1) with unknown regression function $f_0$. Let $\widehat{f}$ be any estimator taking values in $\mathcal{F}$. Define*

$$\Delta_n := \Delta_n(\widehat{f}, f_0, \mathcal{F}) := E_{f_0}\left[\frac{1}{n}\sum_{i=1}^n (Y_i - \widehat{f}(\mathbf{X}_i))^2 - \inf_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n (Y_i - f(\mathbf{X}_i))^2\right],$$

*and assume $\{f_0\} \cup \mathcal{F} \subset \{f : [0,1]^d \to [-F, F]\}$ for some $F \geq 1$. If $\mathcal{N}_n := \mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_\infty) \geq 3$, then,*

$$(1-\varepsilon)^2\Delta_n - F^2\frac{18\log\mathcal{N}_n + 76}{n\varepsilon} - 38\delta F$$
$$\leq R(\widehat{f}, f_0) \leq (1+\varepsilon)^2\left[\inf_{f\in\mathcal{F}} E\big[(f(\mathbf{X}) - f_0(\mathbf{X}))^2\big] + F^2\frac{18\log\mathcal{N}_n + 72}{n\varepsilon} + 32\delta F + \Delta_n\right],$$

*for all $\delta, \varepsilon \in (0,1]$.*

The proof of the lemma can be found in the Supplementary Material [43]. Next, we prove a covering entropy bound. Recall the definition of the network function class $\mathcal{F}(L, \mathbf{p}, s, F)$ in (4).

LEMMA 5. *If $V := \prod_{\ell=0}^{L+1}(p_\ell + 1)$, then, for any $\delta > 0$,*

$$\log\mathcal{N}\big(\delta, \mathcal{F}(L, \mathbf{p}, s, \infty), \|\cdot\|_\infty\big) \leq (s+1)\log\big(2\delta^{-1}(L+1)V^2\big).$$

For a proof, see the Supplementary Material [43]. A related result is Theorem 14.5 in [1].

REMARK 1. Identity (19) applied to Lemma 5 yields

$$\log\mathcal{N}\big(\delta, \mathcal{F}(L, \mathbf{p}, s, \infty), \|\cdot\|_\infty\big) \leq (s+1)\log\big(2^{2L+5}\delta^{-1}(L+1)p_0^2 p_{L+1}^2 s^{2L}\big).$$

PROOF OF THEOREM 2. The assertion follows from Lemma 5 with $\delta = 1/n$, Lemma 4 and Remark 1 since $F \geq 1$. □

7.3. *Proof of Theorem 3.* Throughout this proof, $\|\cdot\|_2 = \|\cdot\|_{L^2[0,1]^d}$. By assumption there exist positive $\gamma \leq \Gamma$ such that the Lebesgue density of $\mathbf{X}$ is lower bounded by $\gamma$ and upper bounded by $\Gamma$ on $[0,1]^d$. For such design, $R(\widehat{f}_n, f_0) \geq \gamma\|\widehat{f}_n - f_0\|_2^2$. Denote by $P_f$ the law of the data in the nonparametric regression model (1). For the Kullback–Leibler divergence we have $\mathrm{KL}(P_f, P_g) = nE[(f(\mathbf{X}_1) - g(\mathbf{X}_1))^2] \leq \Gamma n\|f - g\|_2^2$. Theorem 2.7 in [50] states that if for some $M \geq 1$ and $\kappa > 0$, $f_{(0)}, \ldots, f_{(M)} \in \mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)$ are such that:

(i) $\|f_{(j)} - f_{(k)}\|_2 \geq \kappa\sqrt{\phi_n}$ for all $0 \leq j < k \leq M$,
(ii) $n\sum_{j=1}^M \|f_{(j)} - f_{(0)}\|_2^2 \leq M\log(M)/(9\Gamma)$,

then there exists a positive constant $c = c(\kappa, \gamma)$, such that

$$\inf_{\widehat{f}_n}\sup_{f_0\in\mathcal{G}(q,\mathbf{d},\mathbf{t},\boldsymbol{\beta},K)} R(\widehat{f}_n, f_0) \geq c\phi_n.$$

In a next step we construct functions $f_{(0)}, \ldots, f_{(M)} \in \mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)$ satisfying (i) and (ii). Define $i^* \in \arg\min_{i=0,\ldots,q} \beta_i^*/(2\beta_i^* + t_i)$. The index $i^*$ determines the estimation rate in the sense that $\phi_n = n^{-2\beta_{i^*}^*/(2\beta_{i^*}^* + t_{i^*})}$. For convenience, we write $\beta^* := \beta_{i^*}$, $\beta^{**} := \beta_{i^*}^*$ and $t^* := t_{i^*}$. One should notice the difference between $\beta^*$ and $\beta^{**}$. Let $K \in L^2(\mathbb{R}) \cap \mathcal{C}_1^{\beta^*}(\mathbb{R}, 1)$ be supported on $[0,1]$. It is easy to see that such a function $K$ exists. Furthermore, define $m_n := \lfloor \rho n^{1/(2\beta^{**}+t^*)} \rfloor$ and $h_n := 1/m_n$ where the constant $\rho$ is chosen such that

$nh_n^{2\beta^*+t^*} \le 1/(72\Gamma\|K^B\|_2^{2t^*})$ with $B := \prod_{\ell=i^*+1}^{q}(\beta_\ell \wedge 1)$. For any $\mathbf{u} = (u_1, \dots, u_{t^*}) \in \mathcal{U}_n := \{(u_1, \dots, u_{t^*}) : u_i \in \{0, h_n, 2h_n, \dots, (m_n - 1)h_n\}$, define

$$\psi_{\mathbf{u}}(x_1, \dots, x_{t^*}) := h_n^{\beta^*} \prod_{j=1}^{t^*} K\Big(\frac{x_j - u_j}{h_n}\Big).$$

For $\boldsymbol{\alpha}$ with $|\boldsymbol{\alpha}| < \beta^*$, we have $\|\partial^{\boldsymbol{\alpha}}\psi_{\mathbf{u}}\|_\infty \le 1$ using $K \in \mathcal{C}_1^{\beta^*}(\mathbb{R}, 1)$. For $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{t^*})$ with $|\boldsymbol{\alpha}| = \lfloor\beta^*\rfloor$, triangle inequality and $K \in \mathcal{C}_1^{\beta^*}(\mathbb{R}, 1)$ gives

$$h_n^{\beta^*-\lfloor\beta^*\rfloor} \frac{|\prod_{j=1}^{t^*} K^{(\alpha_j)}(\frac{x_j-u_j}{h_n}) - \prod_{j=1}^{t^*} K^{(\alpha_j)}(\frac{y_j-u_j}{h_n})|}{\max_i |x_i - y_i|^{\beta^*-\lfloor\beta^*\rfloor}} \le t^*.$$

Hence, $\psi_{\mathbf{u}} \in \mathcal{C}_{t^*}^{\beta^*}([0,1]^{t^*}, (\beta^*)^{t^*} t^*)$. For a vector $\mathbf{w} = (w_{\mathbf{u}})_{\mathbf{u}\in\mathcal{U}_n} \in \{0,1\}^{|\mathcal{U}_n|}$, define

$$\phi_{\mathbf{w}} = \sum_{\mathbf{u}\in\mathcal{U}_n} w_{\mathbf{u}}\psi_{\mathbf{u}}.$$

By construction, $\psi_{\mathbf{u}}$ and $\psi_{\mathbf{u}'}$, $\mathbf{u}, \mathbf{u}' \in \mathcal{U}_n$, $\mathbf{u} \ne \mathbf{u}'$ have disjoint support. As a consequence $\phi_{\mathbf{w}} \in \mathcal{C}_{t^*}^{\beta^*}([0,1]^{t^*}, 2(\beta^*)^{t^*} t^*)$.

For $i < i^*$, let $g_i(\mathbf{x}) := (x_1, \dots, x_{d_i})^\top$. For $i = i^*$, define $g_{i^*,\mathbf{w}}(\mathbf{x}) = (\phi_{\mathbf{w}}(x_1, \dots, x_{t_{i^*}}), 0, \dots, 0)^\top$. For $i > i^*$, set $g_i(\mathbf{x}) := (x_1^{\beta_i\wedge 1}, 0, \dots, 0)^\top$. Recall that $B = \prod_{\ell=i^*+1}^{q}(\beta_\ell \wedge 1)$. We will frequently use that $\beta^{**} = \beta^* B$. Because of $t_i \le \min(d_0, \dots, d_{i-1})$ and the disjoint supports of the $\psi_{\mathbf{u}}$,

$$\begin{aligned} f_{\mathbf{w}}(\mathbf{x}) &= g_q \circ \cdots \circ g_{i^*+1} \circ g_{i*,\mathbf{w}} \circ g_{i^*-1} \circ \cdots \circ g_0(\mathbf{x}) \\ &= \phi_{\mathbf{w}}(x_1, \dots, x_{t_{i^*}})^B \\ &= \sum_{\mathbf{u}\in\mathcal{U}_n} w_{\mathbf{u}}\psi_{\mathbf{u}}(x_1, \dots, x_{t_{i^*}})^B \end{aligned}$$

and $f_{\mathbf{w}} \in \mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\beta}, K)$, provided $K$ is taken sufficiently large.

For all $\mathbf{u}$, $\|\psi_{\mathbf{u}}\|_2^2 = h_n^{2\beta^{**}+t^*}\|K^B\|_2^{2t^*}$. If $\mathrm{Ham}(\mathbf{w}, \mathbf{w}') = \sum_{\mathbf{u}\in\mathcal{U}_n} \mathbf{1}(w_{\mathbf{u}} \ne w_{\mathbf{u}'})$ denotes the Hamming distance, we find

$$\|f_{\mathbf{w}} - f_{\mathbf{w}'}\|_2^2 = \mathrm{Ham}(\mathbf{w}, \mathbf{w}')h_n^{2\beta^{**}+t^*}\|K^B\|_2^{2t^*}.$$

By the Varshamov–Gilbert bound ([50], Lemma 2.9) and because of $m_n^{t^*} = |\mathcal{U}_n|$, we conclude that there exists a subset $\mathcal{W} \subset \{0,1\}^{m_n^{t^*}}$ of cardinality $|\mathcal{W}| \ge 2^{m_n^{t^*}/8}$, such that $\mathrm{Ham}(\mathbf{w}, \mathbf{w}') \ge m_n^{t^*}/8$ for all $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$, $\mathbf{w} \ne \mathbf{w}'$. This implies that, for $\kappa = \|K^B\|_2^{t^*}/(\sqrt{8}\rho^{\beta^{**}})$,

$$\|f_{\mathbf{w}} - f_{\mathbf{w}'}\|_2^2 \ge \frac{1}{8}h_n^{2\beta^{**}}\|K^B\|_2^{2t^*} \ge \kappa^2\phi_n \quad \text{for all } \mathbf{w}, \mathbf{w}' \in \mathcal{W}, \mathbf{w} \ne \mathbf{w}'.$$

Using the definition of $h_n$ and $\rho$,

$$n\|f_{\mathbf{w}} - f_{\mathbf{w}'}\|_2^2 \le nm_n^{t^*}h_n^{2\beta^{**}+t^*}\|K^B\|_2^{2t^*} \le \frac{m_n^{t^*}}{72\Gamma} \le \frac{\log|\mathcal{W}|}{9\Gamma}.$$

This shows that the functions $f_{\mathbf{w}}$ with $\mathbf{w} \in \mathcal{W}$ satisfy (i) and (ii). The assertion follows. □

7.4. *Proof of Lemma 1.* We will choose $c_2 \le 1$. Since $\|f_0\|_\infty \le K$, it is therefore enough to consider the infimum over $\mathcal{F}(L, \mathbf{p}, s, F)$ with $F = K+1$. Let $\widetilde{f}_n$ be an empirical risk minimizer. Recall that $\Delta_n(\widetilde{f}_n, f_0) = 0$. Because of the minimax lower bound in Theorem 3, there exists a constant $c_3$ such that $c_3 n^{-2\beta/(2\beta+d)} \le \sup_{f_0 \in \mathcal{C}_1^\beta([0,1]^d, K)} R(\widetilde{f}_n, f_0)$ for all sufficiently large $n$. Because of $p_0 = d$ and $p_{L+1} = 1$, Theorem 2 yields

$$\begin{aligned} c_3 n^{-2\beta/(2\beta+d)} &\le \sup_{f_0 \in \mathcal{C}_d^\beta([0,1]^d, K)} R(\widetilde{f}_n, f_0) \\ &\le 4 \sup_{f_0 \in \mathcal{C}_d^\beta([0,1]^d, K)} \inf_{f \in \mathcal{F}(L, \mathbf{p}, s, K+1)} \|f - f_0\|_\infty^2 \\ &\quad + C(K+1)^2 \frac{(s+1)\log(n(s+1)^L d)}{n} \end{aligned}$$

for some constant $C$. Given $\varepsilon$, set $n_\varepsilon := \lfloor (\sqrt{8}\varepsilon/\sqrt{c_3})^{-(2\beta+d)/\beta} \rfloor$. Observe that for $\varepsilon \le \sqrt{c_3/8}$, $n_\varepsilon^{-1} \le 2(\sqrt{8}\varepsilon/\sqrt{c_3})^{(2\beta+d)/\beta}$ and $8\varepsilon^2/c_3 \le n_\varepsilon^{-2\beta/(2\beta+d)}$. For sufficiently small $c_2 > 0$ and all $\varepsilon \le c_2$, we can insert $n_\varepsilon$ in the previous inequality and find

$$8\varepsilon^2 \le 4 \sup_{f_0 \in \mathcal{C}_d^\beta([0,1]^d, K)} \inf_{f \in \mathcal{F}(L, \mathbf{p}, s, K+1)} \|f - f_0\|_\infty^2 + C_1 \varepsilon^{\frac{2\beta+d}{\beta}} s\big(\log(\varepsilon^{-1} s^L) + C_2\big)$$

for constants $C_1, C_2$ depending on $K, \beta$ and $d$. The result follows using the condition $s \le c_1 \varepsilon^{-d/\beta}/(L\log(1/\varepsilon))$ and choosing $c_1$ small enough. □

7.5. *Proofs for Section 5.* PROOF OF LEMMA 2. Denote by $r$ the smallest positive integer such that $\mu_r := \int x^r \psi(x) dx \ne 0$. Such an $r$ exists because $\{x^r : r = 0, 1, \ldots\}$ spans $L^2[0, A]$ and $\psi$ cannot be constant. If $h \in L^2(\mathbb{R})$, then we have for the wavelet coefficients

$$\int h(x_1 + \cdots + x_d) \prod_{\ell=1}^d \psi_{j,k_\ell}(x_\ell)\, d\mathbf{x} = 2^{-\frac{jd}{2}} \int_{[0,2^q]^d} h\left(2^{-j}\left(\sum_{\ell=1}^d x_\ell + k_\ell\right)\right) \prod_{\ell=1}^d \psi(x_\ell)\, d\mathbf{x}. \tag{27}$$

For a real number $u$, denote by $\{u\}$ the fractional part of $u$.

We need to study the cases $\mu_0 \ne 0$ and $\mu_0 = 0$ separately. If $\mu_0 \ne 0$, define $g(u) = r^{-1}u^r \mathbf{1}_{[0,1/2]}(u) + r^{-1}(1-u)^r \mathbf{1}_{(1/2,1]}(u)$. Notice that $g$ is Lipschitz with Lipschitz constant one. Let $h_{j,\alpha}(u) = K 2^{-j\alpha-1} g(\{2^{j-q-\nu}u\})$ with $q$ and $\nu$ as defined in the statement of the lemma. For a $T$-periodic function $u \mapsto s(u)$, the $\alpha$-Hölder seminorm for $\alpha \le 1$ can be shown to be $\sup_{u \ne v, |u-v| \le T} |s(u) - s(v)|/|u-v|^\alpha$. Since $g$ is 1-Lipschitz, we have for $u, v$ with $|u - v| \le 2^{q+\nu-j}$,

$$|h_{j,\alpha}(u) - h_{j,\alpha}(v)| \le K 2^{-j\alpha-1} 2^{j-q-\nu} |u - v| \le \frac{K}{2} |u - v|^\alpha.$$

Since $\|h_{j,\alpha}\|_\infty \le K/2$, $h_{j,\alpha} \in \mathcal{C}_1^\alpha([0,d], K)$. Let $f_{j,\alpha}(\mathbf{x}) = h_{j,\alpha}(x_1 + \cdots + x_d)$. Recall that the support of $\psi$ is contained in $[0, 2^q]$ and $2^\nu \ge 2d$. By definition of the wavelet coefficients, equation (27), the definitions of $h_{j,\alpha}$ and using $\mu_r = \int x^r \psi(x) dx$, we find for $p_1, \ldots, p_d \in \{0, 1, \ldots, 2^{j-q-2} - 1\}$,

$$\begin{aligned} &d_{(j,2^{q+\nu}p_1)\ldots(j,2^{q+\nu}p_d)}(f_{j,\alpha}) \\ &\quad = 2^{-\frac{jd}{2}} \int_{[0,2^q]^d} h_{j,\alpha}\left(2^{-j}\left(\sum_{\ell=1}^d x_\ell + 2^{q+\nu}p_\ell\right)\right) \prod_{\ell=1}^d \psi(x_\ell)\, d\mathbf{x} \end{aligned}$$

$$= K2^{-\frac{jd}{2}-j\alpha-1}\int_{[0,2^q]^d} g\left(\left\{\frac{\sum_{\ell=1}^d x_\ell}{2^{q+\nu}}\right\}\right)\prod_{\ell=1}^d \psi(x_\ell)\,d\mathbf{x}$$

$$= r^{-1}2^{-qr-\nu r-1}K2^{-\frac{j}{2}(2\alpha+d)}\int_{[0,2^q]^d}(x_1+\cdots+x_d)^r\prod_{\ell=1}^d \psi(x_\ell)\,d\mathbf{x}$$

$$= dr^{-1}2^{-qr-\nu r-1}K\mu_0^{d-1}\mu_r 2^{-\frac{j}{2}(2\alpha+d)},$$

where we used for the last identity that by definition of $r$, $\mu_1 = \cdots = \mu_{r-1} = 0$.

In the case that $\mu_0 = 0$, we can take $g(u) = (dr)^{-1}u^{dr}\mathbf{1}_{[0,1/2]}(u) + (dr)^{-1} \times (1-u)^{dr}\mathbf{1}_{(1/2,1]}(u)$. Following the same arguments as before and using the multinomial theorem, we obtain

$$d_{(j,2^{q+\nu}p_1)\ldots(j,2^{q+\nu}p_r)}(f_{j,\alpha}) = \binom{dr}{r}\frac{1}{dr}2^{-dqr-d\nu r-1}K\mu_r^d 2^{-\frac{j}{2}(2\alpha+d)}.$$

The claim of the lemma follows. □

PROOF OF THEOREM 4. Let $c(\psi, d)$ be as in Lemma 2. Choose an integer $j^*$ such that

$$\frac{1}{n} \le c(\psi,d)^2K^2 2^{-j^*(2\alpha+d)} \le \frac{2^{2\alpha+d}}{n}.$$

This means that $2^{j^*} \ge \frac{1}{2}(c(\psi,d)^2K^2n)^{1/(2\alpha+d)}$. By Lemma 2, there exists a function $f_{j^*,\alpha}$ of the form $h(x_1+\cdots+x_d)$, $h \in \mathcal{C}_1^\alpha([0,d],K)$, such that with (16),

$$R(\widehat{f}_n, f_{j^*,\alpha}) \ge \sum_{p_1,\ldots,p_d\in\{0,1,\ldots,2^{j^*-q-\nu}-1\}}\frac{1}{n} = \frac{1}{n}2^{j^*d-qd-\nu d} \gtrsim n^{-\frac{2\alpha}{2\alpha+d}}.$$

□

**Acknowledgments.** The author is grateful for all the insights, comments and suggestions that arose from discussions on the topic with other researchers. In particular, he wants to thank the Associate Editor, two referees, Thijs Bos, Hyunwoong Chang, Konstantin Eckle, Kenji Fukumizu, Maximilian Graf, Roy Han, Masaaki Imaizumi, Michael Kohler, Matthias Löffler, Patrick Martin, Hrushikesh Mhaskar, Gerrit Oomens, Tomaso Poggio, Richard Samworth, Taiji Suzuki, Dmitry Yarotsky and Harry van Zanten.

## SUPPLEMENTARY MATERIAL

**Supplement to "Nonparametric regression using deep neural networks with ReLU activation function"** (DOI: 10.1214/19-AOS1875SUPP; .pdf). Some of the proofs are given in the supplement.

## REFERENCES

[1] ANTHONY, M. and BARTLETT, P. L. (1999). *Neural Network Learning*: *Theoretical Foundations*. Cambridge Univ. Press, Cambridge. MR1741038 https://doi.org/10.1017/CBO9780511624216

[2] BACH, F. (2017). Breaking the curse of dimensionality with convex neutral networks. *J. Mach. Learn. Res.* **18** Art. ID 19. MR3634886

[3] BARAUD, Y. and BIRGÉ, L. (2014). Estimating composite functions by model selection. *Ann. Inst. Henri Poincaré Probab. Stat.* **50** 285–314. MR3161532 https://doi.org/10.1214/12-AIHP516

[4] BARRON, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory* **39** 930–945. MR1237720 https://doi.org/10.1109/18.256500

[5] BARRON, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Mach. Learn.* **14** 115–133. https://doi.org/10.1007/BF00993164

[6] BAUER, B. and KOHLER, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Ann. Statist.* **47** 2261–2285. MR3953451 https://doi.org/10.1214/18-AOS1747
[7] BÖLCSKEI, H., GROHS, P., KUTYNIOK, G. and PETERSEN, P. (2019). Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math. Data Sci.* **1** 8–45. MR3949699 https://doi.org/10.1137/18M118709X
[8] CANDÈS, E. J. (2002). New ties between computational harmonic analysis and approximation theory. In *Approximation Theory, X (St. Louis, MO, 2001). Innov. Appl. Math.* 87–153. Vanderbilt Univ. Press, Nashville, TN. MR1924879
[9] CHOROMANSKA, A., HENAFF, M., MATHIEU, M., AROUS, G. B. and LECUN, Y. (2015). The loss surface of multilayer networks. In *Aistats. Proceedings of Machine Learning Research* **38** 192–204.
[10] COHEN, A., DAUBECHIES, I. and VIAL, P. (1993). Wavelets on the interval and fast wavelet transforms. *Appl. Comput. Harmon. Anal.* **1** 54–81. MR1256527 https://doi.org/10.1006/acha.1993.1005
[11] CYBENKO, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems* **2** 303–314. MR1015670 https://doi.org/10.1007/BF02551274
[12] GINÉ, E. and KOLTCHINSKII, V. (2006). Concentration inequalities and asymptotic results for ratio type empirical processes. *Ann. Probab.* **34** 1143–1216. MR2243881 https://doi.org/10.1214/009117906000000070
[13] GLOROT, X., BORDES, A. and BENGIO, Y. (2011). Deep sparse rectifier neural networks. In *Aistats. Proceedings of Machine Learning Research* **15** 315–323.
[14] GOODFELLOW, I., BENGIO, Y. and COURVILLE, A. (2016). *Deep Learning. Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA. MR3617773
[15] GYÖRFI, L., KOHLER, M., KRZYŻAK, A. and WALK, H. (2002). *A Distribution-Free Theory of Nonparametric Regression. Springer Series in Statistics*. Springer, New York. MR1920390 https://doi.org/10.1007/b97848
[16] HAMERS, M. and KOHLER, M. (2006). Nonasymptotic bounds on the $L_2$ error of neural network regression estimates. *Ann. Inst. Statist. Math.* **58** 131–151. MR2281209 https://doi.org/10.1007/s10463-005-0005-9
[17] HE, K. and SUN, J. (2015). Convolutional neural networks at constrained time cost. In *CVPR* 5353–5360.
[18] HE, K., ZHANG, X., REN, S. and SUN, J. (2016). Deep residual learning for image recognition. In *CVPR* 770–778. https://doi.org/10.1109/CVPR.2016.90
[19] HORNIK, K., STINCHCOMBE, M. and WHITE, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Netw.* **2** 359–366.
[20] HORNIK, K., STINCHCOMBE, M. and WHITE, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Netw.* **3** 551–560.
[21] HOROWITZ, J. L. and MAMMEN, E. (2007). Rate-optimal estimation for a general class of nonparametric regression models with unknown link functions. *Ann. Statist.* **35** 2589–2619. MR2382659 https://doi.org/10.1214/009053607000000415
[22] JUDITSKY, A. B., LEPSKI, O. V. and TSYBAKOV, A. B. (2009). Nonparametric estimation of composite functions. *Ann. Statist.* **37** 1360–1404. MR2509077 https://doi.org/10.1214/08-AOS611
[23] KLUSOWSKI, J. M. and BARRON, A. R. (2016). Risk bounds for high-dimensional ridge function combinations including neural networks. Preprint. Available at arXiv:1607.01434.
[24] KLUSOWSKI, J. M. and BARRON, A. R. (2016). Uniform approximation by neural networks activated by first and second order ridge splines. Preprint. Available at arXiv:1607.07819.
[25] KOHLER, M. and KRZYŻAK, A. (2005). Adaptive regression estimation with multilayer feedforward neural networks. *J. Nonparametr. Stat.* **17** 891–913. MR2192165 https://doi.org/10.1080/10485250500309608
[26] KOHLER, M. and KRZYŻAK, A. (2017). Nonparametric regression based on hierarchical interaction models. *IEEE Trans. Inf. Theory* **63** 1620–1630. MR3625984 https://doi.org/10.1109/TIT.2016.2634401
[27] KOLTCHINSKII, V. (2006). Local Rademacher complexities and oracle inequalities in risk minimization. *Ann. Statist.* **34** 2593–2656. MR2329442 https://doi.org/10.1214/009053606000001019
[28] KRIZHEVSKY, A., SUTSKEVER, I. and HINTON, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* 25 (F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger, eds.) 1097–1105. Curran Associates, Red Hook, NY.
[29] LESHNO, M., LIN, V. YA., PINKUS, A. and SCHOCKEN, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.* **6** 861–867.
[30] LIANG, S. and SRIKANT, R. (2017). Why deep neural networks for function approximation? In *ICLR* 2017.
[31] LIAO, Q. and POGGIO, T. (2017). Theory II: Landscape of the empirical risk in deep learning. Preprint. Available at arXiv:1703.09833.
[32] MASSART, P. (2007). *Concentration Inequalities and Model Selection. Lecture Notes in Math.* **1896**. Springer, Berlin. MR2319879

[33] McCaffrey, D. F. and Gallant, A. R. (1994). Convergence rates for single hidden layer feedforward networks. *Neural Netw.* **7** 147–158. https://doi.org/10.1016/0893-6080(94)90063-9

[34] Mhaskar, H. N. (1993). Approximation properties of a multilayered feedforward artificial neural network. *Adv. Comput. Math.* **1** 61–80. MR1230251 https://doi.org/10.1007/BF02070821

[35] Mhaskar, H. N. and Poggio, T. (2016). Deep vs. shallow networks: An approximation theory perspective. *Anal. Appl. (Singap.)* **14** 829–848. MR3564936 https://doi.org/10.1142/S0219530516400042

[36] Montúfar, G. F. (2014). Universal approximation depth and errors of narrow belief networks with discrete units. *Neural Comput.* **26** 1386–1407. MR3222078 https://doi.org/10.1162/NECO_a_00601

[37] Montúfar, G. F., Pascanu, R., Cho, K. and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems* 27 2924–2932. Curran Associates, Red Hook, NY.

[38] Pedamonti, D. (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task. Preprint. Available at arXiv:1804.02763.

[39] Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numer.* **8** 143–195. MR1819645 https://doi.org/10.1017/S0962492900002919

[40] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B. and Liao, Q. (2017). Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *Int. J. Autom. Comput.* **14** 503–519. https://doi.org/10.1007/s11633-017-1054-2

[41] Ray, K. and Schmidt-Hieber, J. (2017). A regularity class for the roots of nonnegative functions. *Ann. Mat. Pura Appl.* (4) **196** 2091–2103. MR3714756 https://doi.org/10.1007/s10231-017-0655-2

[42] Scarselli, F. and Tsoi, A. C. (1998). Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Netw.* **11** 15–37. https://doi.org/10.1016/S0893-6080(97)00097-X

[43] Schmidt-Hieber, J. (2020). Supplement to "Nonparametric regression using deep neural networks with ReLU activation function." https://doi.org/10.1214/19-AOS1875SUPP

[44] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15** 1929–1958. MR3231592

[45] Srivastava, R. K., Greff, K. and Schmidhuber, J. (2015). Highway networks. Preprint. Available at arXiv:1505.00387.

[46] Stinchcombe, M. B. (1999). Neural network approximation of continuous functionals and continuous functions on compactifications. *Neural Netw.* **12** 467–477. https://doi.org/10.1016/s0893-6080(98)00108-7

[47] Suzuki, T. (2018). Fast generalization error bound of deep learning from a kernel perspective. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. *Proceedings of Machine Learning Research* **84** 1397–1406.

[48] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. A. (2016). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *ICLR* 2016 *Workshop*.

[49] Telgarsky, M. (2016). Benefits of depth in neural networks. In 29*th Annual Conference on Learning Theory*. *Proceedings of Machine Learning Research* **49** 1517–1539.

[50] Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. *Springer Series in Statistics*. Springer, New York. MR2724359 https://doi.org/10.1007/b13794

[51] Wasserman, L. (2006). *All of Nonparametric Statistics*. *Springer Texts in Statistics*. Springer, New York. MR2172729

[52] Yarotsky, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Netw.* **94** 103–114. https://doi.org/10.1016/j.neunet.2017.07.002