On the rate of convergence of an over-parametrized deep neural network regression estimate learned by gradient descent [*]

Michael Kohler

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

October 16, 2024

Abstract

Nonparametric regression with random design is considered. The L_2 error with integration with respect to the design measure is used as the error criterion. An over-parametrized deep neural network regression estimate with logistic activation function is defined, where all weights are learned by gradient descent. It is shown that the estimate achieves a nearly optimal rate of convergence in case that the regression function is (p, C)–smooth.

AMS classification: Primary 62G08; secondary 62G20.

Key words and phrases: Deep neural networks, gradient descent, over-parametrization, rate of convergence, regression estimation.

1 Introduction

1.1 Scope of this paper

Deep learning has achieved tremendous success in applications, e.g., in image classification (cf., e.g., Krizhevsky, Sutskever and Hinton (2012)), language recognition (cf., e.g., Kim (2014)) machine translation (cf., e.g., Wu et al. (2016)), mastering of games (cf., e.g., Silver et al. (2017)) or simulation of human conversation (cf., e.g., Zong and Krishnamachari (2022)). From a theoretical point of view this great success is still a mystery. In particular, it is unclear why the use of over-parametrized deep neural networks, which have much more weights than there are data points, does not lead to an overfitting of the estimate, and why gradient descent is able to minimize the nonlinear and non-convex empirical risk in the definition of the estimates in such a way that the estimates can achieve a small risk for randomly initialized starting values for the weights of the networks. In a standard regression setting we give answers to these two questions in the special situation that we want to estimate a (p, C)–smooth regression function.

[*] Running title: *Neural network estimate learned by gradient descent*

1.2 Nonparametric regression

We study deep neural networks in the context of nonparametric regression. Here we have given an $\mathbb{R}^d \times \mathbb{R}$–valued random vector (X, Y) with $\mathbf{E}Y^2 < \infty$, and our goal is to predict the value of Y given the value of X. Let $m(x) = \mathbf{E}\{Y|X = x\}$ be the so–called regression function. Then any measurable $f : \mathbb{R}^d \to \mathbb{R}$ satisfies

$$\mathbf{E}\{|f(X) - Y|^2\} = \mathbf{E}\{|m(X) - Y|^2\} + \int |f(x) - m(x)|^2 \mathbf{P}_X(dx) \tag{1}$$

(cf., e.g., Section 1.1 in Györfi et al. (2002)), hence in view of minimizing the so-called L_2 risk (1) of f the regression function m is the optimal predictor, and the so–called L_2 error

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

describes how far the L_2 risk of a function f is away from its optimal value.

In applications typically the distribution of (X, Y) and hence also the corresponding regression function m is unknown. But often it is possible to observe data from the underlying distribution, and the task is to use this data to estimate the unknown regression function. In view of minimization of the L_2 risk of the estimate, here it is natural to use the L_2 error as an error criterion.

In order to introduce this problem formally, let (X, Y), (X_1, Y_1), $\dots$, (X_n, Y_n) be independent and identically distributed. In nonparametric regression the data set

$$\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\} \tag{3}$$

is given, and the task is to construct an estimate

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

such that its L_2 error

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

is small. A systematic introduction to nonparametric regression, its estimates and known results can be found, e.g., in Györfi et al. (2002).

1.3 Rate of convergence

Stone (1982) determined the optimal Minimax rate of convergence of the L_2 error in case of a smooth regression function. Here he considered so-called (p, C)–smooth regression functions, which are defined as follows.

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

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

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

Stone (1982) showed that in case of a (p, C)–smooth regression function the optimal Minimax rate of convergence for the expected L_2 error is

$$n^{-\frac{2p}{2p+d}}.$$

This rate suffers from the so–called curse of dimensionality: If the dimension d is large compared to the smoothness p of the regression function, the exponent will be close to zero and the rate of convergence will be rather slow. Since this rate is optimal, the only way to circumvent this is to impose additional assumptions on the structure of the regression function. For this, various assumptions exists, e.g., additive models (cf., e.g., Stone (1985)), interaction models (cf., e.g., Stone (1994)), single index models (cf., e.g., Härdle, Hall and Ichimura (1993), Härdle and Stoker (1989), Yu and Ruppert (2002) and Kong and Xia (2007)) or projection pursuit (cf, e.g., Friedman and Stuetzle (1981)), where corresponding low dimensional rates of convergence can be achieved (cf., e.g., Stone (1985, 1994) and Chapter 22 in Györfi et al. (2002)).

1.4 Least squares neural network estimates

One way to estimate a regression function is to define a function space $\mathcal{F}_n$ consisting of functions $f : \mathbb{R}^d \to \mathbb{R}$ and to use the principle of least squares to select one of its functions as the regression estimate, i.e., to define

$$m_n(\cdot) = \arg \min_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^{n} |f(X_i) - Y_i|^2.$$

In view of the L_2 error of the estimate it is important that the function space is on the one hand large enough such that a function is contained in it which approximates the unknown regression function well, and that on the other hand the function space is not too complex so that the empirical L_2 risk

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

is not too far from the L_2 risk for functions in this function space.

One possible way to define such function spaces in case of d large is to use feedforward neural networks. These function depend on an activation function $\sigma : \mathbb{R} \to \mathbb{R}$, e.g.,

$$\sigma(x) = \max\{x, 0\}$$

(so-called ReLU-activation function) or

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

(so-called logistic activation function).

The most simple form of neural networks are shallow networks, i.e., neural networks with one hidden layer, in which a simple linear combination of artifical neurons defined by applying the activation function to a linear combination of the components of the input is used to define a function $f : \mathbb{R}^d \to \mathbb{R}$ by

$$f(x) = \sum_{k=1}^{K} \alpha_k \cdot \sigma \left(\sum_{j=1}^{d} \beta_{k,j} \cdot x^{(j)} + \beta_{k,0} \right) + \alpha_0. \tag{4}$$

Here $K \in \mathbb{N}$ is the number of neurons, and the weights $\alpha_k \in \mathbb{R}$ ($k = 0, \dots, K$), $\beta_{k,j} \in \mathbb{R}$ ($k = 1, \dots, K, j = 0, \dots, d$) are chosen by the principle of least squares.

The rate of convergence of shallow neural networks regression estimates has been analyzed in Barron (1994) and McCaffrey and Gallant (1994). Barron (1994) proved a dimensionless rate of $n^{-1/2}$ (up to some logarithmic factor), provided the Fourier transform of the regression function has a finite first moment, which basically requires that the function becomes smoother with increasing dimension d of X. McCaffrey and Gallant (1994) showed a rate of $n^{-\frac{2p}{2p+d+5}+\varepsilon}$ in case of a (p, C)-smooth regression function, but their study was restricted to the use of a certain cosine squasher as activation function.

In deep learning neural networks with several hidden layers are used to define classes of functions. Here a feedforward neural network with $L \in \mathbb{N}$ hidden layers and $k_s \in \mathbb{N}$ neurons in the layers $s \in \{1, \dots, L\}$ is recursively defined by

$$f_{\mathbf{w}}(x) = \sum_{j \in \{1, \dots, k_L\}} w_{1,j}^{(L)} \cdot f_j^{(L)}(x), \tag{5}$$

where

$$f_i^{(s)}(x) = \sigma \left(\sum_{j \in \{1, \dots, k_{s-1}\}} w_{i,j}^{(s-1)} \cdot f_j^{(s-1)}(x) + w_{i,0}^{(s-1)} \right) \quad \text{for } s \in \{2, \dots, L\} \text{ and } i > 0 \tag{6}$$

and

$$f_i^{(1)}(x) = \sigma \left(\sum_{j \in \{1, \dots, d\}} w_{i,j}^{(0)} \cdot x^{(j)} + w_{i,0}^{(0)} \right) \quad \text{for } i > 0. \tag{7}$$

The rate of convergence of least squares estimates based on multilayer neural networks has been analyzed in Kohler and Krzyżak (2017), Imaizumi and Fukamizu (2018), Bauer and Kohler (2019), Suzuki and Nitanda (2019), Schmidt-Hieber (2020) and Kohler and Langer (2021). One of the main results achieved in this context shows that neural networks can achieve some kind of dimension reduction under rather general assumptions. The most general form goes back to Schmidt-Hieber (2020) and can be formalized as follows:

Definition 2 *Let $d \in \mathbb{N}$ and $m : \mathbb{R}^d \to \mathbb{R}$ and let $\mathcal{P}$ be a subset of $(0, \infty) \times \mathbb{N}$.*
a) *We say that m satisfies a hierarchical composition model of level 0 with order and smoothness constraint $\mathcal{P}$, if there exists a $K \in \{1, \dots, d\}$ such that*

$$m(\mathbf{x}) = x^{(K)} \quad \textit{for all } \mathbf{x} = (x^{(1)}, \dots, x^{(d)})^\top \in \mathbb{R}^d.$$

b) *We say that m satisfies a hierarchical composition model of level $l+1$ with order and smoothness constraint $\mathcal{P}$, if there exist $(p,K) \in \mathcal{P}$, $C > 0$, $g : \mathbb{R}^K \to \mathbb{R}$ and $f_1, \dots, f_K : \mathbb{R}^d \to \mathbb{R}$, such that g is (p,C)–smooth, $f_1, \dots, f_K$ satisfy a hierarchical composition model of level l with order and smoothness constraint $\mathcal{P}$ and*

$$m(\mathbf{x}) = g(f_1(\mathbf{x}), \dots, f_K(\mathbf{x})) \quad \textit{for all } \mathbf{x} \in \mathbb{R}^d.$$

Schmid-Hieber (2020) showed that suitable least squares neural network regression estimates achieve (up to some logarithmic factor) a rate of convergence of order

$$\max_{(p,K)\in\mathcal{P}} n^{-\frac{2p}{2p+K}}$$

in case that the regression function satisfies a hierarchical composition model of some finite level with order and smoothness constraint $\mathcal{P}$. Since this rate of convergence does not depend on the dimension d of X, this results shows that least squares neural network regression estimates are able to circumvent the curse of dimensionality in case that the regression function satisfies a hierarchical composition model.

1.5 Learning of neural network estimates

In applications the least squares estimates of the previous subsection cannot be used, since it is not clear how one can minimize the empirical L_2 risk, which is a nonlinear and non-convex function of the weights. Instead, one uses gradient descent applied to a randomly chosen starting vector of weights to minimize it approximately. Typically, here the estimates are over-parameterized, i.e., they use much more weights than there are data points, so in principle it is possible to choose the weights such that the data points are interpolated (at least, if the x values are all distinct).

In practice it has been observed, that this procedure leads to estimates which predict well on new independent test data. There have been various attemps to explain this using some models for deep learning. E.g., Choromanska et al. (2015) used random matrix theory to show that in this model of deep learning the so–called landscape hypothesis is true, which states that the loss surface contains many deep local minima. Other popular models for deep learning include the neural tangent kernel setting proposed by Jacot, Gabriel and Hongler (2020) or the meanfield approach (cf., e.g., Mei, Montanari, and Nguyen (2018)). The problem with studying deep neural networks in equivalent models is that it is unclear how close the behaviour of the deep networks in the proposed equivalent model is to the behaviour of the deep networks in the applications, because they are based on some approximation of the application using e.g. some asymptotic expansions.

There exits also various articles which study over-parametrized deep neural network estimates directly in a standard regression setting. Kohler and Krzyżak (2022) showed that these estimates can achieve a nearly optimal rate of convergence in case that the regression function is (p,C)–smooth with $p = 1/2$. Furthermore it was shown there that these estimates can be modified such that they achieve a dimension reduction in an interaction model. These results require that a penalized empirical L_2 risk is minimized by gradient descent. That such results also hold without using any regularization by a penalty term was shown in Drews and Kohler (2023). Again, the estimates achieve a nearly optimal rate of convergence only in case of a (p, C)–smooth regression function with $p = 1/2$.

1.6 Main results

In this article we extend the results from Kohler and Krzyżak (2022) and Drews and Kohler (2023) to the case of a general $p \in [1/2, \infty)$. To do this, we study over-parametrized deep neural network regression estimates with logistic activation function, where the values of all the weights are learned by gradient descent, in a standard regression model. We consider a (p, C)–smooth regression function, and we choose as topology of the neural network a linear combination of many parallel computed deep neural networks of fixed depth and width. We show that for a suitable initialization of the weights, a suitably chosen stepsize of the gradient descent, and a suitably chosen number of gradient descent steps the expected L_2 error of our estimates converges to zero with rate

$$n^{-\frac{2p}{2p+d}+\epsilon},$$

where $\epsilon > 0$ can be chosen as an arbitrary small constant.

In order to prove this result we show three crucial auxiliary results: Firstly, we show that during gradient descent our estimates stay in a function space which has a finite complexity (measured by its supremum norm covering number). We achieve this by showing that the weights remain bounded and consequently the derivatives of the estimate stay bounded, which enables us to bound the covering number using metric entropy bounds. Secondly, we derive new approximation results for neural networks with bounded weights, where the bounds fit the upper bounds on the covering number derived by using the metric entropy bounds. And thirdly, we show that the gradient descent is linked to a gradient descent applied to a linear Taylor polynomial of our network, and therefore can be analyzed by techniques develloped for the analysis of gradient descent for smooth convex functions.

In our theory over-parametrized deep neural networks do not overfit because the weights remain bounded during training and consequently the networks stay in a function space of bounded complexity. Furthermore, the gradient descent can find neural networks which approximate the unknown regression function well since the over-parametrized structure and the initialization of our network is such that with high probability there is a network with good approximation properties close to our initial network.

1.7 Discussion of related results

Motivated by the huge success of deep learning in applications, there have been already quite a few results derived concerning the theoretical analysis of these methods. E.g., there exist many results in approximation theory for deep neural networks, see, e.g., Yarotsky (2018), Yarotsky and Zhevnerchute (2019), Lu et al. (2020), Langer (2021) and the literature cited therein. These results show that smooth functions can be approximated well by deep neural networks and analyze what kind of topology and how many nonzero weights are necessary to approximate a smooth function up to some given error. In applications, the functions which one wants to approximate has to be estimated from observed data, which usually contains some random error. It has been also already analyzed how well a network learned from such noisy data generalizes on new independent test data. This has been done within the framework of the classical VC theory (using e.g. the result of Bartlett et al. (2019) to bound the VC dimension of classes of neural networks) or in case of over-parametrized deep neural networks (where the number of free parameters adjusted to the observed data set is much larger than the sample size) by using bounds on the Rademacher complexity (cf., e.g., Liang, Rakhlin and Sridharan (2015), Golowich, Rakhlin and Shamir (2019), Lin and Zhang (2019), Wang and Ma (2022) and the literature cited therein). By combining these kind of results it was possible to analyze the error of least squares regression estimates. Here it was shown in a series of papers (cf., e.g., Kohler and Krzyżak (2017), Bauer and Kohler (2019), Schmidt-Hieber (2020) and Kohler and Langer (2021)) that least squares regression estimates based on deep networks can achieve a dimension reduction in case that the function to be estimates satisfies a hierarchical composition model, i.e., in case that it is a composition of smooth functions which do either depend only on a few components or are rather smooth. This is due to the network structure of deep networks, which implies that the composition of networks is itself a deep network. Consequently, any approximation result of some kind of functions by deep networks can be extended to an approximation result of a composition of such function by a deep network representing a composition of the approximating networks. And hereby the number of weights and the depth of the network, which determine the VC dimension and hence the complexity of the network in case that it is not over-parametrized (cf., Bartlett et al. (2019)), changes not much. So such a network has the approximation properties and the complexity of a network for low dimensional predictors and hence can achieve a dimension reduction.

There also exist quite a few results on the optimization of deep neural networks. E.g., Zou et al. (2018), Du et al. (2019), Allen-Zhu, Li and Song (2019) and Kawaguchi and Huang (2019) analyzed the application of gradient descent to over-parameterized deep neural networks. It was shown in these papers that this leads to neural networks which (globally) minimize the empirical risk considered. Unfortunately, as was shown in Kohler and Krzyżak (2021), the corresponding estimates do not behave well on new independent data.

As pointed out by Kutyniok (2020), it is essential for a theoretical analysis of deep learning estimates to study simultaneously the approximation error, the generalization error and the optimization error, and none of the results mentioned above controlls all these three aspects together.

There exists various approaches where these three things are studied simultaneously in some equivalent models of deep learning. The most prominent approach here is the neural tangent kernel setting proposed by Jacot, Gabriel and Hongler (2020). Here instead of a neural network estimate a kernel estimate is studied and its error is used to bound the error of the neural network estimate. For further results in this context see Hanin and

Nica (2019) and the literature cited therein. As was pointed out in Nitanda and Suzuki (2021) in most studies in the neural tangent kernel setting the equivalence to deep neural networks holds only pointwise and not for the global L_2 error, hence from these result it is not clear how the L_2 error of the deep neural network estimate behaves. Nitanda and Suzuki (2021) were able to analyze the global error of an over-parametrized shallow neural network learned by gradient descent based on this approach. However, due to the use of the neural tangent kernel, also the smoothness assumption of the function to be estimated has to be defined with the aid of a norm involving the kernel, which does not lead to classical smoothness conditions, which makes it hard to understand the meaning of the results. Furthermore, their result did not specify how many neurons the shallow neural network must have, it was only shown that the results hold if this number of neurons is sufficiently large, and it is not clear whether it must grow, e.g., exponentially in the sample size or not. Another approach where the estimate is studied in some asymptotically equivalent model is the mean field approach, cf., Mei, Montanari, and Nguyen (2018), Chizat and Bach (2018) or Nguyen and Pham (2020).

The theory presented in this article is an extension of the theory develloped in Braun et al. (2023), Drews and Kohler (2023, 2024) and Kohler and Krzyżak (2022, 2023). The basic idea there is that for smooth activation functions the inner weights do not change much during learning if the stepsizes are sufficiently small and it was shown that at the same time the outer weights will be chosen suitably by gradient descent. In this article we extend this theory by showing that in our special topology gradient descent is also able to learn the inner weights locally, and by deriving a new approximation result for the approximation of (p, C)–smooth functions by deep neural network with bounded weights. In fact, it is the new approximation results which is essential to extend the previous results from (p, C)–smooth regression functions with $p = 1/2$ to the case of (p, C)–smooth regression function with general $p \geq 1/2$, and it can be shown that the rate of convergence of this article can also be achieved if only the weights of the output layer are changed during gradient descent and all other weights keep their initial values (cf., Remark 1). This approach is related to the so–called random feature networks, where the inner weights are not learned at all and gradient descent is applied only to the weights in the output level, cf., e.g., Huang, Chen and Siew (2006) and Rahimi and Recht (2008a, 2008b, 2009).

1.8 Notation

The sets of natural numbers, real numbers and nonnegative real numbers are denoted by $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}_+$, respectively. For $z \in \mathbb{R}$, we denote the smallest integer greater than or equal to z by $\lceil z \rceil$. The Euclidean norm of $x \in \mathbb{R}^d$ is denoted by $\|x\|$. For a closed and convex set $A \subseteq \mathbb{R}^d$ we denote by $Proj_A x$ that element $Proj_A x \in A$ with

$$\|x - Proj_A x\| = \min_{z \in A} \|x - z\|.$$

For $f : \mathbb{R}^d \to \mathbb{R}$

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

is its supremum norm, and we set

$$\|f\|_{\infty,A} = \sup_{x \in A} |f(x)|$$

for $A \subseteq \mathbb{R}^d$.

A finite collection $f_1, \ldots, f_N : \mathbb{R}^d \to \mathbb{R}$ is called an L_p ε–covering of $\mathcal{F}$ on x_1^n if for all $f \in \mathcal{F}$

$$\min_{1 \le j \le N} \left(\frac{1}{n} \sum_{k=1}^{n} |f(x_k) - f_j(x_k)|^p \right)^{1/p} \le \varepsilon$$

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

For $z \in \mathbb{R}$ and $\beta > 0$ we define $T_\beta z = \max\{-\beta, \min\{\beta, z\}\}$. If $f : \mathbb{R}^d \to \mathbb{R}$ is a function then we set $(T_\beta f)(x) = T_\beta\,(f(x))$.

1.9 Outline

The main result is formulated in Section 2 and proven in Section 3.

2 Estimation of a (p, C)–smooth regression function

Throughout the paper we let $\sigma(x) = 1/(1 + e^{-x})$ be the logistic squasher. We define the topology of our neural networks as follows: We let $K_n, L, r \in \mathbb{N}$ be parameters of our estimate and using these parameters we set

$$f_{\mathbf{w}}(x) = \sum_{j=1}^{K_n} w_{1,1,j}^{(L)} \cdot f_{j,1}^{(L)}(x) \tag{8}$$

for some $w_{1,1,1}^{(L)}, \ldots, w_{1,1,K_n}^{(L)} \in \mathbb{R}$, where $f_{j,1}^{(L)} = f_{\mathbf{w},j,1}^{(L)}$ are recursively defined by

$$f_{k,i}^{(l)}(x) = f_{\mathbf{w},k,i}^{(l)}(x) = \sigma\left(\sum_{j=1}^{r} w_{k,i,j}^{(l-1)} \cdot f_{k,j}^{(l-1)}(x) + w_{k,i,0}^{(l-1)} \right) \tag{9}$$

for some $w_{k,i,0}^{(l-1)}, \ldots, w_{k,i,r}^{(l-1)} \in \mathbb{R}$ $(l = 2, \ldots, L)$ and

$$f_{k,i}^{(1)}(x) = f_{\mathbf{w},k,i}^{(1)}(x) = \sigma\left(\sum_{j=1}^{d} w_{k,i,j}^{(0)} \cdot x^{(j)} + w_{k,i,0}^{(0)} \right) \tag{10}$$

for some $w_{k,i,0}^{(0)}, \ldots, w_{k,i,d}^{(0)} \in \mathbb{R}$.

This means that we consider neural networks which consist of K_n fully connected neural networks of depth L and width r computed in parallel and compute a linear combination of the outputs of these K_n neural networks. The weights in the k-th such network are denoted by $(w_{k,i,j}^{(l)})_{i,j,l}$, where $w_{k,i,j}^{(l)}$ is the weight between neuron j in layer l and neuron i in layer $l+1$.

We initialize the weights $\mathbf{w}^{(0)} = ((\mathbf{w}^{(0)})_{k,i,j}^{(l))})_{k,i,j,l}$ as follows: We set

$$(\mathbf{w}^{(0)})_{1,1,k}^{(L)} = 0 \quad (k = 1, \dots, K_n), \tag{11}$$

we choose $(\mathbf{w}^{(0)})_{k,i,j}^{(l)}$ uniformly distributed on $[-c_1, c_1]$ if $l \in \{1, \dots, L-1\}$, and we choose $(\mathbf{w}^{(0)})_{k,i,j}^{(0)}$ uniformly distributed on $[-c_2 \cdot (\log n) \cdot n^{\tau}, c_2 \cdot (\log n) \cdot n^{\tau}]$, where $c_1, c_2, \tau > 0$ are parameters of the estimate. Here the random values are defined such that all components of $\mathbf{w}^{(0)}$ are independent.

After initialization of the weights we perform $t_n \in \mathbb{N}$ gradient descent steps each with a step size $\lambda_n > 0$. Here we try to minimize the empirical L_2 risk

$$F_n(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} |Y_i - f_{\mathbf{w}}(X_i)|^2. \tag{12}$$

To do this we set

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t-1)}) \quad (t = 1, \dots, t_n). \tag{13}$$

Finally we define our estimate as a truncated version of the neural network with weight vector $\mathbf{w}^{(t_n)}$, i.e., we set

$$m_n(x) = T_{\beta_n}(f_{\mathbf{w}^{(t_n)}}(x)) \tag{14}$$

where $\beta_n = c_3 \cdot \log n$ and $T_\beta z = \max\{\min\{z, \beta\}, -\beta\}$ for $z \in \mathbb{R}$ and $\beta > 0$.

Our main result is the following bound on the expected L_2 error of this estimate.

Theorem 1 *Let $n \in \mathbb{N}$, let (X, Y), (X_1, Y_n), ..., (X_n, Y_n) be independent and identically distributed $\mathbb{R}^d \times \mathbb{R}$–valued random variables such that supp(X) is bounded and that*

$$\mathbf{E}\left\{e^{c_4 \cdot Y^2}\right\} < \infty \tag{15}$$

holds for some $c_4 > 0$. Let $p, C > 0$ where $p = q + \beta$ for some $q \in \mathbb{N}_0$ and $\beta \in (0, 1]$ with $p \geq 1/2$, and assume that the regression function $m : \mathbb{R}^d \to \mathbb{R}$ is (p, C)–smooth.

Set $\beta_n = c_3 \cdot \log n$ for some $c_3 > 0$ which satisfies $c_3 \cdot c_4 \geq 2$. Let $K_n \in \mathbb{N}$ be such that for some $\kappa > 0$

$$\frac{K_n}{n^\kappa} \to 0 \quad (n \to \infty) \quad \textit{and} \quad \frac{K_n}{n^{4 \cdot r \cdot (r+1) \cdot (L-1) + r \cdot (4d+6) + 6}} \to \infty \quad (n \to \infty).$$

Set

$$L = \lceil \log_2(q + d) \rceil + 1, \quad r = 2 \cdot \lceil (2p + d)^2 \rceil, \quad \tau = \frac{1}{2p + d}, \quad \lambda_n = \frac{c_5}{n \cdot K_n^3}$$

and

$$t_n = \left\lceil c_6 \cdot \frac{K_n^3}{\beta_n} \right\rceil$$

for some $c_5, c_6 > 0$. Let $\sigma(x) = 1/(1+e^{-x})$ be the logistic squasher, let $c_1, c_2 > 0$ be sufficiently large, and define the estimate m_n as above.

Then we have for any $\epsilon > 0$:

$$\mathbf{E} \int |m_n(x) - m(x)|^2 \mathbf{P}_X(dx) \le c_7 \cdot n^{-\frac{2p}{2p+d}+\epsilon}.$$

Remark 1. By combining the approximation result derived in the proof of Theorem 1 with the proof strategy presented in Kohler and Krzyżak (2022) and Drews and Kohler (2023) it is possible to show that the rate of convergence in Theorem 1 also holds if the inner weights are not learned at all and gradient descent is applied only to the weights in the output level.

Remark 2. It should be easy to extend the above result to interaction models as in Kohler and Krzyżak (2022) and Drews and Kohler (2023), i.e., to modify the estimate in Theorem 1 such that it achieves the rate of convergence

$$n^{-\frac{2p}{2p+d^*}+\epsilon}$$

in case that the regression function is given by a (p, C)–smooth interaction model where each function in the sum depends on at most $d^* \in \{1, \dots, d\}$ of the d components of X.

Remark 3. It is an open problem whether the above result can be extended to the case of an hierarchical composition model.

3 Proof of Theorem 1

Before we present the proof of Theorem 1 we present in separate subsections the key auxiliary results needed in the proof concerning optimization, approximation and generalization.

3.1 Neural network optimization

Our first lemma is our main tool to analyze gradient descent. In it we relate the gradient descent of our deep neural network to the gradient descent of the linear Taylor polynomial of the deep network, and use methods for the analysis of gradient descent applied to smooth convex functions in order to analyze the latter.

Lemma 1 *Let $d, J_n \in \mathbb{N}$, and for $\mathbf{w} \in \mathbb{R}^{J_n}$ let $f_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ be a (deep) neural network with weight vector $\mathbf{w}$. Assume that for each $x \in \mathbb{R}^d$*

$$\mathbf{w} \mapsto f_{\mathbf{w}}(x)$$

is a continuously differentiable function on $\mathbb{R}^{J_n}$. Let

$$F_n(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} |Y_i - f_{\mathbf{w}}(X_i)|^2$$

be the empirical L_2 risk of $f_{\mathbf{w}}$, and use gradient descent in order to minimize $F_n(\mathbf{w})$. To do this, choose a starting weight vector $\mathbf{w}^{(0)} \in \mathbb{R}^{J_n}$, choose $\delta_n \geq 0$ and let

$$A \subset \left\{ \mathbf{w} \in \mathbb{R}^{J_n} \, : \, \|\mathbf{w} - \mathbf{w}^{(0)}\| \leq \delta_n \right\}$$

be a closed and convex set of weight vectors. Choose a stepsize $\lambda_n \geq 0$ and a number of gradient descent steps $t_n \in \mathbb{N}$ and compute

$$\mathbf{w}^{(t+1)} = Proj_A \left(\mathbf{w}^{(t)} - \lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)}) \right)$$

for $t = 0, \dots, t_n - 1$.

Let $C_n, D_n \geq 0$, $\beta_n \geq 1$ and assume

$$\sum_{j=1}^{J_n} \left| \frac{\partial}{\partial w^{(j)}} f_{\mathbf{w}_1}(x) - \frac{\partial}{\partial w^{(j)}} f_{\mathbf{w}_2}(x) \right|^2 \leq C_n^2 \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2 \tag{16}$$

for all $\mathbf{w}_1, \mathbf{w}_2 \in A$, $x \in \{X_1, \dots, X_n\}$,

$$\|\nabla_{\mathbf{w}} F_n(\mathbf{w})\| \leq D_n \quad \textit{for all } \mathbf{w} \in A, \tag{17}$$

$$|Y_i| \leq \beta_n \quad (i = 1, \dots, n) \tag{18}$$

and

$$C_n \cdot \delta_n^2 \leq 1. \tag{19}$$

Let $\mathbf{w}^* \in A$ and assume

$$|f_{\mathbf{w}^*}(x)| \leq \beta_n \quad (x \in \{X_1, \dots, X_n\}). \tag{20}$$

Then

$$\min_{t=0,\dots,t_n-1} F_n(\mathbf{w}^{(t)}) \leq F_n(\mathbf{w}^*) + \frac{\|\mathbf{w}^* - \mathbf{w}^{(0)}\|^2}{2 \cdot \lambda_n \cdot t_n} + 12 \cdot \beta_n \cdot C_n \cdot \delta_n^2 + \frac{1}{2} \cdot \lambda_n \cdot D_n^2.$$

Proof. The basic idea of the proof is to analyze the gradient descent by relating it to the gradient descent of the linear Taylor polynomial of $f_{\mathbf{w}}$. To do this, we define for $\mathbf{w}_0, \mathbf{w} \in \mathbb{R}^{J_n}$ the linear Taylor polynomial of $f_{\mathbf{w}}(x)$ around $\mathbf{w}_0$ by

$$f_{lin,\mathbf{w}_0,\mathbf{w}}(x) = f_{\mathbf{w}_0}(x) + \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)})$$

and introduce the empirical L_2 risk of this linear approximation of $f_{\mathbf{w}}$ by

$$F_{n,lin,\mathbf{w}_0}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} |Y_i - f_{lin,\mathbf{w}_0,\mathbf{w}}(X_i)|^2.$$

Let $\alpha \in [0,1]$ and $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^{J_n}$. Then

$$\begin{aligned}
& f_{lin,\mathbf{w}_0,\alpha\cdot\mathbf{w}_1+(1-\alpha)\cdot\mathbf{w}_2}(x) \\
&= f_{\mathbf{w}_0}(x) + \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\alpha \cdot \mathbf{w}_1^{(j)} + (1-\alpha)\cdot \mathbf{w}_2^{(j)} - \mathbf{w}_0^{(j)}) \\
&= \alpha \cdot f_{\mathbf{w}_0}(x) + (1-\alpha)\cdot f_{\mathbf{w}_0}(x) + \alpha \cdot \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}_1^{(j)} - \mathbf{w}_0^{(j)}) \\
&\quad +(1-\alpha)\cdot \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}_1^{(j)} - \mathbf{w}_0^{(j)}) \\
&= \alpha \cdot f_{lin,\mathbf{w}_0,\mathbf{w}_1}(x) + (1-\alpha)\cdot f_{lin,\mathbf{w}_0,\mathbf{w}_2}(x),
\end{aligned}$$

which implies

$$\begin{aligned}
& F_{n,lin,\mathbf{w}_0}(\alpha \cdot \mathbf{w}_1 + (1-\alpha)\cdot \mathbf{w}_2) \\
&= \frac{1}{n}\sum_{i=1}^{n} |\alpha \cdot (Y_i - f_{lin,\mathbf{w}_0,\mathbf{w}_1}(X_i)) + (1-\alpha)\cdot (Y_i - f_{lin,\mathbf{w}_0,\mathbf{w}_2}(X_i))|^2 \\
&\leq \alpha \cdot F_{n,lin,\mathbf{w}_0}(\mathbf{w}_1) + (1-\alpha)\cdot F_{n,lin,\mathbf{w}_0}(\mathbf{w}_2).
\end{aligned}$$

Hence $F_{n,lin,\mathbf{w}_0}(\mathbf{w})$ is as a function of $\mathbf{w}$ a convex function.

Because of $f_{lin,\mathbf{w}_0,\mathbf{w}_0}(x) = f_{\mathbf{w}_0}(x)$ and $\nabla_{\mathbf{w}} f_{lin,\mathbf{w}_0,\mathbf{w}_0}(x) = \nabla_w f_{\mathbf{w}_0}(x)$ we have

$$F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^{(t)}) = F_n(\mathbf{w}^{(t)}) \quad \text{and} \quad \nabla_w F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^{(t)}) = \nabla_w F_n(\mathbf{w}^{(t)}),$$

hence $\mathbf{w}^{(t+1)}$ is computed from $\mathbf{w}^{(t)}$ by one gradient descent step

$$\mathbf{w}^{(t+1)} = Proj_A \left(\mathbf{w}^{(t)} - \lambda_n \cdot \nabla_{\mathbf{w}} F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^{(t)}) \right)$$

applied to the convex function $F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w})$. This will enable us to use techniques for the analysis of the gradient descent for convex functions in order to analyze the gradient descent applied to the nonconvex function $F_n(\mathbf{w})$.

In order to do this we observe

$$\begin{aligned}
& \min_{t=0,\dots,t_n-1} F_n(\mathbf{w}^{(t)}) - F_n(\mathbf{w}^*) \\
&\leq \frac{1}{t_n}\sum_{t=0}^{t_n-1} (F_n(\mathbf{w}^{(t)}) - F_n(\mathbf{w}^*)) \\
&= \frac{1}{t_n}\sum_{t=0}^{t_n-1} (F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^{(t)}) - F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^*)) + \frac{1}{t_n}\sum_{t=0}^{t_n-1} (F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^*) - F_n(\mathbf{w}^*)) \\
&=: T_{1,n} + T_{2,n}.
\end{aligned}$$

Next we show that assumption (16) implies

$$|f_{\mathbf{w}}(x) - f_{lin,\mathbf{w}_0,\mathbf{w}}(x)| \leq \frac{1}{2} \cdot C_n \cdot \|\mathbf{w} - \mathbf{w}_0\|^2$$

for all $x \in \{X_1, \dots, X_n\}$ and all $\mathbf{w}_0, \mathbf{w} \in A$. To do this, set

$$H(s) = f_{\mathbf{w}_0 + s\cdot(\mathbf{w}-\mathbf{w}_0)}(x) \quad \text{for } s \in [0,1].$$

Let $\mathbf{w}, \mathbf{w}_0 \in A$. Then A convex implies

$$\mathbf{w}_0 + s \cdot (\mathbf{w} - \mathbf{w}_0) = (1-s) \cdot \mathbf{w}_0 + s \cdot \mathbf{w} \in A$$

for all $s \in [0,1]$, hence we can conclude from (16)

$$\begin{aligned}
&|f_{\mathbf{w}}(x) - f_{lin,\mathbf{w}_0,\mathbf{w}}(x)| \\
&= |f_{\mathbf{w}}(x) - f_{\mathbf{w}_0}(x) - \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)})| \\
&= |H(1) - H(0) - \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)})| \\
&= |\int_0^1 H'(s)\, ds - \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)})| \\
&= |\int_0^1 \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0+s\cdot(\mathbf{w}-\mathbf{w}_0)}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)})\, ds - \sum_{j=1}^{J_n} \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}} \cdot (\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)})| \\
&= |\int_0^1 \sum_{j=1}^{J_n} (\frac{\partial f_{\mathbf{w}_0+s\cdot(\mathbf{w}-\mathbf{w}_0)}(x)}{\partial \mathbf{w}^{(j)}} - \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}}) \cdot (\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)})\, ds| \\
&\leq \int_0^1 \sum_{j=1}^{J_n} |\frac{\partial f_{\mathbf{w}_0+s\cdot(\mathbf{w}-\mathbf{w}_0)}(x)}{\partial \mathbf{w}^{(j)}} - \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}}| \cdot |\mathbf{w}^{(j)} - \mathbf{w}_0^{(j)}|\, ds \\
&\leq \int_0^1 \sqrt{\sum_{j=1}^{J_n} |\frac{\partial f_{\mathbf{w}_0+s\cdot(\mathbf{w}-\mathbf{w}_0)}(x)}{\partial \mathbf{w}^{(j)}} - \frac{\partial f_{\mathbf{w}_0}(x)}{\partial \mathbf{w}^{(j)}}|^2} \cdot \|\mathbf{w} - \mathbf{w}_0\|\, ds \\
&\leq \int_0^1 \sqrt{C_n^2 \cdot \|\mathbf{w}_0 + s \cdot (\mathbf{w} - \mathbf{w}_0) - \mathbf{w}_0\|^2} \cdot \|\mathbf{w} - \mathbf{w}_0\|\, ds \\
&\leq C_n \cdot \|\mathbf{w} - \mathbf{w}_0\|^2 \cdot \int_0^1 s\, ds = \frac{1}{2} \cdot C_n \cdot \|\mathbf{w} - \mathbf{w}_0\|^2.
\end{aligned}$$

Using (18)–(20) we can conclude for all $\mathbf{w}_0 \in A$

$$|F_n(\mathbf{w}^*) - F_{n,lin,\mathbf{w}_0}(\mathbf{w}^*)|$$

$$
\begin{aligned}
&\leq \frac{1}{n}\sum_{i=1}^{n} |Y_i - f_{\mathbf{w}^*}(X_i) + Y_i - f_{lin,\mathbf{w}_0,\mathbf{w}^*}(X_i)| \cdot |f_{\mathbf{w}^*}(X_i) - f_{lin,\mathbf{w}_0,\mathbf{w}^*}(X_i)| \\
&\leq \frac{1}{n}\sum_{i=1}^{n}(4 \cdot \beta_n + \frac{1}{2} \cdot C_n \cdot \|\mathbf{w}^* - \mathbf{w}_0\|^2) \cdot \frac{1}{2} \cdot C_n \cdot \|\mathbf{w}^* - \mathbf{w}_0\|^2 \\
&\leq \frac{1}{n}\sum_{i=1}^{n}(4 \cdot \beta_n + \frac{1}{2} \cdot C_n \cdot 4\delta_n^2) \cdot \frac{1}{2} \cdot C_n \cdot 4\delta_n^2 \\
&\leq 12 \cdot \beta_n \cdot C_n \cdot \delta_n^2.
\end{aligned}
$$

This proves

$$T_{2,n} \leq 12 \cdot \beta_n \cdot C_n \cdot \delta_n^2,$$

and it suffices to show

$$T_{1,n} \leq \frac{\|\mathbf{w}^* - \mathbf{w}^{(0)}\|^2}{2 \cdot \lambda_n \cdot t_n} + \frac{1}{2} \cdot \lambda_n \cdot D_n^2. \tag{21}$$

The convexity of $F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w})$ together with $\mathbf{w}^* \in A$ implies

$$
\begin{aligned}
&F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^{(t)}) - F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^*) \\
&\leq < \nabla_{\mathbf{w}} F_{n,lin,\mathbf{w}^{(t)}}(\mathbf{w}^{(t)}), \mathbf{w}^{(t)} - \mathbf{w}^* > \\
&= < \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)}), \mathbf{w}^{(t)} - \mathbf{w}^* > \\
&= \frac{1}{2 \cdot \lambda_n} \cdot 2 \cdot < \lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)}), \mathbf{w}^{(t)} - \mathbf{w}^* > \\
&= \frac{1}{2 \cdot \lambda_n} \cdot \left(\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^* - \lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})\|^2 + \|\lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})\|^2 \right) \\
&= \frac{1}{2 \cdot \lambda_n} \cdot \left(\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2 - \|\mathbf{w}^{(t)} - \lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)}) - \mathbf{w}^*\|^2 \right) + \frac{1}{2} \cdot \lambda_n \cdot \|\nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})\|^2 \\
&\leq \frac{1}{2 \cdot \lambda_n} \cdot \left(\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2 - \|Proj_A\left(\mathbf{w}^{(t)} - \lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})\right) - \mathbf{w}^*\|^2 \right) \\
&\quad + \frac{1}{2} \cdot \lambda_n \cdot \|\nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})\|^2 \\
&= \frac{1}{2 \cdot \lambda_n} \cdot \left(\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2 - \|\mathbf{w}^{(t+1)} - \mathbf{w}^*\|^2 \right) + \frac{1}{2} \cdot \lambda_n \cdot \|\nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})\|^2.
\end{aligned}
$$

This together with (17) implies

$$
\begin{aligned}
T_{1,n} &\leq \frac{1}{t_n}\sum_{t=0}^{t_n-1}\left(\frac{1}{2 \cdot \lambda_n} \cdot \left(\|\mathbf{w}^{(t)} - \mathbf{w}^*\|^2 - \|\mathbf{w}^{(t+1)} - \mathbf{w}^*\|^2 \right) + \frac{1}{2} \cdot \lambda_n \cdot \|\nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})\|^2 \right) \\
&\leq \frac{\|\mathbf{w}^{(0)} - \mathbf{w}^*\|^2}{2 \cdot \lambda_n \cdot t_n} + \frac{1}{2} \cdot \lambda_n \cdot D_n^2,
\end{aligned}
$$

which proves (21). □

Next we consider the topology of the deep neural network introduced in Section 2 (cf., (8)-(10)) and investigate when the assumptions of Lemma 1 are satisfied.

Our next lemma considers inequality (16) in this case.

Lemma 2 *Let σ be the logistic squasher. Let $a, B_n, \gamma_n^* \geq 1$, $L, r \in \mathbb{N}$ and define the deep neural network $f_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ with weight vector $\mathbf{w}$ by (8)–(10). Assume that the weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ satisfy*

$$|w_{1,1,k}^{(L)}| \leq \gamma_n^* \quad \textit{and} \quad |w_{k,i,j}^{(l)}| \leq B_n$$

for all $l \in \{1, \dots, L-1\}$. Then we have for any $x \in [-a,a]^d$

$$\sum_{k,i,j,l} \left| \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}_1}(x) - \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}_2}(x) \right|^2 \leq c_8 \cdot B_n^{4L} \cdot (\gamma_n^*)^2 \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2$$

for some $c_8 = c_8(d, L, r, a) > 0$.

Proof. We have

$$\begin{aligned}
&\sum_{k,i,j,l} \left| \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}_1}(x) - \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}_2}(x) \right|^2 \\
&= \sum_{k=1}^{K_n} |f_{\mathbf{w}_1,k,1}^{(L)}(x) - f_{\mathbf{w}_2,k,1}^{(L)}(x)|^2 \\
&\quad + \sum_{k=1}^{K_n} \sum_{i,j,l:l<L} \left| (\mathbf{w}_1)_{1,1,k}^{(L)} \cdot \frac{\partial}{\partial \mathbf{w}_{k,i,j}^{(l)}} f_{\mathbf{w}_1,k,1}^{(L)}(x) - (\mathbf{w}_2)_{1,1,k}^{(L)} \cdot \frac{\partial}{\partial \mathbf{w}_{k,i,j}^{(l)}} f_{\mathbf{w}_2,k,1}^{(L)}(x) \right|^2 \\
&\leq \sum_{k=1}^{K_n} |f_{\mathbf{w}_1,k,1}^{(L)}(x) - f_{\mathbf{w}_2,k,1}^{(L)}(x)|^2 \\
&\quad + 2 \cdot \sum_{k=1}^{K_n} \sum_{i,j,l:l<L} \left| (\mathbf{w}_1)_{1,1,k}^{(L)} - (\mathbf{w}_2)_{1,1,k}^{(L)} \right|^2 \cdot |\frac{\partial}{\partial \mathbf{w}_{k,i,j}^{(l)}} f_{\mathbf{w}_1,k,1}^{(L)}(x)|^2 \\
&\quad + 2 \cdot \sum_{k=1}^{K_n} \sum_{i,j,l:l<L} \left| (\mathbf{w}_2)_{1,1,k}^{(L)} \right|^2 \cdot \left| \frac{\partial}{\partial \mathbf{w}_{k,i,j}^{(l)}} f_{\mathbf{w}_1,k,1}^{(L)}(x) - \frac{\partial}{\partial \mathbf{w}_{k,i,j}^{(l)}} f_{\mathbf{w}_2,k,1}^{(L)}(x) \right|^2 .
\end{aligned}$$

The chain rule implies

$$\begin{aligned}
\frac{\partial f_{\mathbf{w},k,1}^{(L)}}{\partial w_{k,i,j}^{(l)}}(x) &= \sum_{s_{l+2}=1}^{r} \cdots \sum_{s_{L-1}=1}^{r} f_{k,j}^{(l)}(x) \cdot \sigma' \left(\sum_{t=1}^{r} w_{k,i,t}^{(l)} \cdot f_{k,t}^{(l)}(x) + w_{k,i,0}^{(l)} \right) \\
&\cdot w_{k,s_{l+2},i}^{(l+1)} \cdot \sigma' \left(\sum_{t=1}^{r} w_{k,s_{l+2},t}^{(l+1)} \cdot f_{k,t}^{(l+1)}(x) + w_{k,s_{l+2},0}^{(l+1)} \right) \cdot w_{k,s_{l+3},s_{l+2}}^{(l+2)} \\
&\cdot \sigma' \left(\sum_{t=1}^{r} w_{k,s_{l+3},t}^{(l+2)} \cdot f_{k,t}^{(l+2)}(x) + w_{k,s_{l+3},0}^{(l+2)} \right) \cdots w_{k,s_{L-1},s_{L-2}}^{(L-2)}
\end{aligned}$$

$$
\begin{aligned}
&\cdot \sigma' \left(\sum_{t=1}^{r} w_{k,s_{L-1},t}^{(L-2)} \cdot f_{k,t}^{(L-2)}(x) + w_{k,s_{L-1},0}^{(L-2)} \right) \cdot w_{k,1,s_{L-1}}^{(L-1)} \\
&\cdot \sigma' \left(\sum_{t=1}^{r} w_{k,1,t}^{(L-1)} \cdot f_{k,t}^{(L-1)}(x) + w_{k,1,0}^{(L-1)} \right), \qquad (22)
\end{aligned}
$$

where we have used the abbreviations

$$
f_{k,j}^{(0)}(x) = \begin{cases} x^{(j)} & \text{if } j \in \{1, \dots, d\} \\ 1 & \text{if } j = 0 \end{cases}
$$

and

$$
f_{k,0}^{(l)}(x) = 1 \quad (l = 1, \dots, L-1).
$$

If $f_{i,1}$, ..., $f_{i,L}$ are real–valued functions defined on $\mathbb{R}^{J_n}$ where $f_{i,l}$ is bounded in absolute value by $B_{i,l} \geq 1$ and Lipschitz continuous (w.r.t. $\| \cdot \|_\infty$) with Lipschitz constant $C_{i,l} \geq 1$ $(i = 1, \dots, r)$ then

$$
\begin{aligned}
&|\sum_{i=1}^{r} \prod_{l=1}^{L} f_{i,l}(\mathbf{w}_1) - \sum_{i=1}^{r} \prod_{l=1}^{L} f_{i,l}(\mathbf{w}_2)| \\
&\leq \sum_{i=1}^{r} \sum_{j=1}^{L} \prod_{l=1}^{j-1} |f_{i,l}(\mathbf{w}_1)| \cdot |f_{j,l}(\mathbf{w}_1) - f_{j,l}(\mathbf{w}_2)| \prod_{l=j+1}^{L} |f_{i,l}(\mathbf{w}_2)| \\
&\leq r \cdot L \cdot \max_{i=1,\dots,r} \prod_{l=1}^{L} B_{i,l} \cdot \max_{i=1,\dots,r} \max_{l=1,\dots,L} C_{i,l} \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|_\infty.
\end{aligned}
$$

Using this,

$$
0 \leq \sigma(x) \leq 1 \quad \text{and} \quad |\sigma'(x)| = |\sigma(x) \cdot (1 - \sigma(x))| \leq 1
$$

and

$$
\begin{aligned}
&|f_{\mathbf{w}_1,k,j}^{(l)}(x) - f_{\mathbf{w}_2,k,j}^{(l)}(x)| \\
&\leq c_9 \cdot a \cdot (\max\{2r, d\} + 1)^l \cdot B_n^{l-1} \cdot \|((\mathbf{w}_1)_{k,\bar{i},\bar{j}}^{(\bar{l})})_{\bar{i},\bar{j},\bar{l}} - ((\mathbf{w}_2)_{k,\bar{i},\bar{j}}^{(\bar{l})})_{\bar{i},\bar{j},\bar{l}}\|_\infty \\
&\leq c_9 \cdot a \cdot (\max\{2r, d\} + 1)^l \cdot B_n^{l-1} \cdot \|((\mathbf{w}_1)_{k,\bar{i},\bar{j}}^{(\bar{l})})_{\bar{i},\bar{j},\bar{l}} - ((\mathbf{w}_2)_{k,\bar{i},\bar{j}}^{(\bar{l})})_{\bar{i},\bar{j},\bar{l}}\|
\end{aligned}
$$

(which can be easily shown by induction on l) we get

$$
\begin{aligned}
&\sum_{k,i,j,l} \left| \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}_1}(x) - \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}_2}(x) \right|^2 \\
&\leq c_{10} \cdot a^2 \cdot (2r + d)^{2L} \cdot B_n^{2L} \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2 \\
&\quad + c_{11} \cdot L \cdot (r \cdot (r + d)) \cdot r^{2L} \cdot a^2 \cdot B_n^{2L} \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2 \\
&\quad + c_{12} \cdot (\gamma_n^*)^2 \cdot L \cdot (r \cdot (r + d)) \cdot r^{2L} \cdot (3L)^2 \cdot a^4 \cdot B_n^{4L} \cdot (2r + d)^{2L} \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2
\end{aligned}
$$

$$\leq c_{13} \cdot (\gamma_n^*)^2 \cdot L^3 \cdot (2r+d)^{4L+2} \cdot B_n^{4L} \cdot a^4 \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2.$$

□

Next we consider inequality (17) in case of the special topology of our networks.

Lemma 3 *Let σ be the logistic squasher. Let $a, \beta_n, B_n, \gamma_n^* \geq 1$, $K_n, L, r \in \mathbb{N}$ and define the deep neural network $f_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ with weight vector $\mathbf{w}$ by (8)–(10), and assume*

$$X_i \in [-a,a]^d \quad \textit{and} \quad |Y_i| \leq \beta_n \quad (i=1,\dots,n)$$

and

$$|w_{1,1,k}^{(L)}| \leq \gamma_n^* \quad \textit{and} \quad |w_{k,i,j}^{(l)}| \leq B_n$$

for all $l \in \{1,\dots,L-1\}$. Assume

$$K_n \cdot \gamma_n^* \geq \beta_n.$$

Then

$$\|\nabla_{\mathbf{w}} F_n(\mathbf{w})\| \leq c_{14} \cdot K_n^{3/2} \cdot (\gamma_n^*)^2 \cdot B_n^L$$

for some $c_{14} = c_{14}(d,L,r,a) > 0$.

Proof. We have

$$|f_{\mathbf{w}}(x)| \leq K_n \cdot \gamma_n^*,$$

which implies

$$\begin{aligned}
&\|\nabla_{\mathbf{w}} F_n(\mathbf{w})\|^2 \\
&= \sum_{k,i,j,l} \left| \frac{1}{n} \sum_{s=1}^n 2 \cdot (Y_s - f_{\mathbf{w}}(X_s)) \cdot \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}}(X_s) \cdot (-1) \right|^2 \\
&\leq 8 \cdot (K_n \cdot \gamma_n^*)^2 \cdot \sum_{k,i,j,l} \max_{s=1,\dots,n} \left| \frac{\partial}{\partial w_{k,i,j}^{(l)}} f_{\mathbf{w}}(X_s) \right|^2 \\
&= 8 \cdot (K_n \cdot \gamma_n^*)^2 \cdot \left(\sum_{k=1}^{K_n} \max_{s=1,\dots,n} |f_{\mathbf{w},k,1}^{(L)}(X_s)|^2 \right. \\
&\qquad \left. + \sum_{k=1}^{K_n} \sum_{i,j,l:l<L} \max_{s=1,\dots,n} \left| \mathbf{w}_{1,1,k}^{(L)} \cdot \frac{\partial}{\partial \mathbf{w}_{k,i,j}^{(l)}} f_{\mathbf{w},k,1}^{(L)}(X_s) \right|^2 \right) \\
&\leq 8 \cdot (K_n \cdot \gamma_n^*) \cdot \Bigg(K_n \cdot 1 \\
&\qquad + K_n \cdot L \cdot (r \cdot (r+d))^L \cdot (\gamma_n^*)^2 \cdot \max_{s=1,\dots,n} \max_{k,i,j,l:l<L} \left| \frac{\partial}{\partial \mathbf{w}_{k,i,j}^{(l)}} f_{\mathbf{w},k,1}^{(L)}(X_s) \right|^2 \Bigg)
\end{aligned}$$

$$\leq 8 \cdot (K_n \cdot \gamma_n^*)^2 \cdot \left(K_n \cdot 1 + K_n \cdot L \cdot (r \cdot (r+d)) \cdot (\gamma_n^*)^2 \cdot r^{2L} \cdot a^2 \cdot B_n^{2L} \right),$$

where the last inequality follows from (22), the assumptions on the weights and the bounds on the logistic squasher mentioned in the proof of Lemma 2. □

In order to avoid the projection step in Lemma 1, we use the following localization lemma for gradient descent proven in Braun et al. (2023).

Lemma 4 *Let $F : \mathbb{R}^K \to \mathbb{R}_+$ be a nonnegative differentiable function. Let $t \in \mathbb{N}$, $L > 0$, $\mathbf{a}_0 \in \mathbb{R}^K$ and set*

$$\lambda = \frac{1}{L}$$

and

$$\mathbf{a}_{k+1} = \mathbf{a}_k - \lambda \cdot (\nabla_{\mathbf{a}} F)(\mathbf{a}_k) \quad (k \in \{0, 1, \dots, t-1\}).$$

Assume

$$\|(\nabla_{\mathbf{a}} F)(\mathbf{a})\| \leq \sqrt{2 \cdot t \cdot L \cdot \max\{F(\mathbf{a}_0), 1\}} \tag{23}$$

for all $\mathbf{a} \in \mathbb{R}^K$ with $\|\mathbf{a} - \mathbf{a}_0\| \leq \sqrt{2 \cdot t \cdot \max\{F(\mathbf{a}_0), 1\}/L}$, and

$$\|(\nabla_{\mathbf{a}} F)(\mathbf{a}) - (\nabla_{\mathbf{a}} F)(\mathbf{b})\| \leq L \cdot \|\mathbf{a} - \mathbf{b}\| \tag{24}$$

for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^K$ satisfying

$$\|\mathbf{a} - \mathbf{a}_0\| \leq \sqrt{8 \cdot \frac{t}{L} \cdot \max\{F(\mathbf{a}_0), 1\}} \quad \textit{and} \quad \|\mathbf{b} - \mathbf{a}_0\| \leq \sqrt{8 \cdot \frac{t}{L} \cdot \max\{F(\mathbf{a}_0), 1\}}. \tag{25}$$

Then we have

$$\|\mathbf{a}_k - \mathbf{a}_0\| \leq \sqrt{2 \cdot \frac{k}{L} \cdot (F(\mathbf{a}_0) - F(\mathbf{a}_k))} \quad \textit{for all } k \in \{1, \dots, t\}$$

and

$$F(\mathbf{a}_k) \leq F(\mathbf{a}_{k-1}) \quad \textit{for all } k \in \{1, \dots, t\}.$$

Proof. See Lemma A.1 in Braun et al. (2023) □

Our next lemma helps us to verify the assumption (24) of Lemma 4.

Lemma 5 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the logistic squasher, let $f_{\mathbf{w}}$ be defined by (8)–(10), and let F_n be defined by (12). Let $a \geq 1$, $\gamma_n^* \geq 1$, $B_n \geq 1$, and assume $X_i \in [-a, a]^d$ $(i = 1, \dots, n)$,*

$$\max\{|(\mathbf{w}_1)_{1,1,k}^{(L)}|, |(\mathbf{w}_2)_{1,1,k}^{(L)}|\} \leq \gamma_n^* \quad (k = 1, \dots, K_n), \tag{26}$$

$$\max\{|(\mathbf{w}_1)_{k,i,j}^{(l)}|, |(\mathbf{w}_2)_{k,i,j}^{(l)}|\} \leq B_n \quad \textit{for } l = 1, \dots, L-1 \tag{27}$$

and

$$K_n \cdot \gamma_n^* \geq \beta_n.$$

Then we have

$$\|(\nabla_{\mathbf{w}} F_n)(\mathbf{w}_1) - (\nabla_{\mathbf{w}} F_n)(\mathbf{w}_2)\| \leq c_{15} \cdot K_n^{3/2} \cdot B_n^{2L} \cdot (\gamma_n^*)^2 \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|.$$

Proof. We have

$$
\begin{aligned}
&\|\nabla_{\mathbf{w}} F_n(\mathbf{w}_1) - \nabla_{\mathbf{w}} F_n(\mathbf{w}_2)\|^2 \\
&= \sum_{k,i,j,l} \left(\frac{2}{n} \sum_{s=1}^{n} (f_{\mathbf{w}_1}(X_s) - Y_s) \cdot \frac{\partial f_{\mathbf{w}_1}}{\partial w_{k,i,j}^{(l)}}(X_s) - \sum_{k,i,j,l} \left(\frac{2}{n} \sum_{s=1}^{n} (f_{\mathbf{w}_2}(X_s) - Y_s) \cdot \frac{\partial f_{\mathbf{w}_2}}{\partial w_{k,i,j}^{(l)}}(X_s) \right)^2 \\
&\leq 8 \cdot \sum_{k,i,j,l} \max_{s=1,\dots,n} \left(\frac{\partial f_{\mathbf{w}_1}}{\partial w_{k,i,j}^{(l)}}(X_s) \right)^2 \cdot \frac{1}{n} \sum_{s=1}^{n} (f_{\mathbf{w}_2}(X_s) - f_{\mathbf{w}_1}(X_s))^2 \\
&\quad +8 \cdot \frac{1}{n} \sum_{s=1}^{n} (Y_s - f_{\mathbf{w}_2}(X_s))^2 \cdot \sum_{k,i,j,l} \left(\frac{\partial f_{\mathbf{w}_1}}{\partial w_{k,i,j}^{(l)}}(X_s) - \frac{\partial f_{\mathbf{w}_2}}{\partial w_{k,i,j}^{(l)}}(X_s) \right)^2 .
\end{aligned}
$$

From the proof of Lemma 2 we can conclude

$$
\sum_{k,i,j,l} \max_{s=1,\dots,n} \left(\frac{\partial f_{\mathbf{w}_1}}{\partial w_{k,i,j}^{(l)}}(X_s) \right)^2 \leq c_{16} \cdot K_n \cdot L \cdot (r \cdot (r+d)) \cdot r^{2L} \cdot (\gamma_n^*)^2 \cdot B_n^{2L} \cdot a^2,
$$

$$
\frac{1}{n} \sum_{s=1}^{n} (f_{\mathbf{w}_2}(X_s) - f_{\mathbf{w}_1}(X_s))^2 \leq c_{17} \cdot K_n^2 \cdot (\gamma_n^*)^2 \cdot (2r+d)^{2L} \cdot B_n^{2L} \cdot a^2 \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2
$$

and from the proof of Lemma 3 we know

$$
\frac{1}{n} \sum_{s=1}^{n} (Y_s - f_{\mathbf{w}_2}(X_s))^2 \leq 4 \cdot K_n^2 \cdot (\gamma_n^*)^2.
$$

And by Lemma 2 we can conclude for any $s \in \{1, \dots, n\}$

$$
\sum_{k,i,j,l} \left(\frac{\partial f_{\mathbf{w}_1}}{\partial w_{k,i,j}^{(l)}}(X_s) - \frac{\partial f_{\mathbf{w}_2}}{\partial w_{k,i,j}^{(l)}}(X_s) \right)^2 \leq c_{18} \cdot B_n^{4L} \cdot (\gamma_n^*)^2 \cdot \|\mathbf{w}_1 - \mathbf{w}_2\|^2.
$$

Summarizing the above results we get the assertion. □

By combining the above results we can show our main result concerning gradient descent.

Theorem 2 *Let $A \geq 1$, $L, r \in \mathbb{N}$ and $K_n \in \mathbb{N}$. Define the deep neural network $f_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ with weight vector $\mathbf{w}$ by (8)–(10). Let $\beta_n, A_n, B_n \geq 1$ with $B_n \leq K_n$, and assume $X_1, \dots, X_n \in [-A, A]^d$, $|Y_i| \leq \beta_n$ $(i = 1, \dots, n)$ and*

$$
c_{19} \cdot K_n \geq \beta_n \quad \textit{and} \quad \frac{\beta_n}{n^2} \leq c_{20}.
$$

Set

$$
F_n(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} |Y_i - f_{\mathbf{w}}(X_i)|^2.
$$

Choose some starting weight vector $\mathbf{w}^{(0)}$ *which satisfies*

$$(\mathbf{w}^{(0)})^{(L)}_{1,1,k} = 0, \quad |(\mathbf{w}^{(0)})^{(l)}_{k,i,j}| \leq B_n \quad \textit{and} \quad |(\mathbf{w}^{(0)})^{(0)}_{k,i,j}| \leq A_n$$

for all $l \in \{1, \dots, L-1\}$ *and all* i, j, k, *and set*

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \lambda_n \cdot \nabla_{\mathbf{w}} F_n(\mathbf{w}^{(t)})$$

for $t = 0, 1, \dots, t_n - 1$. *Set*

$$\lambda_n = \frac{c_{21}}{K_n^3 \cdot B_n^{2L} \cdot n} \quad \textit{and} \quad t_n = \left\lceil c_{22} \cdot B_n^L \frac{K_n^3}{\beta_n} \right\rceil.$$

Let $\mathbf{w}^* \in A$ *where*

$$A = \left\{ \mathbf{w} \quad : \quad \|\mathbf{w} - \mathbf{w}^{(0)}\| \leq \frac{c_{23}}{\sqrt{n} \cdot B_n^L} \right\}$$

and assume that (20) holds. Then

$$F_n(\mathbf{w}) \leq F_n(\mathbf{w}^*) + c_{24} \cdot \beta_n \cdot B_n^{2L} \cdot n \cdot \|\mathbf{w}^* - \mathbf{w}^{(0)}\|^2 + c_{25} \cdot \frac{\beta_n}{n}.$$

Proof. Set

$$L_n = c_{26} \cdot K_n^3 \cdot B_n^{2L} \cdot n \quad \text{and} \quad C_n = c_{27} \cdot B_n^L,$$

which implies

$$t_n \cdot \lambda_n = c_{28} \cdot t_n \cdot \frac{1}{L_n} = c_{29} \cdot \frac{1}{\beta_n \cdot C_n \cdot n} \quad \text{and} \quad \beta_n \cdot \sqrt{t_n \cdot \lambda_n} \leq c_{30}.$$

Because of $(\mathbf{w}^{(0)})^{(L)}_{1,1,k} = 0$ we know $F_n(\mathbf{w}^{(0)}) \leq \beta_n^2$. From Lemma 3 and Lemma 5, which we apply with

$$\gamma_n^* = c_{31} + c_{32} \cdot \beta_n \cdot \sqrt{t_n \cdot \lambda_n}, \quad B_n = B_n + c_{32} \cdot \beta_n \cdot \sqrt{t_n \cdot \lambda_n}$$

and

$$A_n = A_n + c_{32} \cdot \beta_n \cdot \sqrt{t_n \cdot \lambda_n}$$

we get that the assumptions of Lemma 4 are satisfied if we set

$$L = c_{33} \cdot K_n^3 \cdot B_n^{2L} \cdot n.$$

(In fact, $c_{34} \cdot K_n^{3/2} \cdot B_n^{2L}$ is here sufficient, but we use a larger value in order to get later that $\lambda_n \cdot D_n^2$ is small.) Hence we have

$$\mathbf{w}^{(t)} \in A := \left\{ \mathbf{w} \, : \, \|\mathbf{w} - \mathbf{w}^{(0)}\| \leq \frac{\sqrt{2\beta_n}}{\sqrt{\beta_n \cdot C_n \cdot n}} \right\}$$

for $t = 1, \dots, t_n$, and

$$F_n(\mathbf{w}^{(t_n)}) \leq \min_{t=0,\dots,t_n-1} F_n(\mathbf{w}^{(t)}).$$

By Lemma 2 and Lemma 3 we know that the assumptions of Lemma 1 are satisfied with $C_n = c_{35} \cdot B_n^{2L}$ and $D_n = c_{36} \cdot K_n^{3/2} \cdot B_n^L$. Application of Lemma 1 with $\delta_n = \frac{c_{37} \cdot \sqrt{\beta_n}}{\sqrt{\beta_n} \cdot C_n \cdot n}$ yields

$$\begin{aligned} F_n(\mathbf{w}^{(t_n)}) &\leq \min_{t=0,\dots,t_n-1} F_n(\mathbf{w}^{(t)}) \\ &\leq F_n(\mathbf{w}^*) + \frac{\|\mathbf{w}^* - \mathbf{w}^{(0)}\|^2}{2 \cdot \lambda_n \cdot t_n} + 24 \cdot \beta_n \cdot C_n \cdot \delta_n^2 + \lambda_n \cdot D_n^2 \\ &\leq F_n(\mathbf{w}^*) + c_{38} \cdot \beta_n \cdot B_n^{2L} \cdot n \cdot \|\mathbf{w}^* - \mathbf{w}^{(0)}\|^2 + c_{39} \cdot \frac{\beta_n}{n}. \end{aligned}$$

□

3.2 Neural network approximation

In the sequel we construct a neural network which approximates a piecewise Taylor polynomial of a function $f : \mathbb{R}^d \to \mathbb{R}$.

Assume that f is (p, C)–smooth for some $p = q + \beta$ where $\beta \in (0, 1]$ and $q \in \mathbb{N}_0$. The multivariate Taylor polynomial of f of degree q around $u \in \mathbb{R}^d$ is defined by

$$(Tf)_{q,u}(x) = \sum_{\substack{j_1,\dots,j_d \in \mathbb{N}_0, \\ j_1+\dots+j_d \leq q}} \frac{\partial^q f}{\partial^{j_1} x^{(1)} \dots \partial^{j_d} x^{(d)}}(u) \cdot (x^{(1)} - u^{(1)})^{j_1} \cdot \dots \cdot (x^{(d)} - u^{(d)})^{j_d}.$$

Since f is (p, C)–smooth, it is possible to show that the error of its Taylor polynomial can be bounded by

$$|f(x) - (Tf)_{q,u}(x)| \leq c_{40} \cdot C \cdot \|x - u\|^p \tag{28}$$

(cf., e.g., Lemma 1 in Kohler (2014)). For functions $f, g : \mathbb{R}^d \to \mathbb{R}$ we have

$$(T(f + g))_{q,u}(x) = (Tf)_{q,u}(x) + (Tg)_{q,u}(x),$$

and if $g : \mathbb{R}^d \to \mathbb{R}$ is a multivariate polynomial of degree q (or less) we have

$$(Tg)_{q,u}(x) = g(x) \quad (x \in \mathbb{R}^d).$$

Next we construct a piecewise Taylor polynomial. To do this, let $A \geq 1$, $K \in \mathbb{N}$ and subdivide $[-A, A]^d$ into K^d many cubes of sidelength

$$\delta = \frac{2A}{K}.$$

Set

$$u_k = -A + k \cdot \frac{2A}{K} \quad (k = 0, \dots, K - 1).$$

Then

$$u_{\mathbf{k}} = (u_{k^{(1)}}, \dots, u_{k^{(d)}}) \quad (\mathbf{k} \in I := \{0, 1, \dots, K - 1\}^d)$$

denote the lower left corners of these cubes.

For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ we write

$$\mathbf{a} \leq \mathbf{b} \quad \text{if} \quad a^{(l)} \leq b^{(l)} \quad \text{for all } l \in \{1, \ldots, d\}$$

and

$$\mathbf{a} < \mathbf{b} \quad \text{if} \quad \mathbf{a} \leq \mathbf{b} \quad \text{and} \quad \mathbf{a} \neq \mathbf{b}.$$

Set

$$[\mathbf{a}, \infty) = [a^{(1)}, \infty) \times \cdots \times [a^{(d)}, \infty) \quad \text{and} \quad [\mathbf{a}, \mathbf{b}) = [a^{(1)}, b^{(1)}) \times \cdots \times [a^{(d)}, b^{(d)}).$$

Our piecewise Taylor polynomial is defined by

$$P(x) = \sum_{\mathbf{k} \in I} P_{\mathbf{k}}(x) \cdot 1_{[u_{\mathbf{k}}, \infty)}(x),$$

where the $P_{\mathbf{k}}$'s are recursively defined by

$$P_0(x) = (Tf)_{q, u_{\mathbf{0}}}(x)$$

and

$$P_{\mathbf{k}}(x) = T\left(f - \sum_{\mathbf{l} \in I \,:\, u_{\mathbf{l}} < u_{\mathbf{k}}} P_{\mathbf{l}}\right)_{q, u_{\mathbf{k}}}(x).$$

As our next lemma shows in this way we define indeed a piecewise Taylor polynomial.

Lemma 6 *Let* $\mathbf{r} \in I$ *and* $x \in [u_{\mathbf{r}}, u_{\mathbf{r}} + \delta \cdot \mathbf{1})$. *Then*

$$P(x) = (Tf)_{q, u_{\mathbf{r}}}(x).$$

Proof. The definition of $P(x)$ and $x \in [u_{\mathbf{r}}, u_{\mathbf{r}} + \delta \cdot \mathbf{1})$ imply

$$P(x) = \sum_{\mathbf{k} \in I} P_{\mathbf{k}}(x) \cdot 1_{[u_{\mathbf{k}}, \infty)}(x) = \sum_{\mathbf{k} \in I \,:\, u_{\mathbf{k}} \leq u_{\mathbf{r}}} P_{\mathbf{k}}(x) = P_{\mathbf{r}}(x) + \sum_{\mathbf{k} \in I \,:\, u_{\mathbf{k}} < u_{\mathbf{r}}} P_{\mathbf{k}}(x).$$

With

$$P_{\mathbf{r}}(x) = T\left(f - \sum_{\mathbf{l} \in I \,:\, u_{\mathbf{l}} < u_{\mathbf{r}}} P_{\mathbf{l}}\right)_{q, u_{\mathbf{r}}}(x) = (Tf)_{q, u_{\mathbf{r}}}(x) - \sum_{\mathbf{l} \in I \,:\, u_{\mathbf{l}} < u_{\mathbf{r}}} P_{\mathbf{l}}(x)$$

we get the assertion. □

Consequently it holds

$$\sup_{x \in [-A, A)^d} |f(x) - P(x)| \leq c_{41} \cdot \frac{1}{K^p}$$

in case that f is (p, C)–smooth (cf., (28)).

In the sequel we will approximate

$$1_{[u_{\mathbf{k}},\infty)}(x)$$

by

$$\prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})),$$

where σ is the logistic squasher and M is a large positive number. This approximation will be bad in case that $x^{(j)}$ is close to $u_{\mathbf{k}}^{(j)}$, and to bound the resulting error in this case the following lemma will be useful.

Lemma 7 *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a (p, C)–smooth function, let $\mathbf{r} \in I$ and let $j \in \{1, \dots, d\}$. Then we have for any $x \in \mathbb{R}^d$ with $\|u_{\mathbf{r}} - x\|_\infty \leq c_{42} \cdot \delta$:*

$$\left| \sum_{\mathbf{k}\in I \,:\, u_{\mathbf{k}} \leq u_{\mathbf{r}} \text{ and } u_{\mathbf{k}}^{(j)} = u_{\mathbf{r}}^{(j)}} P_{\mathbf{k}}(x) \right| \leq \frac{c_{43}}{K^p}.$$

Proof. Let $\mathbf{e}_j$ be the j-th unit vector in $\mathbb{R}^d$. By the proof of Lemma 6 and by (28) we have

$$\begin{aligned}
\left| \sum_{\mathbf{k}\in I \,:\, u_{\mathbf{k}} \leq u_{\mathbf{r}} \text{ and } u_{\mathbf{k}}^{(j)} = u_{\mathbf{r}}^{(j)}} P_{\mathbf{k}}(x) \right| &= \left| \sum_{\mathbf{k}\in I \,:\, u_{\mathbf{k}} \leq u_{\mathbf{r}}} P_{\mathbf{k}}(x) - \sum_{\mathbf{k}\in I \,:\, u_{\mathbf{k}} \leq u_{\mathbf{r}} - \delta \cdot \mathbf{e}_j} P_{\mathbf{k}}(x) \right| \\
&= \left| (Tf)_{q,u_{\mathbf{r}}}(x) - (Tf)_{q,u_{\mathbf{r}}-\delta\cdot\mathbf{e}_j}(x) \right| \\
&\leq \left| (Tf)_{q,u_{\mathbf{r}}}(x) - f(x) \right| + \left| f(x) - (Tf)_{q,u_{\mathbf{r}}-\delta\cdot\mathbf{e}_j}(x) \right| \\
&\leq c_{44} \cdot \|x - u_{\mathbf{r}}\|^p + c_{45} \cdot \|x - (u_{\mathbf{r}} - \delta \cdot \mathbf{e}_j)\|^p \\
&\leq \frac{c_{46}}{K^p},
\end{aligned}$$

where the last inequality followed from $\|u_{\mathbf{r}} - x\|_\infty \leq c_{42} \cdot \delta = c_{42} \cdot (2A)/K$. □

Next we want to approximate

$$P(x) = \sum_{\mathbf{k}\in I} P_{\mathbf{k}}(x) \cdot 1_{[u_{\mathbf{k}},\infty)}(x)$$

by a neural network. Here we consider in an intermediate step

$$\bar{P}(x) = P_0(x) + \sum_{\mathbf{k}\in I, \mathbf{k}\neq 0} P_{\mathbf{k}}(x) \cdot \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})),$$

where the indicator function is approximated by a product of neurons.

Lemma 8 *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a (p, C)–smooth function, let $A \geq 1$, let $K \in \mathbb{N}$ with $K \geq e^p$, let σ be the logistic squasher, and define $P(x)$ and $\bar{P}(x)$ as above. Assume*

$$M \geq K \cdot (\log K)^2.$$

Then

$$\sup_{x \in [-A,A)^d} |P(x) - \bar{P}(x)| \leq c_{47} \cdot \frac{1}{K^p}.$$

Proof. Let $x \in [-A, A]^d$ be arbitrary, and let $\mathbf{r} \in I$ be such that $x \in [u_{\mathbf{r}}, u_{\mathbf{r}} + \delta)$. Since $x \in [u_{\mathbf{r}}, u_{\mathbf{r}} + \delta) \subseteq [u_0, \infty)$ we have

$$\begin{aligned}
&P(x) - \bar{P}(x) \\
&= \sum_{\mathbf{k} \in I, \mathbf{k} \neq 0} P_{\mathbf{k}}(x) \cdot \left(1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \right) \\
&= \sum_{\substack{\mathbf{k} \in I, \mathbf{k} \neq 0: \\ u_{\mathbf{k}}^{(j)} \geq u_{\mathbf{r}}^{(j)} + 2\delta \text{ for some } j \in \{1,\dots,d\}}} P_{\mathbf{k}}(x) \cdot \left(1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \right) \\
&\quad + \sum_{\substack{\mathbf{k} \in I, \mathbf{k} \neq 0: \\ u_{\mathbf{k}}^{(j)} \leq u_{\mathbf{r}}^{(j)} - \delta \text{ for all } j \in \{1,\dots,d\}}} P_{\mathbf{k}}(x) \cdot \left(1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \right) \\
&\quad + \sum_{\substack{\mathbf{k} \in I, \mathbf{k} \neq 0 \,:\, u_{\mathbf{k}}^{(i)} < u_{\mathbf{r}}^{(i)} + 2\delta \text{ for all } i \in \{1,\dots,d\}, \\ u_{\mathbf{k}}^{(j)} > u_{\mathbf{r}}^{(j)} - \delta \text{ for some } j \in \{1,\dots,d\}}} P_{\mathbf{k}}(x) \cdot \left(1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \right) \\
&=: T_{1,n} + T_{2,n} + T_{3,n}.
\end{aligned}$$

If $u_{\mathbf{k}}^{(i)} \geq u_{\mathbf{r}}^{(i)} + 2\delta$ for some $i \in \{1, \dots, d\}$, then $M \geq K \cdot (\log K)^2$ and $\delta = 2A/K \geq 1/K$ imply

$$\begin{aligned}
\left| 1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \right| &= \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \leq \sigma(M \cdot (x^{(i)} - u_{\mathbf{k}}^{(i)})) \\
&\leq \sigma(-(\log K)^2) \leq e^{-(\log K)^2},
\end{aligned}$$

which together with

$$\begin{aligned}
|P_{\mathbf{k}}(x)| &\leq \left| \sum_{\mathbf{l} \in I, u_{\mathbf{l}} \leq u_{\mathbf{k}}} P_{\mathbf{l}}(x) - \sum_{\mathbf{l} \in I, u_{\mathbf{l}} \leq u_{\mathbf{k}} - \delta \cdot \mathbf{e}_1} P_{\mathbf{l}}(x) \right| = \left| T\left(f\right)_{q,u_{\mathbf{k}}}(x) - T\left(f\right)_{q,u_{\mathbf{k}} - \delta \cdot \mathbf{e}_1}(x) \right| \\
&\leq c_{48}
\end{aligned}$$

yields

$$|T_{1,n}| \leq K^d \cdot c_{48} \cdot e^{-(\log K)^2} \leq \frac{c_{49}}{K^p}.$$

If $u_{\mathbf{k}}^{(i)} \leq u_{\mathbf{r}}^{(i)} - \delta$ for all $i \in \{1, \dots, d\}$, then

$$M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)}) \geq M \cdot \delta \geq (\log K)^2 \quad \text{for all } j \in \{1, \dots, d\},$$

hence

$$\begin{aligned}
&\left| 1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \right| \\
&= 1 - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \\
&= 1 - \prod_{j=1}^{d} \frac{1}{1 + e^{-M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})}} \\
&\leq 1 - \prod_{j=1}^{d} \frac{1}{1 + e^{-(\log K)^2}} \\
&= \sum_{l=1}^{d} \left(\prod_{j=1}^{l-1} \frac{1}{1 + e^{-(\log K)^2}} - \prod_{j=1}^{l} \frac{1}{1 + e^{-(\log K)^2}} \right) \\
&\leq d \cdot \left(1 - \frac{1}{1 + e^{-(\log K)^2}} \right) \\
&\leq d \cdot e^{-(\log K)^2},
\end{aligned}$$

which implies

$$|T_{2,n}| \leq K^d \cdot c_{48} \cdot d \cdot e^{-(\log K)^2} \leq \frac{c_{50}}{K^p}.$$

So it remains to bound $|T_{3,n}|$.

$T_{3,n}$ is a sum of less than

$$3^d$$

terms of the form

$$\sum_{\substack{\mathbf{k} \in I, \mathbf{k} \neq 0 \,:\, u_{\mathbf{k}}^{(i)} \leq u_{\mathbf{r}}^{(i)} - \delta \text{ for all } i \in \{1,\dots,d\} \setminus \{j_1,\dots,j_s\}, \\ u_{\mathbf{k}}^{(j_t)} = u_{\mathbf{r}}^{(j_t)} + l_t \cdot \delta \text{ for all } t \in \{1,\dots,s\}}} P_{\mathbf{k}}(x) \cdot \left(1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) \right),$$

where $s \in \{1, \dots, d\}$, $1 \leq j_1 < j_2 < \cdots < j_s \leq d$, $l_1, \dots, l_s \in \{0, 1\}$. The absolute value of the difference of this term and the term

$$\sum_{\substack{\mathbf{k} \in I, \mathbf{k} \neq 0 \,:\, u_{\mathbf{k}}^{(i)} \leq u_{\mathbf{r}}^{(i)} - \delta \text{ for all } i \in \{1,\dots,d\} \setminus \{j_1,\dots,j_s\}, \\ u_{\mathbf{k}}^{(j_t)} = u_{\mathbf{r}}^{(j_t)} + l_t \cdot \delta \text{ for all } t \in \{1,\dots,s\}}} P_{\mathbf{k}}(x) \cdot \left(1_{[u_{\mathbf{k}},\infty)}(x) - \prod_{t=1}^{s} \sigma(M \cdot (x^{(j_t)} - u_{\mathbf{k}}^{(j_t)})) \right)$$

is because of

$$\begin{aligned}
&\left|\prod_{j=1}^{d}\sigma(M\cdot(x^{(j)}-u_{\mathbf{k}}^{(j)}))-\prod_{t=1}^{s}\sigma(M\cdot(x^{(j_t)}-u_{\mathbf{k}}^{(j_t)}))\right| \\
&\leq\left|1-\prod_{j\in\{1,\dots,d\}\setminus\{j_1,\dots,j_s\}}\sigma(M\cdot(x^{(j)}-u_{\mathbf{k}}^{(j)}))\right| \\
&\leq d\cdot e^{-(\log K)^2}
\end{aligned}$$

(which follows as above) bounded from above by c_{51}/K^p.

Hence it suffices to show that

$$\sum_{\substack{\mathbf{k}\in I,\mathbf{k}\neq 0\,:\,u_{\mathbf{k}}^{(i)}\leq u_{\mathbf{r}}^{(i)}-\delta \text{ for all } i\in\{1,\dots,d\}\setminus\{j_1,\dots,j_s\},\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)}+l_t\cdot\delta \text{ for all } t\in\{1,\dots,s\}}} P_{\mathbf{k}}(x)\cdot 1_{[u_{\mathbf{k}},\infty)}(x) \tag{29}$$

and

$$\sum_{\substack{\mathbf{k}\in I,\mathbf{k}\neq 0\,:\,u_{\mathbf{k}}^{(i)}\leq u_{\mathbf{r}}^{(i)}-\delta \text{ for all } i\in\{1,\dots,d\}\setminus\{j_1,\dots,j_s\},\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)}+l_t\cdot\delta \text{ for all } t\in\{1,\dots,s\}}} P_{\mathbf{k}}(x)\cdot\prod_{t=1}^{s}\sigma(M\cdot(x^{(j_t)}-u_{\mathbf{k}}^{(j_t)}))$$

are bounded in absolute value by c_{52}/K^p. Since

$$\begin{aligned}
&\sum_{\substack{\mathbf{k}\in I,\mathbf{k}\neq 0\,:\,u_{\mathbf{k}}^{(i)}\leq u_{\mathbf{r}}^{(i)}-\delta \text{ for all } i\in\{1,\dots,d\}\setminus\{j_1,\dots,j_s\},\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)}+l_t\cdot\delta \text{ for all } t\in\{1,\dots,s\}}} P_{\mathbf{k}}(x)\cdot\prod_{t=1}^{s}\sigma(M\cdot(x^{(j_t)}-u_{\mathbf{k}}^{(j_t)})) \\
&=\prod_{t=1}^{s}\sigma(M\cdot(x^{(j_t)}-u_{\mathbf{r}}^{(j_t)}-l_t\cdot\delta))\cdot\sum_{\substack{\mathbf{k}\in I,\mathbf{k}\neq 0\,:\,u_{\mathbf{k}}^{(i)}\leq u_{\mathbf{r}}^{(i)}-\delta \text{ for all } i\in\{1,\dots,d\}\setminus\{j_1,\dots,j_s\},\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)}+l_t\cdot\delta \text{ for all } t\in\{1,\dots,s\}}} P_{\mathbf{k}}(x)
\end{aligned}$$

for this it suffices to show that terms of the form

$$\sum_{\substack{\mathbf{k}\in I,\mathbf{k}\neq 0\,:\,u_{\mathbf{k}}^{(i)}\leq u_{\mathbf{r}}^{(i)} \text{ for all } i\in\{1,\dots,d\}\setminus\{j_1,\dots,j_s\},\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)} \text{ for all } t\in\{1,\dots,s\}}} P_{\mathbf{k}}(x),$$

where $u_{\mathbf{r}}$ satisfies $\|x-u_{\mathbf{r}}\|_\infty\leq 2\delta$, are bounded in absolute value by c_{53}/K^p, which we do in the sequel. (Here we have used that in (29) w.l.o.g. $1_{[u_{\mathbf{k}},\infty)}(x)=1$ holds because otherwise (29) is zero.)

Since

$$\sum_{\substack{\mathbf{k}\in I,\mathbf{k}\neq 0\,:\,u_{\mathbf{k}}^{(i)}\leq u_{\mathbf{r}}^{(i)} \text{ for all } i\in\{1,\dots,d\}\setminus\{j_1,\dots,j_s\}\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)} \text{ for all } t\in\{1,\dots,s\}}} P_{\mathbf{k}}(x)$$

$$= \sum_{\substack{\mathbf{k}\in I, \mathbf{k}\neq 0\,:\, u_{\mathbf{k}}\leq u_{\mathbf{r}},\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)} \text{ for all } t\in\{2,\dots,s\}}} P_{\mathbf{k}}(x) - \sum_{\substack{\mathbf{k}\in I, \mathbf{k}\neq 0\,:\, u_{\mathbf{k}}\leq u_{\mathbf{r}}-\mathbf{e}_{j_1}\cdot\delta,\\ u_{\mathbf{k}}^{(j_t)}=u_{\mathbf{r}}^{(j_t)} \text{ for all } t\in\{2,\dots,s\}}} P_{\mathbf{k}}(x)$$

we see that the term above is equal to a sum of at most 2^{s-1} terms of the form

$$\sum_{\mathbf{k}\in I\,:\, u_{\mathbf{k}}\leq u_{\mathbf{r}} \text{ and } u_{\mathbf{k}}^{(j)}=u_{\mathbf{r}}^{(j)}} P_{\mathbf{k}}(x),$$

where $\|u_{\mathbf{r}} - x\|_\infty \leq 3\cdot\delta$. From this the assertion follows by an application of Lemma 7.
□

Next we want to approximate

$$\bar{P}(x) = P_0(x) + \sum_{\mathbf{k}\in I, \mathbf{k}\neq 0} P_{\mathbf{k}}(x)\cdot\prod_{j=1}^{d}\sigma(M\cdot(x^{(j)}-u_{\mathbf{k}}^{(j)}))$$

by a neural network. In order to do this, we need to represent monomials by neural networks and need to be able to multiply real numbers by using neural networks. The starting point for both is the following lemma, which is a modification of Theorem 2 in Scarselli and Tsoi (1998).

Lemma 9 *Let σ be the logistic squasher, let $k\in\mathbb{N}$, let $t_\sigma\in\mathbb{R}$ be such that $\sigma^{(k)}(t_\sigma)\neq 0$. Then for any $N\in\mathbb{N}$ with $N>k$ there exist*

$$\alpha_j, \beta_j \in \mathbb{R} \quad (j=0,\dots,N-1)$$

such that

$$f_{net,x^k}(x) = \frac{k!}{\sigma^{(k)}(t_\sigma)}\cdot\sum_{j=0}^{N-1}\alpha_j\cdot\sigma(\beta_j\cdot x + t_\sigma)$$

satisfies for all $A>0$ and all $x\in[-A,A]$:

$$\left|f_{net,x^k}(x) - x^k\right| \leq c_{54}\cdot A^N$$

for some $c_{54} = c_{54}(N,k,\sigma^{(k)}(t_\sigma),\|\sigma^{(N)}\|_\infty,\alpha_0,\dots,\alpha_{N-1},\beta_0,\dots,\beta_{N-1}) \geq 0$.

Proof. Let $\beta_j\in\mathbb{R}$ $(j=0,\dots,N-1)$ be pairwise distinct. Then the vectors

$$\mathbf{v}_l = (\beta_0^l,\dots,\beta_{N-1}^l)^T \quad (l=0,\dots,N-1)$$

are linearly independent since

$$\sum_{l=0}^{N-1}\alpha_l\cdot\mathbf{v}_l = 0$$

implies that the polynomial

$$p(x) = \sum_{l=0}^{N-1}\alpha_l\cdot x^l$$

of degree $N-1$ has the N zero points $\beta_0, \dots, \beta_{N-1}$, which is possible only in case $\alpha_0 = \cdots = \alpha_{N-1} = 0$. Hence we can choose $\alpha_0, \dots, \alpha_{N-1} \in \mathbb{R}$ such that

$$\alpha_0 \cdot \mathbf{v}_0 + \cdots + \alpha_{N-1} \cdot \mathbf{v}_{N-1}$$

is equal to the k-th unit vector in $\mathbb{R}^N$, which implies

$$\sum_{j=0}^{N-1} \alpha_j \cdot \beta_j^l = \begin{cases} 1, & \text{if } l = k \\ 0, & \text{if } l \in \{0, \dots, N-1\} \setminus \{k\}. \end{cases} \tag{30}$$

Using these values for the α_j and β_j, a Taylor expansion of

$$x \mapsto \sigma(\beta_j \cdot x + t_\sigma)$$

around t_σ of order $N-1$ implies

$$\begin{aligned} f_{net,x^k}(x) &= \frac{k!}{\sigma^{(k)}(t_\sigma)} \cdot \sum_{j=0}^{N-1} \alpha_j \cdot \left(\sum_{l=0}^{N-1} \frac{\sigma^{(l)}(t_\sigma)}{l!} \cdot (\beta_j \cdot x)^l + \frac{\sigma^{(N)}(\xi_j)}{l!} \cdot (\beta_j \cdot x)^N \right) \\ &= \frac{k!}{\sigma^{(k)}(t_\sigma)} \cdot \sum_{l=0}^{N-1} \left(\sum_{j=0}^{N-1} \alpha_j \cdot \beta_j^l \right) \cdot \frac{\sigma^{(l)}(t_\sigma)}{l!} \cdot x^l \\ &\quad + \frac{k!}{\sigma^{(k)}(t_\sigma)} \cdot \sum_{j=0}^{N-1} \alpha_j \cdot \frac{\sigma^{(N)}(\xi_j)}{l!} \cdot \beta_j^N \cdot x^N \\ &= x^k + \frac{k!}{\sigma^{(k)}(t_\sigma)} \cdot \sum_{j=0}^{N-1} \alpha_j \cdot \frac{\sigma^{(N)}(\xi_j)}{l!} \cdot \beta_j^N \cdot x^N, \end{aligned}$$

where the last equality follows from (30). Hence

$$\left| f_{net,x^k}(x) - x^k \right| \leq \left| \frac{k!}{\sigma^{(k)}(t_\sigma)} \cdot \sum_{j=0}^{N-1} \alpha_j \cdot \frac{\sigma^{(N)}(\xi_j)}{l!} \cdot \beta_j^N \cdot x^N \right| \leq c_{55} \cdot |x|^N \leq c_{55} \cdot A^N.$$

□

Our next lemma uses Lemma 9 in order to construct a neural network which can multiply two numbers.

Lemma 10 *Let $A > 0$, let $N \in \mathbb{N}$ with $N > 2$ and let f_{net,x^2} be the neural network from Lemma 9 (which has one hidden layer with N neurons). Then*

$$f_{mult}(x, y) = \frac{1}{4} \cdot \left(f_{net,x^2}(x+y) - f_{net,x^2}(x-y) \right)$$

satisfies for all $x, y \in [-A, A]$:

$$|f_{mult}(x, y) - x \cdot y| \leq c_{56} \cdot A^N$$

for some $c_{56} > 0$ (which depends on the constant c_{54} in Lemma 9 and on N).

Proof. By Lemma 9 we get

$$\begin{aligned}
&|f_{mult}(x,y) - x \cdot y| \\
&= \left| \frac{1}{4} \cdot \left(f_{net,x^2}(x+y) - f_{net,x^2}(x-y)\right) - \frac{1}{4} \cdot \left((x+y)^2 - (x-y)^2\right) \right| \\
&\leq \frac{1}{4} \cdot |f_{net,x^2}(x+y) - (x+y)^2| + \frac{1}{4} \cdot |f_{net,x^2}(x-y) - (x-y)^2| \\
&\leq \frac{1}{4} \cdot c_{54} \cdot (2A)^N + \frac{1}{4} \cdot c_{54} \cdot (2A)^N \leq c_{56} \cdot A^N.
\end{aligned}$$

□

Next we extend the multiplication network from the previous lemma in such a way that it can multiply a finite number of real values simultaneously.

Lemma 11 *Let $\sigma(x) = 1/(1+\exp(-x))$, let $0 < A \leq 1$, let $N \in \mathbb{N}$ with $N > 2$ and let $d \in \mathbb{N}$. Assume*

$$c_{56} \cdot 4^{d \cdot N} \cdot A^{N-1} \leq 1, \tag{31}$$

where c_{56} is the constant from Lemma 10. Then there exists a neural network

$$f_{mult,d}$$

with at most $\lceil \log_2 d \rceil$ many layers, at most $2 \cdot N \cdot d$ many neurons and activation function σ, where all the weights are bounded in absolute value by some constant, such that for all $x_1, \dots, x_d \in [-A, A]$ it holds:

$$|f_{mult,d}(x_1, \dots, x_d) - \prod_{j=1}^{d} x_j| \leq c_{57} \cdot A^N,$$

where $c_{57} \geq 0$.

Proof. The proof is a modification of the proof of Lemma 7 in Kohler and Langer (2021).

We set $q = \lceil \log_2(d) \rceil$. The feedforward neural network $f_{mult,d}$ with $L = q$ hidden layers and $r = 2 \cdot N \cdot d$ neurons in each layer is constructed as follows: Set

$$(z_1, \dots, z_{2^q}) = \left(x^{(1)}, x^{(2)}, \dots, x^{(d)}, \underbrace{1, \dots, 1}_{2^q - d \; times} \right). \tag{32}$$

In the construction of our network we will use the network f_{mult} of Lemma 10, which satisfies

$$|f_{mult}(x,y) - x \cdot y| \leq c_{56} \cdot 4^{d \cdot N} \cdot A^N \tag{33}$$

for $x, y \in [-4^d A, 4^d A]$. In the first layer we compute

$$f_{mult}(z_1, z_2), f_{mult}(z_3, z_4), \dots, f_{mult}(z_{2^q - 1}, z_{2^q}),$$

which can be done by one layer of $2 \cdot N \cdot 2^{q-1} \leq 2 \cdot N \cdot d$ neurons. As a result of the first layer we get a vector of outputs which has length 2^{q-1}. Next we pair these outputs and apply f_{mult} again. This procedure is continued until there is only one output left. Therefore we need $L = q$ hidden layers and at most $2 \cdot N \cdot d$ neurons in each layer.

By (31) and (33) we get for any $l \in \{1, \ldots, d\}$ and any $z_1, z_2 \in [-(4^l - 1) \cdot A, (4^l - 1) \cdot A]$

$$|\hat{f}_{mult}(z_1, z_2)| \leq |z_1 \cdot z_2| + |\hat{f}_{mult}(z_1, z_2) - z_1 \cdot z_2| \leq (4^l - 1)^2 A^{2l} + c_{56} \cdot 4^{d \cdot N} \cdot A^N \leq (4^{2l} - 1) \cdot A.$$

From this we get successively that all outputs of layer $l \in \{1, \ldots, q-1\}$ are contained in the interval $[-(4^{2^l} - 1) \cdot A, (4^{2^l} - 1) \cdot A]$, hence in particular they are contained in the interval $[-4^d A, 4^d A]$ where inequality (33) does hold.

Define $\hat{f}_{2^q}$ recursively by

$$\hat{f}_{2^q}(z_1, \ldots, z_{2^q}) = \hat{f}_{mult}(\hat{f}_{2^{q-1}}(z_1, \ldots, z_{2^{q-1}}), \hat{f}_{2^{q-1}}(z_{2^{q-1}+1}, \ldots, z_{2^q}))$$

and

$$\hat{f}_2(z_1, z_2) = \hat{f}_{mult}(z_1, z_2),$$

and set

$$\Delta_l = \sup_{z_1, \ldots, z_{2^l} \in [-A, A]} |\hat{f}_{2^l}(z_1, \ldots, z_{2^l}) - \prod_{i=1}^{2^l} z_i|.$$

Then

$$|\hat{f}_{mult,d}(x_1, \ldots, x_d) - \prod_{i=1}^{d} x_i| \leq \Delta_q$$

and from

$$\Delta_1 \leq c_{56} \cdot 4^{d \cdot N} \cdot A^N$$

(which follows from (33)) and

$$\begin{aligned}
\Delta_q \leq & \sup_{z_1, \ldots, z_{2^q} \in [-A, A]} |\hat{f}_{mult}(\hat{f}_{2^{q-1}}(z_1, \ldots, z_{2^{q-1}}), \hat{f}_{2^{q-1}}(z_{2^{q-1}+1}, \ldots, z_{2^q})) \\
& \qquad - \hat{f}_{2^{q-1}}(z_1, \ldots, z_{2^{q-1}}) \cdot \hat{f}_{2^{q-1}}(z_{2^{q-1}+1}, \ldots, z_{2^q})| \\
& + \sup_{z_1, \ldots, z_{2^q} \in [-A, A]} \Bigg| \hat{f}_{2^{q-1}}(z_1, \ldots, z_{2^{q-1}}) \cdot \hat{f}_{2^{q-1}}(z_{2^{q-1}+1}, \ldots, z_{2^q}) \\
& \qquad - \left(\prod_{i=1}^{2^{q-1}} z_i \right) \cdot \hat{f}_{2^{q-1}}(z_{2^{q-1}+1}, \ldots, z_{2^q}) \Bigg| \\
& + \sup_{z_1, \ldots, z_{2^q} \in [-A, A]} \Bigg| \left(\prod_{i=1}^{2^{q-1}} z_i \right) \cdot \hat{f}_{2^{q-1}}(z_{2^{q-1}+1}, \ldots, z_{2^q}) \\
& \qquad - \left(\prod_{i=1}^{2^{q-1}} z_i \right) \cdot \prod_{i=2^{q-1}+1}^{2^q} z_i \Bigg| \\
\leq & \; c_{56} \cdot 4^{d \cdot N} \cdot A^N + 4^{2^{q-1}} \cdot A \cdot \Delta_{q-1} + A^{2^{q-1}} \cdot \Delta_{q-1}
\end{aligned}$$

$$\leq \quad c_{56} \cdot 4^{d \cdot N} \cdot A^N + 2 \cdot 4^{2^{q-1}} \cdot \Delta_{q-1}$$

(where the second inequality follows from (33) and the fact that all outputs of layer $l \in \{1, \dots, q-1\}$ are contained in the interval $[-4^{2^l} A, 4^{2^l} A]$) we get for $x \in [-A, A]^d$

$$\begin{aligned} |\hat{f}_{mult,d}(\mathbf{x}) - \prod_{i=1}^{d} x^{(i)}| &\leq \Delta_q \\ &\leq c_{56} \cdot 4^{d \cdot N} \cdot A^N \cdot 4^{1+2+\dots+2^{q-1}} \cdot \left(1 + 2 + \dots + 2^{q-1}\right) \\ &\leq c_{56} \cdot 4^{d \cdot N} \cdot A^N \cdot 4^{2d+1} \cdot d \\ &= c_{56} \cdot 4^{d \cdot N + 2d + 1} \cdot d \cdot A^N, \end{aligned}$$

where the last inequality was implied by

$$1 + 2 + \dots + 2^{q-1} = 2^q \leq 2 \cdot d.$$

□

We are now ready to formulate and prove our main result about the approximation of (p, C)–smooth function by deep neural networks with bounded weights.

Theorem 3 *Let $d \in \mathbb{N}$, $p = q + \beta$ where $\beta \in (0, 1]$ and $q \in \mathbb{N}_0$, $C > 0$, $A \geq 1$ and $A_n, B_n, \gamma_n^* \geq 1$. For $L, r, K \in \mathbb{N}$ let $\mathcal{F}$ be the set of all networks $f_{\mathbf{w}}$ defined by (8)–(10) with K_n replaced by r, where the weight vector satisfies*

$$|w_{i,j}^{(0)}| \leq A_n, \quad |w_{i,j}^{(l)}| \leq B_n \quad \text{and} \quad |w_{i,j}^{(L)}| \leq \gamma_n^*$$

for all $l \in \{1, \dots, L-1\}$ and all i, j, and set

$$\mathcal{H} = \left\{ \sum_{k=1}^{K^d} f_k \quad : \quad f_k \in \mathcal{F} \quad (k = 1, \dots, K) \right\}.$$

Let $L, r \in \mathbb{N}$ with

$$L \geq \lceil \log_2(q + d) \rceil \quad \text{and} \quad r \geq 2 \cdot (2p + d) \cdot (q + d),$$

and set

$$A_n = A \cdot K \cdot \log K, \quad B_n = c_{58} \quad \text{and} \quad \gamma_n^* = c_{59} \cdot K^{q+d}.$$

Assume $K \geq c_{60}$ for $c_{60} > 0$ sufficiently large. Then there exists for any (p, C)–smooth $f : \mathbb{R}^d \to \mathbb{R}$ a neural network $h \in \mathcal{H}$ such that

$$\sup_{x \in [-A, A)^d} |f(x) - h(x)| \leq \frac{c_{61}}{K^p}.$$

Proof. Define $P(x)$ and $\bar{P}(x)$ as above with $M = K \cdot (\log K)^2$. Then

$$\sup_{x \in [-A,A)^d} |f(x) - P(x)| \leq c_{62} \cdot \frac{1}{K^p}$$

and

$$\sup_{x \in [-A,A)^d} |P(x) - \bar{P}(x)| \leq c_{63} \cdot \frac{1}{K^p}$$

(cf., Lemma 8) imply that it suffices to show that there exists $h \in \mathcal{H}$ such that

$$\sup_{x \in [-A,A)^d} |h(x) - \bar{P}(x)| \leq c_{64} \cdot \frac{1}{K^p}.$$

Since $\bar{P}(x)$ is a sum of $P_0(x)$ and $(K^d - 1)$ terms of the form

$$P_{\mathbf{k}}(x) \cdot \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})),$$

where each $P_k(x)$ is a multivariate polynomial polynomial of degree q with bounded coefficients, if suffices to show that for all $i_1, \dots, i_q \in \{0, \dots, d\}$, all $u \in [-A, A]^d$ and $z_0 = 1$, $z_j = x^{(j)} - u^{(j)}$ $(j = 1, \dots, d)$ there exists $f_1, f_2 \in \mathcal{F}$ such that

$$\sup_{x \in [-A,A]^d} |\prod_{s=1}^{q} z_{i_s} - f_1(x)| \leq \frac{c_{65}}{K^{p+d}} \tag{34}$$

and

$$\sup_{x \in [-A,A]^d} |\prod_{s=1}^{q} z_{i_s} \cdot \prod_{j=1}^{d} \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)})) - f_2(x)| \leq \frac{c_{66}}{K^{p+d}}. \tag{35}$$

Let $f_{id} = f_{net,x}$ be the network of Lemma 9 which satisfies

$$|f_{id}(x) - x| \leq \frac{c_{67}}{K^{2p+2d}} \tag{36}$$

for all $x \in [-c_{68}/K, c_{68}/K]$ (so we use $N = \lceil 2p + 2d \rceil$). Set

$$f_{id}^{(1)} = f_{id} \quad \text{and} \quad f_{id}^{(l+1)} = f_{id}^{(l)} \circ f_{id}$$

for $l \in \mathbb{N}$. Because of (36) and

$$|f_{id}^{(l+1)}(x) - x| \quad \leq \quad |f_{id}^{(l+1)}(x) - f_{id}^{(l)}(x)| + |f_{id}^{(l)}(x) - x|$$

an easy induction shows

$$|f_{id}^{(l)}(x) - x| \leq \frac{c_{69,l}}{K^{2p+2d}} \tag{37}$$

for all $x \in [-c_{70}/K, c_{70}/K]$.

Furthermore, let $f_{mult,q}$ and $f_{mult,q+d}$ be the networks from Lemma 11 which satisfy

$$|f_{mult,q}(x_1,\dots,x_q) - \prod_{j=1}^{q} x_j| \leq \frac{c_{71}}{K^{2p+d}},$$

for all $x_1,\dots,x_q \in [-c_{72}/K, c_{72}/K]$ and

$$|f_{mult,q+d}(x_1,\dots,x_{q+d}) - \prod_{j=1}^{q+d} x_j| \leq \frac{c_{73}}{K^{2p+2d}},$$

for all $x_1,\dots,x_{q+d} \in [-c_{74}/K, c_{74}/K]$.

Then we define

$$f_1(x) = K^q \cdot f_{id}^{(L-\lceil \log_2 q \rceil)}(f_{mult,q}(z_{i_1}/K,\dots,z_{i_q}/K))$$

and

$$\begin{aligned} f_2(x) \quad &= \quad K^{q+d} \cdot f_{id}^{(L-\lceil \log_2 (q+d) \rceil)}\Bigg(f_{mult,q+d}\Bigg(f_{id}(z_{i_1}/K),\dots,f_{id}(z_{i_q}/K), \\ & \qquad \frac{1}{K}\cdot\sigma(M\cdot(x^{(1)}-u^{(1)})),\dots,\frac{1}{K}\cdot\sigma(M\cdot(x^{(d)}-u^{(d)}))\Bigg)\Bigg). \end{aligned}$$

Then f_1 and f_2 are both contained in $\mathcal{F}$. Using (34) and (37) we get

$$\begin{aligned} & |\prod_{s=1}^{q} z_{i_s} - f_1(x)| \\ & \leq K^q \cdot \left|\prod_{s=1}^{q} z_{i_s}/K^q - f_{id}^{(L-\lceil \log_2 q \rceil)}(f_{mult,q}(z_{i_1}/K,\dots,z_{i_q}/K))\right| \\ & \leq K^q \cdot \left(\left|\prod_{s=1}^{q} z_{i_s}/K^q - f_{mult,q}(z_{i_1}/K,\dots,z_{i_q}/K)\right| + \frac{c_{75}}{K^{2p+2d}}\right) \\ & \leq K^q \cdot \left(\frac{c_{76}}{K^{2p+2d}} + \frac{c_{75}}{K^{2p+2d}}\right) \leq \frac{c_{77}}{K^{p+d}} \end{aligned}$$

and

$$\begin{aligned} & |\prod_{s=1}^{q} z_{i_s} \cdot \prod_{j=1}^{d} \sigma(M\cdot(x^{(j)}-u_{\mathbf{k}}^{(j)})) - f_2(x)| \\ & = K^{q+d} \cdot \left|\prod_{s=1}^{q} z_{i_s}/K \cdot \prod_{j=1}^{d} \frac{1}{K}\cdot\sigma(M\cdot(x^{(j)}-u_{\mathbf{k}}^{(j)}))\right. \\ & \quad -f_{id}^{(L-\lceil \log_2 (q+d) \rceil)}\Bigg(f_{mult,q+d}\Bigg(f_{id}(z_{i_1}/K), \end{aligned}$$

$$\ldots, f_{id}(z_{i_q}/K), \sigma(M \cdot (x^{(1)} - u^{(1)}))/K, \ldots, \sigma(M \cdot (x^{(d)} - u^{(d)})/K)\Bigg)\Bigg)\Bigg|$$

$$\leq K^{q+d} \cdot \Bigg(\frac{c_{78}}{K^{2p+2d}} + \Bigg|\prod_{s=1}^{q} f_{id}(z_{i_s}/K) \cdot \prod_{j=1}^{d} \frac{1}{K} \cdot \sigma(M \cdot (x^{(j)} - u_{\mathbf{k}}^{(j)}))$$

$$-f_{mult,q+d}\Bigg(f_{id}(z_{i_1}/K), \ldots, f_{id}(z_{i_q}/K), \sigma(M \cdot (x^{(1)} - u^{(1)}))/K,$$

$$\ldots, \sigma(M \cdot (x^{(d)} - u^{(d)})/K)\Bigg)\Bigg|\Bigg)$$

$$\leq K^{q+d} \cdot \left(\frac{c_{78}}{K^{2p+2d}} + \frac{c_{79}}{K^{2p+2d}}\right) \leq \frac{c_{80}}{K^{p+d}},$$

which implies the assertion. □

3.3 Neural network generalization

In order to control the generalization error of our over-parameterized spcaes of deep neural networks we use the following metric entropy bound.

Lemma 12 *Let $\alpha, \beta \geq 1$ and let $A, B, C \geq 1$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be k-times differentiable such that all derivatives up to order k are bounded on $\mathbb{R}$. Let $\mathcal{F}$ be the set of all functions $f_{\mathbf{w}}$ defined by (8)–(10) where the weight vector $\mathbf{w}$ satisfies*

$$\sum_{j=1}^{K_n} |w_{1,1,j}^{(L)}| \leq C, \tag{38}$$

$$|w_{k,i,j}^{(l)}| \leq B \quad (k \in \{1, \ldots, K_n\}, i, j \in \{1, \ldots, r\}, l \in \{1, \ldots, L-1\}) \tag{39}$$

and

$$|w_{k,i,j}^{(0)}| \leq A \quad (k \in \{1, \ldots, K_n\}, i \in \{1, \ldots, r\}, j \in \{1, \ldots, d\}). \tag{40}$$

Then we have for any $1 \leq p < \infty$, $0 < \epsilon < 1$ and $x_1^n \in \mathbb{R}^d$

$$\mathcal{N}_p\left(\epsilon, \{T_\beta f \cdot 1_{[-\alpha,\alpha]^d} \; : \; f \in \mathcal{F}\}, x_1^n\right)$$
$$\leq \left(c_{81} \cdot \frac{\beta^p}{\epsilon^p}\right)^{c_{82} \cdot \alpha^d \cdot B^{(L-1)\cdot d} \cdot A^d \cdot \left(\frac{C}{\epsilon}\right)^{d/k} + c_{83}}.$$

Proof. This result follows from Lemma 4 in Drews and Kohler (2024). For the sake of completeness we repeat the proof below.

In the *first step* of the proof we show for any $f_{\mathbf{w}} \in \mathcal{F}$, any $x \in \mathbb{R}^d$ and any $s_1, \ldots, s_k \in \{1, \ldots, d\}$

$$\left|\frac{\partial^k f_{\mathbf{w}}}{\partial x^{(s_1)} \ldots \partial x^{(s_k)}}(x)\right| \leq c_{84} \cdot C \cdot B^{(L-1)\cdot k} \cdot A^k =: c. \tag{41}$$

The definition of $f_{\mathbf{w}}$ implies

$$\frac{\partial^k f_{\mathbf{w}}}{\partial x^{(s_1)} \dots \partial x^{(s_k)}}(x) = \sum_{j=1}^{K_n} w_{1,1,j}^{(L)} \cdot \frac{\partial^k f_{j,1}^{(L)}(x)}{\partial x^{(s_1)} \dots \partial x^{(s_k)}}(x),$$

hence (41) is implied by

$$\left| \frac{\partial^k f_{j,1}^{(L)}}{\partial x^{(s_1)} \dots \partial x^{(s_k)}}(x) \right| \leq c_{85} \cdot B^{(L-1)\cdot k} \cdot A^k. \tag{42}$$

We have

$$\begin{aligned} \frac{\partial f_{k,i}^{(l)}}{\partial x^{(s)}}(x) &= \sigma' \left(\sum_{t=1}^{r} w_{k,i,t}^{(l-1)} \cdot f_{k,t}^{(l-1)}(x) + w_{k,i,0}^{(l-1)} \right) \cdot \sum_{j=1}^{r} w_{k,i,j}^{(l-1)} \cdot \frac{\partial f_{k,j}^{(l-1)}}{\partial x^{(s)}}(x) \\ &= \sum_{j=1}^{r} w_{k,i,j}^{(l-1)} \cdot \sigma' \left(\sum_{t=1}^{r} w_{k,i,t}^{(l-1)} \cdot f_{k,t}^{(l-1)}(x) + w_{k,i,0}^{(l-1)} \right) \cdot \frac{\partial f_{k,j}^{(l-1)}}{\partial x^{(s)}}(x) \end{aligned}$$

and

$$\frac{\partial f_{k,i}^{(1)}}{\partial x^{(s)}}(x) = \sigma' \left(\sum_{j=1}^{d} w_{k,i,j}^{(0)} \cdot x^{(j)} + w_{k,i,0}^{(0)} \right) \cdot w_{k,i,s}^{(0)}.$$

By the product rule of derivation we can conclude for $l > 1$ that

$$\frac{\partial^k f_{k,i}^{(l)}}{\partial x^{(s_1)} \dots \partial x^{(s_k)}}(x) \tag{43}$$

is a sum of at most $r \cdot (r+k)^{k-1}$ terms of the form

$$\begin{aligned} w \cdot \sigma^{(s)} &\left(\sum_{j=1}^{r} w_{k,i,j}^{(l-1)} \cdot f_{k,j}^{(l-1)}(x) + w_{k,i,0}^{(l-1)} \right) \\ &\qquad \cdot \frac{\partial^{t_1} f_{k,j_1}^{(l-1)}}{\partial x^{(r_{1,1})} \dots \partial x^{(r_{1,t_1})}}(x) \cdot \dots \cdot \frac{\partial^{t_s} f_{k,j_s}^{(l-1)}}{\partial x^{(r_{s,1})} \dots \partial x^{(r_{s,t_s})}}(x) \end{aligned}$$

where we have $s \in \{1, \dots, k\}$, $|w| \leq B^s$ and $t_1 + \dots + t_s = k$. Furthermore

$$\frac{\partial^k f_{k,i}^{(1)}}{\partial x^{(s_1)} \dots \partial x^{(s_k)}}(x)$$

is a given by

$$\prod_{j=1}^{k} w_{k,i,s_j}^{(0)} \cdot \sigma^{(k)} \left(\sum_{t=1}^{d} w_{k,i,t}^{(0)} \cdot x^{(t)} + w_{k,i,0}^{(0)} \right).$$

Because of the boundedness of the derivatives of σ we can conclude from (40)

$$\left| \frac{\partial^k f_{k,i}^{(1)}}{\partial x^{(s_1)} \dots \partial x^{(s_k)}}(x) \right| \leq c_{86} \cdot A^k$$

for all $k \in \mathbb{N}$ and $s_1, \dots, s_k \in \{1, \dots, d\}$.

Recursively we can conclude from the above representation of (43) that we have

$$\left| \frac{\partial^k f_{k,i}^{(l)}}{\partial x^{(s_1)} \dots \partial x^{(s_k)}}(x) \right| \leq c_{86,r,l,k} \cdot B^{(l-1)\cdot k} \cdot A^k.$$

Setting $l = L$ we get (42).

In the *second step* of the proof we show

$$\mathcal{N}_p\left(\epsilon, \{T_\beta f \cdot 1_{[-\alpha,\alpha]^d} \,:\, f \in \mathcal{F}\}, x_1^n\right) \leq \mathcal{N}_p\left(\frac{\epsilon}{2}, T_\beta \mathcal{G} \circ \Pi, x_1^n\right), \tag{44}$$

where $\mathcal{G}$ is the set of all polynomials of degree less than or equal to $k-1$ which vanish outside of $[-\alpha,\alpha]^d$ and Π is the family of all partitions of $\mathbb{R}^d$ which consist of a partition of $[-\alpha,\alpha]^d$ into

$$K = \left(\left\lceil \frac{2\cdot\alpha}{\left(c_{87}\cdot\frac{\epsilon}{c}\right)^{1/k}} \right\rceil\right)^d$$

many cubes of sidelenght at most

$$\left(c_{87}\cdot\frac{\epsilon}{c}\right)^{1/k}$$

where $c_{87} = c_{87}(d,k) > 0$ is a suitable small constant greater than zero, and the additional set $\mathbb{R}^d \setminus [-\alpha,\alpha]^d$.

A standard bound on the remainder of a multivariate Taylor polynomial together with (41) shows that for each $f_{\mathbf{w}}$ we can find $g \in \mathcal{G} \circ \Pi$ such that

$$|f_{\mathbf{w}}(x) - g(x)| \leq \frac{\epsilon}{2}$$

holds for all $x \in [-\alpha,\alpha]^d$, which implies (44).

In the *third step* of the proof we show the assertion of Lemma 12. Since $\mathcal{G} \circ \Pi$ is a linear vector space of dimension less than or equal to

$$c_{88}\cdot\alpha^d\cdot\left(\frac{c}{\epsilon}\right)^{d/k}$$

we conclude from Theorem 9.4 and Theorem 9.5 in Györfi et al. (2002),

$$\mathcal{N}_p\left(\frac{\epsilon}{2}, T_\beta\mathcal{G}\circ\Pi, x_1^n\right) \leq 3\left(\frac{2e(2\beta)^p}{(\epsilon/2)^p}\log\left(\frac{3e(2\beta)^p}{(\epsilon/2)^p}\right)\right)^{c_{88}\cdot\alpha^d\cdot\left(\frac{c}{\epsilon}\right)^{d/k}+1}.$$

Together with (44) this implies the assertion. □

3.4 Proof of Theorem 1

W.l.o.g. we assume throughout the proof that n is sufficiently large and that $\|m\|_\infty \leq \beta_n$ holds. Let $A > 0$ with $supp(X) \subseteq [-A, A]^d$. Set

$$\tilde{K}_n = \left\lceil c_{89} \cdot n^{\frac{d}{2p+d}} \right\rceil$$

and

$$N_n = \left\lceil c_{90} \cdot n^{3+\frac{d}{2p+d}} \right\rceil$$

and let $\mathbf{w}^*$ be a weight vector of a neural networks where the results of $N_n \cdot \tilde{K}_n$ in parallel computed neural networks with L hidden layers and r neurons per layer are computed such that the corresponding network

$$f_{\mathbf{w}^*}(x) = \sum_{k=1}^{N_n \cdot \tilde{K}_n} (\mathbf{w}^*)_{1,1,k} \cdot f^{(L)}_{\mathbf{w}^*,k,1}(x)$$

satisfies

$$\sup_{x \in [-A,A]^d} |f_{\mathbf{w}^*}(x) - m(x)| \leq \frac{c_{91}}{\tilde{K}_n^{p/d}} \tag{45}$$

and

$$|(\mathbf{w}^*)_{1,1,k}| \leq \frac{c_{92} \cdot \tilde{K}_n^{(q+d)/d}}{N_n} \quad (k = 1, \dots, N_n \cdot \tilde{K}_n).$$

Note that such a network exists according Theorem 3 if we repeat in the outer sum of the function space $\mathcal{H}$ each of the f_k's in Theorem 3 N_n–times with outer weights divided by N_n. Set

$$\epsilon_n = \frac{c_{93}}{n \cdot \sqrt{N_n \cdot \tilde{K}_n}} \geq \frac{c_{94}}{n^4}$$

where the last inequality holds because of $p \geq 1/2$. Let A_n be the event that firstly the weight vector $\mathbf{w}^{(0)}$ satisfies

$$|(\mathbf{w}^{(0)})^{(l)}_{j_s,k,i} - (\mathbf{w}^*)^{(l)}_{s,k,i}| \leq \epsilon_n \quad \text{for all } l \in \{0, \dots, L-1\}, s \in \{1, \dots, N_n \cdot \tilde{K}_n\}$$

for some pairwise distinct $j_1, \dots, j_{N_n \cdot \tilde{K}_n} \in \{1, \dots, K_n\}$ and such that secondly

$$\max_{i=1,\dots,n} |Y_i| \leq \sqrt{\beta_n}$$

holds.

We decompose the L_2 error of m_n in a sum of several terms. Set

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

We have

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

$$
\begin{aligned}
&= \left(\mathbf{E}\left\{|m_n(X)-Y|^2|\mathcal{D}_n\right\} - \mathbf{E}\{|m(X)-Y|^2\}\right)\cdot 1_{A_n} + \int |m_n(x)-m(x)|^2\mathbf{P}_X(dx)\cdot 1_{A_n^c} \\
&= \Big[\mathbf{E}\left\{|m_n(X)-Y|^2|\mathcal{D}_n\right\} - \mathbf{E}\{|m(X)-Y|^2\} \\
&\qquad\qquad - \left(\mathbf{E}\left\{|m_n(X)-T_{\beta_n}Y|^2|\mathcal{D}_n\right\} - \mathbf{E}\{|m_{\beta_n}(X)-T_{\beta_n}Y|^2\}\right)\Big]\cdot 1_{A_n} \\
&\quad + \Big[\mathbf{E}\left\{|m_n(X)-T_{\beta_n}Y|^2|\mathcal{D}_n\right\} - \mathbf{E}\{|m_{\beta_n}(X)-T_{\beta_n}Y|^2\} \\
&\qquad\qquad -2\cdot\frac{1}{n}\sum_{i=1}^n \left(|m_n(X_i)-T_{\beta_n}Y_i|^2 - |m_{\beta_n}(X_i)-T_{\beta_n}Y_i|^2\right)\Big]\cdot 1_{A_n} \\
&\quad + \Big[2\cdot\frac{1}{n}\sum_{i=1}^n |m_n(X_i)-T_{\beta_n}Y_i|^2 - 2\cdot\frac{1}{n}\sum_{i=1}^n |m_{\beta_n}(X_i)-T_{\beta_n}Y_i|^2 \\
&\qquad\qquad - \left(2\cdot\frac{1}{n}\sum_{i=1}^n |m_n(X_i)-Y_i|^2 - 2\cdot\frac{1}{n}\sum_{i=1}^n |m(X_i)-Y_i|^2\right)\Big]\cdot 1_{A_n} \\
&\quad + \Big[2\cdot\frac{1}{n}\sum_{i=1}^n |m_n(X_i)-Y_i|^2 - 2\cdot\frac{1}{n}\sum_{i=1}^n |m(X_i)-Y_i|^2\Big]\cdot 1_{A_n} \\
&\quad + \int |m_n(x)-m(x)|^2\mathbf{P}_X(dx)\cdot 1_{A_n^c} \\
&=: \sum_{j=1}^5 T_{j,n}.
\end{aligned}
$$

In the remainder of the proof we bound

$$\mathbf{E}T_{j,n}$$

for $j \in \{1,\dots,5\}$.

In the *first step of the proof* we show

$$\mathbf{E}T_{j,n} \le c_{95}\cdot\frac{\log n}{n} \quad \text{for } j\in\{1,3\}.$$

This follows as in the proof of Lemma 1 in Bauer and Kohler (2019).

In the *second step of the proof* we show

$$\mathbf{E}T_{5,n} \le c_{96}\cdot\frac{(\log n)^2}{n}.$$

The definition of m_n implies $\int |m_n(x)-m(x)|^2\mathbf{P}_X(dx) \le 4\cdot c_3^2\cdot(\log n)^2$, hence it suffices to show

$$\mathbf{P}(A_n^c) \le \frac{c_{97}}{n^2}. \tag{46}$$

To do this, we consider a sequential choice of the weights of the K_n fully connected neural networks. The probability that the weights in the first of these networks differ in all components at most by ϵ_n from $(\mathbf{w}^*)_{1,i,j}^{(l)}$ $(l = 0, \dots, L-1)$ is for large n bounded from below by

$$\left(\frac{c_{94}}{2 \cdot c_1 \cdot n^4} \right)^{r \cdot (r+1) \cdot (L-1)} \cdot \left(\frac{1}{2 \cdot c_2 \cdot (\log n) \cdot n^\tau \cdot n^4} \right)^{r \cdot (d+1)}$$
$$\geq n^{-r \cdot (r+1) \cdot (L-1) \cdot 4 - r \cdot 4 \cdot (d+1) - r \cdot 4 \cdot \tau - 0.5}.$$

Hence probability that none of the first $n^{r \cdot (r+1) \cdot (L-1) \cdot 4 + r \cdot 4 \cdot (d+1) + r \cdot 4 \cdot \tau + 1}$ neural networks satisfies this condition is for large n bounded above by

$$(1 - n^{-r \cdot (r+1) \cdot (L-1) \cdot 4 - r \cdot 4 \cdot (d+1) - r \cdot 4 \cdot \tau - 0.5})^{n^{r \cdot (r+1) \cdot (L-1) \cdot 4 + r \cdot 4 \cdot (d+1) + r \cdot 4 \cdot \tau + 1}}$$
$$\leq \left(\exp \left(-n^{-r \cdot (r+1) \cdot (L-1) \cdot 4 - r \cdot 4 \cdot (d+1) - r \cdot 4 \cdot \tau - 0.5} \right) \right)^{n^{r \cdot (r+1) \cdot (L-1) \cdot 4 + r \cdot 4 \cdot (d+1) + r \cdot 4 \cdot \tau + 1}}$$
$$= \exp(-n^{0.5}).$$

Since we have $K_n \geq n^{r \cdot (r+1) \cdot (L-1) \cdot 4 + r \cdot 4 \cdot (d+1) + r \cdot 4 \cdot \tau + 1} \cdot N_n \cdot \tilde{K}_n$ for n large we can successively use the same construction for all of $N_n \cdot \tilde{K}_n$ weights and we can conclude: The probability that there exists $k \in \{1, \dots, N_n \cdot \tilde{K}_n\}$ such that none of the K_n weight vectors of the fully connected neural network differs by at most ϵ_n from $((\mathbf{w}^*)_{i,j,k}^{(l)})_{i,j,l}$ is for large n bounded from above by

$$N_n \cdot \tilde{K}_n \cdot \exp(-n^{0.5}) \leq c_{98} \cdot n^5 \cdot \exp(-n^{0.5}) \leq \frac{c_{99}}{n^2}.$$

This implies for large n

$$\begin{aligned} \mathbf{P}(A_n^c) &\leq \frac{c_{99}}{n^2} + \mathbf{P}\{\max_{i=1,\dots,n} |Y_i| > \sqrt{\beta_n}\} \leq \frac{c_{99}}{n^2} + n \cdot \mathbf{P}\{|Y| > \sqrt{\beta_n}\} \\ &\leq \frac{c_{99}}{n^2} + n \cdot \frac{\mathbf{E}\{\exp(c_4 \cdot Y^2)}{\exp(c_4 \cdot \beta_n)} \leq \frac{c_{97}}{n^2}, \end{aligned}$$

where the last inequality holds because of (15) and $c_3 \cdot c_4 \geq 2$.

Let $\epsilon > 0$ be arbitrary. In the *third step of the proof* we show

$$\mathbf{E}T_{2,n} \leq c_{100} \cdot \frac{n^{\tau \cdot d + \epsilon}}{n}.$$

Let $\mathcal{W}_n$ be the set of all weight vectors $(w_{i,j,k}^{(l)})_{i,j,k,l}$ which satisfy

$$|w_{1,1,k}^{(L)}| \leq c_{101} \quad (k = 1, \dots, K_n),$$

$$|w_{i,j,k}^{(l)}| \leq c_{102} \quad (l = 1, \dots, L-1)$$

and

$$|w_{i,j,k}^{(0)}| \leq (c_2 + c_{103}) \cdot (\log n) \cdot n^\tau.$$

By Lemma 4, Lemma 5 and Lemma 6 we can conclude that on A_n we have

$$\|\mathbf{w}^{(t)} - \mathbf{w}^{(0)}\| \leq c_{104} \quad (t = 1, \dots, t_n). \tag{47}$$

This follows from the fact that on A_n we have

$$F_n(\mathbf{w}^{(0)}) = \frac{1}{n}\sum_{i=1}^n Y_i^2 \leq \beta_n$$

and that

$$2 \cdot t_n \cdot \lambda_n \cdot \beta_n \leq c_{105}.$$

Together with the initial choice of $\mathbf{w}^{(0)}$ this implies that on A_n we have

$$\mathbf{w}^{(t)} \in \mathcal{W}_n \quad (t = 0, \dots, t_n).$$

Hence, for any $u > 0$ we get

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

where

$$\mathcal{F}_n = \{T_{\beta_n}f_{\mathbf{w}} \quad : \quad \mathbf{w} \in \mathcal{W}_n\}.$$

By Lemma 12 we get

$$\begin{aligned}
&\mathcal{N}_1\left(\delta, \left\{\frac{1}{\beta_n}\cdot f : f \in \mathcal{F}_n\right\}, x_1^n\right) \leq \mathcal{N}_1\left(\delta\cdot\beta_n, \mathcal{F}_n, x_1^n\right) \\
&\leq \left(\frac{c_{106}}{\delta}\right)^{c_{107}\cdot(\log n)^d n^{\tau\cdot d}\cdot(c_{108})^{(L-1)\cdot d}\cdot\left(\frac{K_n\cdot c_{109}}{\beta_n\cdot\delta}\right)^{d/k}+c_{110}}.
\end{aligned}$$

By choosing k large enough we get for $\delta > 1/n^2$

$$\mathcal{N}_1\left(\delta, \left\{\frac{1}{\beta_n}\cdot f : f \in \mathcal{F}_n\right\}, x_1^n\right) \leq c_{111}\cdot n^{c_{112}\cdot n^{\tau\cdot d+\epsilon/2}}.$$

This together with Theorem 11.4 in Györfi et al. (2002) leads for $u \geq 1/n$ to

$$\mathbf{P}\{T_{2,n} > u\} \leq 14\cdot c_{111}\cdot n^{c_{112}\cdot n^{\tau\cdot d+\epsilon/2}}\cdot\exp\left(-\frac{n}{5136\cdot\beta_n^2}\cdot u\right).$$

For $\epsilon_n \geq 1/n$ we can conclude

$$\mathbf{E}\{T_{2,n}\} \quad\leq\quad \epsilon_n + \int_{\epsilon_n}^\infty \mathbf{P}\{T_{2,n} > u\}\,du$$

$$\leq \quad \epsilon_n + 14 \cdot c_{111} \cdot n^{c_{112} \cdot n^{\tau \cdot d + \epsilon/2}} \cdot \exp\left(-\frac{n}{5136 \cdot \beta_n^2} \cdot \epsilon_n\right) \cdot \frac{5136 \cdot \beta_n^2}{n}.$$

Setting

$$\epsilon_n = \frac{5136 \cdot \beta_n^2}{n} \cdot c_{112} \cdot n^{\tau \cdot d + \epsilon/2} \cdot \log n$$

yields the assertion of the fourth step of the proof.

In the *fourth step of the proof* we show

$$\mathbf{E}\{T_{4,n}\} \leq c_{113} \cdot n^{-\frac{2p}{2p+d}}.$$

Using

$$|T_{\beta_n} z - y| \leq |z - y| \quad \text{for } |y| \leq \beta_n$$

we get

$$\begin{aligned}
& T_{4,n}/2 \\
&= \left[\frac{1}{n}\sum_{i=1}^{n} |m_n(X_i) - Y_i|^2 - \frac{1}{n}\sum_{i=1}^{n} |m(X_i) - Y_i|^2\right] \cdot 1_{A_n} \\
&\leq \left[\frac{1}{n}\sum_{i=1}^{n} |f_{\mathbf{w}^{(t_n)}}(X_i) - Y_i|^2 - \frac{1}{n}\sum_{i=1}^{n} |m(X_i) - Y_i|^2\right] \cdot 1_{A_n} \\
&\leq \left[F_n(\mathbf{w}^{(t_n)}) - \frac{1}{n}\sum_{i=1}^{n} |m(X_i) - Y_i|^2\right] \cdot 1_{A_n}.
\end{aligned}$$

On A_n we have

$$\begin{aligned}
\|\mathbf{w}^* - \mathbf{w}^{(0)}\|^2 \quad &\leq \quad \sum_{k=1}^{K_n} |(\mathbf{w}^*)_{1,1,k}^{(L)}|^2 + N_n \cdot \tilde{K}_n \cdot L \cdot (r \cdot (r+d))^L \cdot \epsilon_n^2 \\
&\leq \quad N_n \cdot \tilde{K}_n \cdot \left(\frac{c_{114} \cdot \tilde{K}_n^{(q+d)/d}}{N_n}\right)^2 + c_{115} \cdot N_n \cdot \tilde{K}_n \cdot \epsilon_n^2 \leq \frac{c_{116}}{n^2}.
\end{aligned}$$

Application of Theorem 2 yields

$$\begin{aligned}
& T_{4,n}/2 \\
&\leq \Bigg(\frac{1}{n}\sum_{i=1}^{n} |f_{\mathbf{w}^*}(X_i) - Y_i|^2 + c_{117} \cdot (\log n) \cdot n \cdot \|\mathbf{w}^* - \mathbf{w}^{(0)}\|^2 + c_{118} \cdot \frac{\log n}{n} \\
&\qquad -\frac{1}{n}\sum_{i=1}^{n} |m(X_i) - Y_i|^2\Bigg) \cdot 1_{A_n} \\
&\leq \Bigg(\frac{1}{n}\sum_{i=1}^{n} |f_{\mathbf{w}^*}(X_i) - Y_i|^2 + c_{119} \cdot (\log n) \cdot n \cdot \frac{1}{n^2} + c_{120} \cdot \frac{\log n}{n}
\end{aligned}$$

$$-\frac{1}{n}\sum_{i=1}^{n}|m(X_i)-Y_i|^2\Bigg) + \frac{1}{n}\sum_{i=1}^{n}|m(X_i)-Y_i|^2 \cdot 1_{A_n^c}.$$

Hence

$$\begin{aligned}
&\mathbf{E}\{T_{4,n}\}\\
&\leq 2 \cdot \int |f_{\mathbf{w}^*}(x)-m(x)|^2 \mathbf{P}_X(dx) + c_{119} \cdot (\log n) \cdot n \cdot \frac{1}{n^2} + c_{120} \cdot \frac{\log n}{n}\\
&\quad + \sqrt{\mathbf{E}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}|m(X_i)-Y_i|^2\right|^2\right\}} \cdot \sqrt{\mathbf{P}(A_n^c)}\\
&\leq 2 \cdot \int |f_{\mathbf{w}^*}(x)-m(x)|^2 \mathbf{P}_X(dx) + c_{121} \cdot \log n \cdot \frac{1}{n}.
\end{aligned}$$

Application of (45) yields the assertion. □

References

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

[2] Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning* **14**, pp. 115-133.

[3] Bartlett, P., Harvey, N., Liaw, C., and Mehrabian, A. (2019). Nearly-tight VC-dimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research* **20**, pp. 1-17.

[4] Bauer, B., and Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Annals of Statistics* **4**, pp. 2261–2285.

[5] Braun, A., Kohler, M., Langer, S., and Walk, H. (2023). Convergence rates for shallow neural networks learned by gradient descent. *Bernoulli* **30**, pp. 475-502.

[6] Chizat, L. and Bach, F. (2018). On the global convergence of gradient descent for over-parameterized models using optimal transport. Preprint, *arXiv: 1805.09545.*

[7] Choromanska, Anna, Henaff, Mikael, Mathieu, Michael, Arous, Gerard Ben, and LeCun, Yann (2015). The loss surfaces of multilayer networks. In AISTATS , 2015.

[8] Drews, S., and Kohler, M. (2023). Analysis of the expected L_2 error of an over-parametrized deep neural network estimate learned by gradient descent without regularization. Preprint.

[9] Drews, S., and Kohler, M. (2024). On the universal consistency of an over-parametrized deep neural network estimate learned by gradient descent. *Annals of the Institue of Statistical Mathematics* **76**, pp. 361-391.

[10] Du, S., Lee, J., Li, H., Wang, L., und Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. International Conference on Machine Learning, Preprint, *arXiv: 1811.03804.*

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

[12] Golowich, N., Rakhlin, A., and Shamir, O. (2019). Size-Independent sample complexity of neural networks. Preprint, *arXiv: 1712.06541.*

[13] Györfi, L., Kohler, M., Krzyżak, A. and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression.* Springer Series in Statistics, Springer-Verlag, New York.

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

[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] Hanin, B., and Nica, M. (2019). Finite depth and width corrections to the neural tangent kernel. *arXiv: 1909.05989.*

[17] Huang, G. B., Chen, L., and Siew, C.-K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks* **17**, pp. 879-892.

[18] Imaizumi, M., and Fukamizu, K. (2018). Deep neural networks learn non-smooth functions effectively. *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019).* Naha, Okinawa, Japan.

[19] Jacot, A., Gabriel, F., and Hongler, C. (2020). Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv: 1806.07572v4.*

[20] Kawaguchi, K, and Huang, J. (2019). Gradient descent finds global minima for generalizable deep neural networks of practical sizes. *57th IEEE Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, pp. 92-99.

[21] Kim, Y. (2014). Convolutional neural networks for sentence classification. Preprint, *arXiv: 1408.5882.*

[22] Kohler, M. (2014). Optimal global rates of convergence for noiseless regression estimation problems with adaptively chosen design. *Journal of Multivariate Analysis* **13**, pp. 197-208.

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

[24] Kohler, M., and Krzyżak, A. (2021). Over-parametrized deep neural networks minimizing the empirical risk do not generalize well. *Bernoulli* **27**, pp. 2564-2597.

[25] Kohler, M., and Krzyżak, A. (2022). Analysis of the rate of convergence of an over-parametrized deep neural network estimate learned by gradient descent. Preprint, *arXiv: 2210.01443.*

[26] Kohler, M., and Krzyżak, A. (2023). On the rate of convergence of an over-parametrized deep neural network regression estimate with ReLU activation function learned by gradient descent. Preprint.

[27] Kohler, M., and Langer, S. (2021). On the rate of convergence of fully connected deep neural network regression estimates using ReLU activation functions. *Annals of Statistics* **49**, pp. 2231-2249. Preprint, *arXiv: 1908.11133.*

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

[29] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira et al. (Eds.), *Advances In Neural Information Processing Systems* Red Hook, NY: Curran. **25**, pp. 1097-1105.

[30] Kutyniok, G. (2020). Discussion of "Nonparametric regression using deep neural networks with ReLU activation function". *Annals of Statistics* **48**, pp. 1902–1905.

[31] Langer, S. (2021). Approximating smooth functions by deep neural networks with sigmoid activation function. *Journal of Multivariate Analysis* **182**.

[32] Liang, T., Rakhlin, A., and Sridharan, K. (2015). Learning with square loss: localization through offset Rademacher complexity. Preprint, *arXiv: 1502.06134.*

[33] Lin, S., and Zhang, J. (2019). Generalization bounds for convolutional neural networks. Preprint, *arXiv: 1910.01487.*

[34] Lu, J., Shen, Z., Yang, H., and Zhang, S. (2020). Deep network approximation for smooth functions. *arxiv: 2001.03040.*

[35] McCaffrey, D. F., and Gallant, A. R. (1994). Convergence rates for single hidden layer feedforward networks. *Neural Networks* **7**, pp. 147-158.

[36] Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. In *Proceedings of the National Academy of Sciences*, **115**, pp. E7665-E7671.

[37] Nguyen, P.-M., and Pham, H. T. (2020). A rigorous framework for the mean field limit of multilayer neural networks. Preprint, *arXiv: 2001.1144.*

[38] Nitanda, A., and Suzuki, T. (2021). Optimal rates for averaged stochastic gradient descent under neural tangent kernel regime. *arXiv: 2006.12297.*

[39] Rahimi, A., and Recht, B. (2008a). Random features for large-scale kernel machines. In *Advances in Neural Information Procesing Systems*, pp. 1177-1184.

[40] Rahimi, A., and Recht, B. (2008b). Uniform approximation of function with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 555-561, IEEE.

[41] Rahimi, A., and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In D. Koller, D. Schuurman, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, Curran Associates, Inc. **21**, pp. 1313-1320.

[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 Networks*, **11**, pp. 15-37.

[43] Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function (with discussion). *Annals of Statistics* **48**, pp. 1875–1897.

[44] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Huber, T., et al. (2017). Mastering the game of go without human knowledge. *Nature* **550**, pp. 354-359.

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

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

[47] Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation. *Annals of Statistics*, **25**, pp. 118-184.

[48] Suzuki, T., and Nitanda, A. (2019). Deep learning is adaptive to intrinsic diemsnionality of model smoothness in anisotropic Besov space. arXiv: 1910.12799.

[49] Wang, M., and Ma, C. (2022). Generalization error bounds for deep neural network trained by SGD. Preprint, *arXiv: 2206.03299v1.*

[50] Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., Krikum, M., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. Preprint, *arXiv: 1609.08144.*

[51] Yarotsky, D. (2018). Optimal approximation of continuous functions by very deep ReLU networks. Preprint, *arXiv: 1802.03620.*

[52] Yarotsky, D., and Zhevnerchuk, A. (2019). The phase diagram of approximation rates for deep neural networks. Preprint, *arXiv: 1906.09477.*

[53] 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.

[54] Zong, M., and Krishnamachari, B.(2022). A survey on GPT-3. *arXiv: 2212.00857*

[55] Zou, D., Cao, Y., Zhou, D., und Gu, Q. (2018). Stochastic gradient descent optimizes over-parameterized deep ReLU networks. Preprint, *arXiv: 1811.08888.*