Learning of deep convolutional network image classifiers via stochastic gradient descent and over-parametrization *

Michael Kohler [1], Adam Krzyżak[2,†] and Alisha Sänger [1]

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

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

March 25, 2024

Abstract

Image classification from independent and identically distributed random variables is considered. Image classifiers are defined which are based on a linear combination of deep convolutional networks with max-pooling layer. Here all the weights are learned by stochastic gradient descent. A general result is presented which shows that the image classifiers are able to approximate the best possible deep convolutional network. In case that the a posteriori probability satisfies a suitable hierarchical composition model it is shown that the corresponding deep convolutional neural network image classifier achieves a rate of convergence which is independent of the dimension of the images.

AMS classification: Primary 62G08; secondary 62G20.

Key words and phrases: Convolutional neural networks, image classification, stochastic gradient descent, over-parametrization, rate of convergence.

1. Introduction

1.1. Scope of the paper

In image classification the task is to learn the functional relationship between input and output, where the input consists of observed images and the output represents classes of the corresponding images that describe what kind of objects are present in the images. Since many years the most successful approaches in the area of image classification are based on deep convolutional neural networks (CNNs), see, e.g., Krizhevsky, Sutskever and Hinton (2012), LeCun, Bengio and Hinton (2015) and Rawat and Wang (2017). Recently, it has been shown that CNN image classifiers that minimize empirical risk are able to

*Running title: *Deep network classifiers*

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

achieve dimension reduction (see Kohler, Krzyżak and Walter (2022), Kohler and Langer (2020), Walter (2021) and Kohler and Walter (2023)). However, in practice, it is not possible to compute the empirical risk minimizer. Instead, a gradient descent approach based on smooth surrogate losses and over-parameterized networks having many more trainable parameters than training samples is used.

In Kohler, Krzyżak and Walter (2023) a plug-in classifier based on convolutional networks, which is learned by gradient descent, has been analyzed. The main result there was that this classifier achieves a dimension reduction in an average-pooling model, however in contrast to the results above for the estimates based on empirical risk minimization the model there does neither use a (more realistic) max-pooling model nor any kind of hierarchical structure.

In the present paper we consider the case of large datasets such as ImageNet, which make use of gradient descent prohibitively expensive. To be able to deal with such large data sets, we define the estimate by using stochastic gradient descent. In addition, we consider surrogate logistic loss and hierarchical models with max-pooling. Here we show dimensionality reduction and independence of rates from image dimensions.

1.2. Pattern recognition

We study image classifiers in the context of pattern recognition. Let $d_1, d_2 \in \mathbb{N}$ and let (X, Y), (X_1, Y_1), ..., (X_n, Y_n) be independent and identically distributed random variables with values in

$$[0,1]^{d_1 \times d_2} \times \{-1, 1\}.$$

Here we use the notation

$$[0,1]^{d_1 \times d_2} = [0,1]^{\{1,\dots,d_1\} \times \{1,\dots,d_2\}}$$

and

$$[0,1]^J = \{(a_j)_{j \in J} : a_j \in [0,1] \quad (j \in J)\}$$

for a nonempty and finite index set J, and we describe a (random) image from (random) class $Y \in \{-1, 1\}$ by a (random) matrix X with d_1 columns and d_2 rows, which contains at position (i, j) the grey scale value of the pixel of the image at the corresponding position. Our aim is to predict Y given X. More precisely, given the data set

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

the goal is to construct a classifier

$$\hat{C}_n(\cdot) = \hat{C}_n(\cdot, \mathcal{D}_n) : [0,1]^{d_1 \times d_2} \to \{-1, 1\}$$

such that the misclassification probability

$$\mathbf{P}\{\hat{C}_n(X) \neq Y | \mathcal{D}_n\}$$

is as small as possible.

Let

$$\eta(x) = \mathbf{P}\{Y = 1|X = x\} \quad (x \in [0,1]^{d_1 \times d_2}) \tag{1}$$

be the so–called a posteriori probability of class 1. Then

$$f^*(x) = \begin{cases} 1, & \text{if } \eta(x) \geq \frac{1}{2} \\ -1, & \text{elsewhere} \end{cases}$$

is the so–called Bayes classifier, i.e., it satisfies

$$\mathbf{P}\{f^*(X) \neq Y\} = \min_{f:[0,1]^{d_1 \times d_2} \to \{0,1\}} \mathbf{P}\{f(X) \neq Y\}$$

(cf., e.g., Theorem 2.1 in Devroye, Györfi and Lugosi (1996)).

In this paper we derive upper bounds on

$$\begin{aligned} &\mathbf{E}\left\{\mathbf{P}\{\hat{C}_n(X) \neq Y|\mathcal{D}_n\} - \mathbf{P}\{f^*(X) \neq Y\}\right\} \\ &= \mathbf{P}\{\hat{C}_n(X) \neq Y\} - \min_{f:[0,1]^{d_1 \times d_2} \to \{0,1\}} \mathbf{P}\{f(X) \neq Y\}. \end{aligned} \tag{2}$$

1.3. Main results

We define deep convolutional neural network estimates by minimizing the empirical logistic loss of a linear combination of networks via stochastic gradient descent. Here we use a projection step on the weights in order to ensure that we can control the over-parametrization of the estimate. We use this estimate to define an image classifier $\hat{C}_n$.

We show, that in case that the a posteriori probability $\eta(x) = \mathbf{P}\{Y = 1|X = x\}$ satisfies a (p, C)–smooth hierarchical max-pooling model of finite level l and $supp(\mathbf{P}_X) \subseteq [0,1]^{d_1 \times d_2}$, we have

$$\mathbf{P}\left\{Y \neq \hat{C}_n(X)\right\} - \mathbf{P}\left\{Y \neq f^*(X)\right\} \leq c_1 \cdot (\log n)^2 \cdot n^{-\min\{\frac{p}{4p+8}, \frac{1}{8}\}}.$$

And if, in addition,

$$\mathbf{P}\left\{X : \max\left\{\frac{\eta(X)}{1-\eta(X)}, \frac{1-\eta(X)}{\eta(X)}\right\} > n^{\frac{1}{4}}\right\} \geq 1 - \frac{1}{n^{\frac{1}{4}}} \quad (n \in \mathbb{N})$$

holds (which implies that with high probability $\eta(X) = \mathbf{P}\{Y = 1|X\}$ is either close to one or close to zero), then we show that the estimates achieve the improved rate of convergence

$$\mathbf{P}\left\{Y \neq \hat{C}_n(X)\right\} - \mathbf{P}\left\{Y \neq f^*(X)\right\} \leq c_2 \cdot (\log n)^4 \cdot n^{-\min\{\frac{p}{2p+4}, \frac{1}{4}\}}.$$

In order to prove these results we derive a general result which gives an upper bound on the expected logistic loss of an over-parametrized linear combination of deep convolutional networks learned by minimizing an empirical logistic loss via stochastic gradient descent.

1.4. Discussion of related results

Stochastic gradient descent has been proposed in Robbins and Monroe (1951) and further discussed in Nemirovsky et al. (2008), Polyak and Yuditsky (1992), Spall (2003) and Kushner and Yin (2003). It is an efficient alternative to the standard batch gradient descent (GD) which has high computational complexity owing it to using all large-scale data stored in memory in each iteration and not allowing online updates. In SGD one sample is used to randomly update the gradient in each iteration, instead of directly calculating the exact value of the gradient. SGD is an unbiased estimate of the real gradient, its cost does not depend on the number of samples and it converges with sublinear rate, but achieves the optimal rate for convex problems, cf., e.g., Nemirovsky et al. (2008). SGD algorithms have been used in many classical machine learning problems such as perceptron, k-means, SVM and lasso, see Bottou (2012). Many improvements of classical SGD have been introduced over the years. They include momentum, Nesterov Accelerated GD, Adaptive Learning Rate Method, Adaptive Moment Estimation (ADAM), Stochastic Average Gradient, Stochastic Variance Reduction Gradient and Altering Direction Method of Multipliers, see Sun et al. (2019) for a comprehensive survey. Asymptotic and finite-sample properties of estimators based on stochastic gradients were investigated by Toulis and Airaldi (2017). Statistical inference for model parameters in SGD has been discussed in Chen et al. (2020). An excellent survey of optimization methods for large-scale machine learning including SGD is provided in Bottou, Curtis and Nocedal (2018).

In recent years much attention has been devoted to properties of deep neural network estimates. There exist quite a few approximation results for neural networks (cf., e.g., Yarotsky (2018), Yarotsky and Zhevnerchute (2019), Lu et al. (2020), Langer (2021) and the literature cited therein). Generalization abilities of deep neural networks can either be analyzed 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).

Combining such results leads to a rich theory showing that owing to the network structure the least squares neural network estimates can achieve suitable dimension reduction in hierarchical composition models for the function to be estimated. For a simple model this was first shown by Kohler and Krzyżak (2017) for Hölder smooth function and later extended to arbitrary smooth functions by Bauer and Kohler (2019). For a more complex hierarchical composition model and the ReLU activation function this was shown in Schmidt-Hieber (2020) under the assumption that the networks satisfy some sparsity constraint. Kohler and Langer (2021) showed that this also possible for fully connected neural networks, i.e., without imposing a sparsity constraint on the network. Adaptation of deep neural network to especially weak smoothness assumptions was shown in Imaizumi and Fukamizu (2018), Suzuki (2018) and Suzuki and Nitanda (2019).

Less well understood is the optimization of deep neural networks. As was shown, e.g., in Zou et al. (2018), Du et al. (2019), Allen-Zhu, Li and Song (2019) and Kawaguchi and Huang (2019) the application of gradient descent to over-parameterized deep neural networks leads to a neural network which (globally) minimizes the empirical risk considered. However, as was shown in Kohler and Krzyżak (2021), the corresponding estimates do not behave well on new independent data. So the main question is why gradient descent (and its variants like stochastic gradient descent) can be used to fit a neural network to observed data in such a way that the resulting estimate achieves good results on new independent data. The challenge here is not only to analyze optimization but to consider it simultaneously with approximation and generalization.

In case of shallow neural networks (i.e., neural networks with only one hidden layer) this has been done successfully in Braun et al. (2024). Here it was possible to show that the classical dimension free rate of convergence of Barron (1994) for estimation of a regression function where its Fourier transform has a finite moment can also be achieved by shallow neural networks learned by gradient descent. The main idea here is that the gradient descent selects a subset of the neural network where random initialization of the inner weights leads to values with good approximation properties, and that it adjusts the outer weights for these neurons properly. A similar idea was also applied in Gonon (2021). Kohler and Krzyżak (2022) applied this idea in the context of over-parametrized deep neural networks where a linear combination of a huge number of deep neural networks of fixed size are computed in parallel. Here the gradient descent selects again a subset of the neural networks computed in parallel and chooses a proper linear combination of the networks. By using metric entropy bounds (cf., e.g., Birman and Solomnjak (1967) and Li, Gu and Ding (2021)) it is possible to control generalization of the over-parametrized neural networks, and as a result the rate of convergence of order close to $n^{-1/(1+d)}$ (or $n^{1/(1+d^*)}$ in case of interaction models, where it is assumed that the regression function is a sum of functions applied to only d^* of the d components of the predictor variable) can be shown for Hölder-smooth regression functions with Hölder exponent $p \in [1/2, 1]$. Universal consistency of such estimates for bounded X was shown in Drews and Kohler (2022).

In all those results adjusting the inner weights with gradient descent is not important. In fact, Gonon (2021) does not do this at all, while Braun et al. (2024) and Kohler and Krzyżak (2022) use the fact that the relevant inner weights do not move too far away from their starting values during gradient descent. Similar ideas have also been applied in Andoni et al. (2014) and Daniely (2017). This whole approach is related to random feature networks (cf., e.g., Huang, Chen and Siew (2006) and Rahimi and Recht (2008a, 2008b, 2009)), where the inner weights are chosen randomly and only the outer weights are learned during gradient descent. Yehudai and Shamir (2022) present a lower bound which implies that either the number of neurons or the absolute value of the coefficients must grow exponential in the dimension in order to learn a single ReLU neuron with random feature networks. But since Braun et al. (2024) was able to prove a useful rate of convergence result for networks similar to random feature networks, the practical relevance of this lower bound is not clear.

The estimates in Kohler and Krzyżak (2022) use a L_2 regularization on the outer

weights during gradient descent. As was shown in Drews and Kohler (2023), it is possible to achieve similar results without L_2 regularization.

Often gradient descent in neural networks is studied in the neural tangent kernel setting proposed by Jacot, Gabriel and Hongler (2020), where 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. Suzuki and Nitanda (2019) 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 the classical smoothness conditions of our paper. 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). A survey of various results on over-parametrized deep neural network estimates learned by gradient descent can be found in Bartlett, Montanari and Rakhlin (2021).

In recent years deep transformer networks became very popular in research and applications. They have been introduced by Vaswani et al. (2017) and their approximation and generalization properties have been investigated by Gurevych et al. (2022). The rates of convergence of over-parametrized transformer classifiers learned by gradient descent have been studied by Kohler and Krzyżak (2023).

1.5. Notation

The sets of natural numbers, real numbers and nonnegative real numbers are denoted by $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}_+$, respectively. We define furthermore $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. For $z \in \mathbb{R}$, we denote the smallest integer greater than or equal to z by $\lceil z \rceil$, the largest integer less than or equal to z by $\lfloor z \rfloor$, and we set $z_+ = \max\{z, 0\}$. 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 $A \subseteq \mathbb{R}^d$.

For $\mathbf{j} = (j^{(1)}, \ldots, j^{(d)}) \in \mathbb{N}_0^d$ we write

$$\|\mathbf{j}\|_1 = j^{(1)} + \cdots + j^{(d)}$$

and for $f : \mathbb{R}^d \to \mathbb{R}$ we set

$$\partial^{\mathbf{j}} f = \frac{\partial^{\|\mathbf{j}\|_1} f}{(\partial x^{(1)})^{j^{(1)}} \dots (\partial x^{(d)})^{j^{(d)}}}.$$

Let $\mathcal{F}$ be a set of functions $f : \mathbb{R}^d \to \mathbb{R}$, let $x_1, \dots, x_n \in \mathbb{R}^d$, set $x_1^n = (x_1, \dots, x_n)$ and let $p \geq 1$. A finite collection $f_1, \dots, f_N : \mathbb{R}^d \to \mathbb{R}$ is called an L_p ε–packing in $\mathcal{F}$ on x_1^n if $f_1, \dots, f_N \in \mathcal{F}$ and

$$\min_{1 \leq i < j \leq N} \left(\frac{1}{n} \sum_{k=1}^{n} |f_i(x_k) - f_j(x_k)|^p \right)^{1/p} \geq \varepsilon$$

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

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 $(T_\beta f)(x) = T_\beta (f(x))$. And $sign(z)$ is the sign of $z \in \bar{\mathbb{R}}$.

1.6. Outline

A general result, namely a bound on the logistic risk of an over-parametrized deep convolutional network fitted to data via stochastic gradient descent, is presented in Section 2. The over-parametrized deep convolutional neural network classifiers considered in this paper are introduced in Section 3 and a bound on their misclassification probability is also presented in this section. Section 4 contains the proofs.

2. A general result

Let $\mathbf{\Theta}$ be a closed and convex set of parameter values (weights) for a deep convolutional network of a given topology. In the sequel we assume that our aim is to learn the parameter $\vartheta \in \mathbf{\Theta}$ (vector of weights) for a deep convolutional network

$$f_\vartheta : [0,1]^{d_1 \times d_2} \to \mathbb{R}$$

from the data $\mathcal{D}_n$ such that

$$sign(f_\vartheta(x))$$

is a good classifier. We do this by considering linear combinations

$$f_{(\mathbf{w},\vartheta)}(x) = \sum_{k=1}^{K_n} w_k \cdot T_{\beta_n}(f_{\vartheta_k}(x)) \tag{3}$$

of truncated versions of deep convolutional networks $f_{\vartheta_k}(x)$ $(k = 1, \dots, K_n)$, where $\mathbf{w} = (w_k)_{k=1,\dots,K_n}$ satisfies

$$w_k \geq 0 \quad (k = 1, \dots, K_n), \quad \sum_{k=1}^{K_n} w_k \leq 1 \quad \text{and} \quad \sum_{k=1}^{K_n} w_k^2 \leq \alpha_n \tag{4}$$

for some $\alpha_n \in [0,1]$, where $\vartheta = (\vartheta_1, \ldots, \vartheta_{K_n}) \in \mathbf{\Theta}^{K_n}$ and where $\beta_n = c_3 \cdot \log n$. Observe that by choosing $\alpha_n = \frac{1}{N_n}$, $w_j = \frac{1}{N_n}$ $(j = 1, \ldots, N_n)$, $\vartheta_j = \vartheta$ $(j = 1, \ldots, N_n)$ and $w_k = 0$ for $k > N_n$ we get

$$f_{(\mathbf{w},\vartheta)}(x) = T_{\beta_n}(f_\vartheta(x))$$

and in this way we can construct an estimate which satisfies

$$sign(f_{(\mathbf{w},\vartheta)}(x)) = sign(f_\vartheta(x))$$

for any $\vartheta \in \mathbf{\Theta}$. And by choosing K_n very large our estimate will be over-parametrized in the sense that the number of parameters of the estimate is much larger than the sample size.

Let

$$\varphi(z) = \log(1 + \exp(-z))$$

be the logistic loss (or cross entropy loss). Our aim in choosing $(\mathbf{w}, \vartheta)$ is the minimization of the logistic risk

$$F((\mathbf{w}, \vartheta)) = \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w},\vartheta)}(X))\right\}.$$

In order to achieve this, we start with a random initialization of $(\mathbf{w}, \vartheta)$: We choose

$$\vartheta_1^{(0)}, \ldots, \vartheta_{K_n}^{(0)} \tag{5}$$

uniformly from some closed and convex set $\mathbf{\Theta}^0 \subseteq \mathbf{\Theta}$ such that the random variables in (5) are independent and also independent from (X, Y), (X_1, Y_1), $\ldots$, (X_n, Y_n), and we set

$$w_k^{(0)} = 0 \quad (k = 1, \ldots, K_n).$$

Then we perform $t_n \in \mathbb{N}$ stochastic gradient descent steps starting with

$$\vartheta^{(0)} = (\vartheta_1^{(0)}, \ldots, \vartheta_{K_n}^{(0)}) \quad \text{and} \quad \mathbf{w}^{(0)} = (w_1^{(0)}, \ldots, w_{K_n}^{(0)}).$$

Here we assume that t_n/n is a natural number, and for $s \in \{1, \ldots, t_n/n\}$ we let

$$j_{(s-1)\cdot n}, \cdots, j_{s \cdot n - 1}$$

be an arbitrary permutation of $1, \ldots, n$, we choose a stepsize $\lambda_n > 0$ and we set

$$\begin{aligned} \mathbf{w}^{(t+1)} &= Proj_A\left(\mathbf{w}^{(t)} - \lambda_n \cdot \nabla_{\mathbf{w}} \varphi\left(Y_{j_t} \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X_{j_t})\right)\right), \\ \vartheta^{(t+1)} &= Proj_B\left(\vartheta^{(t)} - \lambda_n \cdot \nabla_{\vartheta} \varphi\left(Y_{j_t} \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X_{j_t})\right)\right) \end{aligned}$$

for $t = 0, \ldots, t_n - 1$. Here A is the set of all $\mathbf{w}$ which satisfy (4), and

$$B = \left\{\vartheta \in \mathbf{\Theta}^{K_n} \,:\, \|\vartheta - \vartheta^{(0)}\| \leq 1\right\},$$

and $Proj_A$ and $Proj_B$ is the L_2 projection on A and B. Our estimate is then defined by

$$f_n(x) = f_{(\hat{\mathbf{w}},\vartheta^{(t_n)})}(x) \tag{6}$$

where

$$\hat{\mathbf{w}} = \frac{1}{t_n} \cdot \sum_{t=0}^{t_n-1} \mathbf{w}^{(t)}. \tag{7}$$

Our main result in this general setting is the following bound on the logistic risk of the above estimate.

Theorem 1 *Let (X,Y), (X_1,Y_1), ..., (X_n,Y_n) be independent and identically distributed random variables with values in $[0,1]^{d_1\times d_2} \times \{-1,1\}$. Let $N_n, I_n, t_n \in \mathbb{N}$ and let $C_n, D_n \geq 0$. Set $\beta_n = c_3 \cdot \log n$,*

$$\alpha_n = \frac{1}{N_n}, \quad \lambda_n = \frac{1}{t_n}, \quad K_n = N_n \cdot I_n$$

and define the estimate f_n as above.

Let $\mathbf{\Theta}^* \subset \mathbf{\Theta}^0$ and set

$$\bar{\mathbf{\Theta}} = \left\{ \vartheta \in \mathbf{\Theta} \ : \ \inf_{\tilde{\vartheta}\in\mathbf{\Theta}^0} \|\vartheta - \tilde{\vartheta}\| \leq 1 \right\}.$$

Assume

$$\|f_\vartheta - f_{\bar{\vartheta}}\|_{\infty,supp(X)} \leq C_n \cdot \|\vartheta - \bar{\vartheta}\| \quad \text{for all } \vartheta, \bar{\vartheta} \in \bar{\mathbf{\Theta}}, \tag{8}$$

$$\epsilon_n = \mathbf{P}\left\{ \vartheta_1^{(0)} \in \mathbf{\Theta}^* \right\} > 0, \tag{9}$$

$$N_n \cdot (1-\epsilon_n)^{I_n} \leq \frac{1}{n} \tag{10}$$

and

$$\|\nabla_{\mathbf{w}} \varphi(y \cdot f_{(\mathbf{w},\vartheta)})\| \leq D_n \tag{11}$$

for all $x \in [0,1]^{d_1\times d_2}$, $y \in \{-1,1\}$, $\mathbf{w} \in A$, $\vartheta \in \bar{\mathbf{\Theta}}$, $t \in \{0,\ldots,t_n-1\}$.

Then we have

$$\begin{aligned}
&\mathbf{E}\left\{\varphi(Y\cdot f_n(X))\right\} - \min_{f:[0,1]^{d_1\times d_2}\to\bar{\mathbb{R}}} \mathbf{E}\left\{\varphi(Y\cdot f(X))\right\} \\
&\leq c_4 \cdot \left(\frac{\log n}{n} + \beta_n \cdot \sup_{x_1,\ldots,x_n\in[0,1]^{d_1\times d_2}} \mathbf{E}\left\{ \left| \sup_{\vartheta\in\bar{\mathbf{\Theta}}} \frac{1}{n} \sum_{i=1}^n \epsilon_i \cdot T_{\beta_n} f_\vartheta(x_i) \right| \right\} + \frac{C_n+1}{\sqrt{N_n}} + \frac{D_n^2}{t_n} \right. \\
&\quad + \frac{n\cdot\beta_n \cdot \left(K_n + C_n \cdot \sup_{\mathbf{w}\in A, \vartheta\in\bar{\Theta}^{K_n}, y\in\{-1,1\}, x\in[0,1]^{d_1\times d_2}} \|\nabla_\vartheta \varphi(y\cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty \right)}{t_n} \\
&\quad \left. + \sup_{\vartheta\in\mathbf{\Theta}^*} \mathbf{E}\left\{\varphi(Y\cdot T_{\beta_n} f_\vartheta(X))\right\} - \min_{f:[0,1]^{d_1\times d_2}\to\bar{\mathbb{R}}} \mathbf{E}\left\{\varphi(Y\cdot f(X))\right\} \right),
\end{aligned}$$

where $\epsilon_1, \ldots, \epsilon_n$ are independent and uniformly distributed on $\{-1,1\}$ (so-called Rademacher random variables).

Remark 1. Our result above extends Theorem 2 in Kohler and Krzyżak (2023) from gradient descent to the case of stochastic gradient descent. To be able to do this we need in the definition of the estimate an additional L_2 penalty on the weights in the linear combination of the networks (depending on α_n). Furthermore, assumption (8) is substantially stronger than the corresponding assumption in Theorem 2 in Kohler and Krzyżak (2023), because there it is only required that (8) holds for networks which have good approximation properties (which is because of the maximal attention used in Transformer networks crucial for Transformer networks).

3. Image classification using deep convolutional neural networks

3.1. Convolutional neural network classifiers

We aim to learn feature representations of the inputs by means of L (hidden) convolutional layers. Each of these $r \in \{1, \ldots, L\}$ feature maps consists of k_r channels. The input image is considered as layer 0 with only one channel, i.e. $k_0 = 1$.
A convolution in layer r is performed by using a window of values of the previous layer $r-1$ of size M_r. The window has to fit within the dimensions of the input image, i.e. $M_r \leq \min\{d_1, d_2\}$. It relies on so-called filters, i.e. a weight matrix that determines how a neuron is computed from a weighted sum of neighboring neurons from the previous layer. The weight matrix is defined by

$$\mathbf{w} = \left(w_{i,j,s_1,s_2}^{(r)}\right)_{1 \leq i,j \leq M_r, s_1 \in \{1,\ldots,k_{r-1}\}, s_2 \in \{1,\ldots,k_r\}, r \in \{1,\ldots,L\}}.$$

Furthermore we need some weights

$$\mathbf{w}_{bias} = (w_{s_2}^{(r)})_{s_2 \in \{1,\ldots,k_r\}, r \in \{1,\ldots,L\}}$$

for the bias in each channel and output weights

$$\mathbf{w}_{out} = (w_s)_{s \in \{1,\ldots,k_L\}},$$

which are required for the max-pooling layer defined below.
In the following the ReLU function $\sigma(x) = \max\{x, 0\}$ is chosen as activation function. The value of a feature map in the s_2-th channel of layer r at the position (i, j) is recursively defined by:

$$o_{(i,j),s_2}^{(r)} = \sigma\left(\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\ldots,M_r\} \\ (i+t_1-1, j+t_2-1) \in D}} w_{t_1,t_2,s_1,s_2}^{(r)} o_{(i+t_1-1,j+t_2-1),s_1}^{(r-1)} + w_{s_2}^{(r)}\right), \tag{12}$$

where $(i, j) \in D = \{1, \ldots, d_1\} \times \{1, \ldots, d_2\}$, $s_2 \in \{1, \ldots, k_r\}$ and $r \in \{1, \ldots, L\}$.
The anchor case $r = 0$ of this recursion reflects the values of the input image

$$o_{(i,j),1}^{(0)} = x_{i,j} \quad \text{for } i \in \{1, \ldots, d_1\} \text{ and } j \in \{1, \ldots, d_2\}.$$

In definition (12) above we see that weights generating the feature map $o^{(r)}_{(:,:),s2}$ are shared. Weight sharing is used to reduce model complexity, thereby increasing the network's computational efficiency. In the last step a max-pooling layer is applied to the values in the k_L channels of the last convolutional layer L, such that the output of the network is given by a real-valued function on $[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ of the form

$$f_{\mathbf{w},\mathbf{w}_{bias},\mathbf{w}_{out}}(x) = \max\Bigg\{ \sum_{s_2=1}^{k_L} w_{s_2} \cdot o^{(L)}_{(i,j),s_2} : i \in \{1,\dots,d_1 - M_L + 1\} \\ , j \in \{1,\dots,d_2 - M_L + 1\} \Bigg\}.$$

Our class of convolutional neural networks with parameters L, $\mathbf{k} = (k_1,\dots,k_L)$ and $\mathbf{M} = (M_1,\dots,M_L)$ is defined by $\mathcal{F}^{CNN}_{L,\mathbf{k},\mathbf{M}}$. As in Kohler, Krzyżak and Walter (2022), the definition of the summation index over $t_1, t_2 \in \{1,\dots,M_r\}$, such that $1 \leq i + t_1 - 1 \leq d_1$ and $1 \leq j + t_2 - 1 \leq d_2$, corresponds to zero padding to the left and to the bottom of the image. Thus, the size of a channel is the same as in the previous layer (see Kohler, Krzyżak and Walter (2022) for a further illustration). Our final estimate is a composition of a convolutional neural network out of the class $\mathcal{F}^{CNN}_{L,\mathbf{k},\mathbf{M}}$ and a shallow neural network, which is defined as follows: The output of this network is produced by a function $g : \mathbb{R} \to \mathbb{R}$ of the form

$$g(x) = \sum_{i=1}^{L_n^{(2)}} w_i^{(1)} \sigma\left(w_{i,1}^{(0)} \cdot x + w_{i,0}^{(0)}\right) + w_0^{(1)}, \tag{13}$$

where $w_0^{(1)}, w_1^{(1)}, w_{1,0}^{(1)}, w_{1,1}^{(1)}, \dots, w^{(1)}_{L_n^{(2)}}, w^{(0)}_{L_n^{(2)},0}, w^{(0)}_{L_n^{(2)},1} \in \mathbb{R}$ denote the weights of this network and $\sigma(z) = \max\{z, 0\}$ is again the ReLU activation function. We define the function class of all real-valued functions on $\mathbb{R}$ of the form (13) with parameter $L_n^{(2)}$ by $\mathcal{F}^{FNN}_{L_n^{(2)}}$.

Our final function class $\mathcal{F}_n$ is then of the form

$$\mathcal{F}_n = \left\{ g \circ f : g \in \mathcal{F}^{FNN}_{L_n^{(2)}}, f \in \mathcal{F}^{CNN}_{L_n^{(1)},\mathbf{k},\mathbf{M}} \right\}, \tag{14}$$

which depends on the parameters

$$\mathbf{L} = (L_n^{(1)}, L_n^{(2)}),\ \mathbf{k} = \left(k_1,\dots,k_{L_n^{(1)}}\right),\ \mathbf{M} = (M_1,\dots,M_{L_n^{(1)}}).$$

3.2. Definition of the estimate

Let Θ be the set of all weights of the function class

$$\mathcal{F}_n = \{f_\vartheta\ :\ \theta \in \Theta\}$$

introduced in the previous subsection. In the sequel we fit a linear combination

$$f_{(\mathbf{w},\vartheta)}(x) = \sum_{k=1}^{K_n} w_k \cdot T_{\beta_n} f_{\vartheta_k}$$

to the data where $\mathbf{w}$ satisfies the assumption (4) and $\vartheta = (\theta_1, \dots, \theta_{K_n}) \in \Theta^{K_n}$.

Depending on some $B_n > 0$, which will be defined in Theorem 2 below, we choose

$$\vartheta_1^{(0)}, \dots, \vartheta_{K_n}^{(0)} \tag{15}$$

uniformly from

$$\mathbf{\Theta}^0 = \{\vartheta \in \mathbf{\Theta} \quad : \quad \|\vartheta\|_\infty \leq B_n\}$$

such that the random variables in (15) are independent and also independent from (X, Y), (X_1, Y_1), ..., (X_n, Y_n) and we set

$$w_k^{(0)} = 0 \quad (k = 1, \dots, K_n).$$

Then we perform $t_n \in \mathbb{N}$ stochastic gradient descent steps starting with

$$\vartheta^{(0)} = (\vartheta_1^{(0)}, \dots, \vartheta_{K_n}^{(0)}) \quad \text{and} \quad \mathbf{w}^{(0)} = (w_1^{(0)}, \dots, w_{K_n}^{(0)}).$$

As in the previous section we assume that t_n/n is a natural number and for $s \in \{1, \dots, t_n/n\}$ we let

$$j_{(s-1)\cdot n}, \cdots, j_{s\cdot n-1}$$

be an arbitrary permutation of $1, \dots, n$. We choose a stepsize $\lambda_n > 0$ and set

$$\begin{aligned} \mathbf{w}^{(t+1)} &= Proj_A \left(\mathbf{w}^{(t)} - \lambda_n \cdot \nabla_{\mathbf{w}} \varphi \left(Y_{j_t} \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t)})}(X_{j_t}) \right) \right), \\ \vartheta^{(t+1)} &= Proj_B \left(\vartheta^{(t)} - \lambda_n \cdot \nabla_{\vartheta} \varphi \left(Y_{j_t} \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t)})}(X_{j_t}) \right) \right) \end{aligned}$$

for $t = 0, \dots, t_n - 1$. Here A is the set of all $\mathbf{w}$ which satisfy (4), and

$$B = \left\{ \vartheta \in (\mathbf{\Theta}^{(0)})^{K_n} \, : \, \|\vartheta - \vartheta^{(0)}\| \leq 1 \right\},$$

and $Proj_A$ and $Proj_B$ is the L_2 projection on A and B. In order to compute the gradient with respect to the inner weights we use the following convention: We set

$$\sigma'(z) = \frac{\partial}{\partial z} \max\{z, 0\} = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{else} \end{cases}$$

$$\frac{\partial}{\partial z} T_{\beta_n} z = \frac{\partial}{\partial z} \max\{-\beta_n, \min\{\beta_n, z\}\} = \begin{cases} 1, & \text{if } |z| \leq \beta_n \\ 0, & \text{else} \end{cases}$$

and

$$\frac{\partial}{\partial \vartheta_j} \max\left\{f_{1,\vartheta_1}(x), \dots, f_{L,\vartheta_L}(x)\right\} = \frac{\partial}{\partial \vartheta_j} f_{l,\vartheta_l}(x)$$

where $l \in \{1, \dots, L\}$ satisfies

$$\max\left\{f_{1,\vartheta_1}(x), \dots, f_{L,\vartheta_L}(x)\right\} = f_{l,\vartheta_l}(x)$$

and

$$l = 1 \quad \text{or} \quad \max\left\{f_{1,\vartheta_1}(x), \ldots, f_{l-1,\vartheta_{l-1}}(x)\right\} < f_{l,\vartheta_l}(x).$$

Our classifier $\hat{C}_n(x)$ is then defined by

$$\hat{C}_n(x) = sign(f_n(x)), \tag{16}$$

where

$$f_n(x) = f_{(\hat{\mathbf{w}},\vartheta^{(t_n)})}(x) \quad \text{and} \quad \hat{\mathbf{w}} = \frac{1}{t_n} \cdot \sum_{t=0}^{t_n-1} \mathbf{w}^{(t)}. \tag{17}$$

3.3. Main result

It is well known that one needs smoothness assumptions on the a posteriori probability in order to derive non-trivial rate of convergence results for the difference between the misclassification risk of any estimate and the optimal misclassification risk (cf., e.g., Cover (1968) and Section 3 in Devroye and Wagner (1982)). For this we will use our next definition.

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

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

for all $x, \mathbf{z} \in \mathbb{R}^d$.

Furthermore we will use a model from Kohler, Krzyżak and Walter (2022) to be able to derive rates of convergence which do not depend on the dimension $d_1 \cdot d_2$ of the images. In order to be able to introduce this model, we need the following notation: For $M \subseteq \mathbb{R}^d$ and $x \in \mathbb{R}^d$ we define

$$x + M = \{x + \mathbf{z} : \mathbf{z} \in M\}.$$

For $I \subseteq \{1, \ldots, d_1\} \times \{1, \ldots, d_2\}$ and $x = (x_i)_{i \in \{1,\ldots,d_1\} \times \{1,\ldots,d_2\}} \in [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}}$ we set

$$x_I = (x_i)_{i \in I}.$$

The basic idea behind the next definition is that the a posteriori probability is a maximum of probabilities that special objects occur in subparts of the image, and that the decision about the latter events are hierarchically decided.

Definition 2 *Let $d_1, d_2 \in \mathbb{N}$ with $d_1, d_2 > 1$ and $m : [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}} \to \mathbb{R}$.*
a) *We say that m satisfies a* **max-pooling model with index set**

$$I \subseteq \{0, \ldots, d_1 - 1\} \times \{0, \ldots, d_2 - 1\},$$

if there exist a function $f : [0,1]^{(1,1)+I} \to \mathbb{R}$ *such that*

$$m(x) = \max_{(i,j)\in\mathbb{Z}^2 \,:\, (i,j)+I \subseteq \{1,\dots,d_1\}\times\{1,\dots,d_2\}} f\left(x_{(i,j)+I}\right) \quad (x \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}).$$

b) *Let* $I = \{0,\dots,2^l-1\} \times \{0,\dots,2^l-1\}$ *for some* $l \in \mathbb{N}$. *We say that*

$$f : [0,1]^{\{1,\dots,2^l\}\times\{1,\dots,2^l\}} \to \mathbb{R}$$

satisfies a **hierarchical model of level** l, *if there exist functions*

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1,\dots,l, s = 1,\dots,4^{l-k})$$

such that we have

$$f = f_{l,1}$$

for some $f_{k,s} : [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}} \to \mathbb{R}$ *recursively defined by*

$$\begin{aligned} f_{k,s}(x) \;=\; & g_{k,s}\big(f_{k-1,4\cdot(s-1)+1}(x_{\{1,\dots,2^{k-1}\}\times\{1,\dots,2^{k-1}\}}), \\ & \quad f_{k-1,4\cdot(s-1)+2}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{1,\dots,2^{k-1}\}}), \\ & \quad f_{k-1,4\cdot(s-1)+3}(x_{\{1,\dots,2^{k-1}\}\times\{2^{k-1}+1,\dots,2^k\}}), \\ & \quad f_{k-1,4\cdot s}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{2^{k-1}+1,\dots,2^k\}})\big) \\ & \qquad \Big(x \in [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}}\Big) \end{aligned}$$

for $k = 2,\dots,l, s = 1,\dots,4^{l-k}$, *and*

$$f_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2}) = g_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2}) \quad (x_{1,1},x_{1,2},x_{2,1},x_{2,2} \in [0,1])$$

for $s = 1,\dots,4^{l-1}$.

c) *We say that* $m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ *satisfies a* **hierarchical max-pooling model of level** l *(where* $2^l \le \min\{d_1,d_2\}$*), if* m *satisfies a max-pooling model with index set*

$$I = \Big\{0,\dots,2^l-1\Big\} \times \Big\{0,\dots,2^l-1\Big\}$$

and the function $f : [0,1]^{(1,1)+I} \to \mathbb{R}$ *in the definition of this max-pooling model satisfies a hierarchical model with level* l.

d) *We say that the hierarchical max-pooling model* $m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ *of level* l *is* (p,C)*–smooth if all functions* $g_{k,s}$ *in the definition of the function* m *are* (p,C)*–smooth for some* $C > 0$.

Our main result is the following theorem, in which bounds on the difference between the misclassification probability of our classifier and the optimal misclassification probability are derived.

Theorem 2 *Let $p \geq 1$ and $C > 0$ be arbitrary. Assume that the a posteriori probability $\eta(x) = \mathbf{P}\{Y = 1|X = x\}$ satisfies a (p, C)–smooth hierarchical max-pooling model of finite level l and assume that $supp(\mathbf{P}_X) \subseteq [0,1]^{d_1 \times d_2}$ holds. Set $\beta_n = c_3 \cdot \log n$,*

$$L_n^{(1)} = \frac{4^l - 1}{3} \cdot \lceil c_5 \cdot n^{2/(2p+4)} \rceil + l \quad \textit{and} \quad L_n^{(2)} = \lceil c_6 \cdot n^{1/4} \rceil,$$

$$M_s = 2^{\pi(s)} \quad (s = 1, \ldots, L_n^{(1)}),$$

where the function $\pi : \{1, \ldots, L_n^{(1)}\} \to \{1, \ldots, l\}$ is defined by

$$\pi(s) = \sum_{i=1}^{l} \mathbb{1}_{\{s \geq i + \sum_{r=l-i+1}^{l-1} 4^r \cdot \lceil c_5 \cdot n^{2/(2p+4)} \rceil\}},$$

choose $\mathbf{k} = (c_7, \ldots, c_7) \in \mathbb{N}^{L_n^{(1)}}$ and set

$$B_n = e^{\sqrt{n}},$$

assume that $K_n \in \mathbb{N}$ satisfies

$$\frac{K_n}{e^{2 \cdot n^{1.5}}} \to \infty \quad (n \to \infty)$$

and set

$$\alpha_n = \frac{1}{n^2 \cdot e^{2 \cdot n}} \quad \textit{and} \quad t_n = \left\lceil n^2 \cdot K_n \right\rceil.$$

Define the classifier $\hat{C}_n$ as in Section 3.2. Assume that the constants c_3, c_5, c_6, c_7 are sufficiently large.

a) *There exists a constant $c_8 > 0$ such that we have for n sufficiently large*

$$\mathbf{P}\left\{Y \neq \hat{C}_n(X)\right\} - \mathbf{P}\left\{Y \neq f^*(X)\right\} \leq c_8 \cdot (\log n)^2 \cdot n^{-\min\{\frac{p}{4p+8}, \frac{1}{8}\}}.$$

b) *If, in addition,*

$$\mathbf{P}\left\{X : \max\left\{\frac{\eta(X)}{1-\eta(X)}, \frac{1-\eta(X)}{\eta(X)}\right\} > n^{\frac{1}{4}}\right\} \geq 1 - \frac{1}{n^{\frac{1}{4}}} \quad (n \in \mathbb{N}) \tag{18}$$

holds, then there exists a constant $c_9 > 0$ such that we have for n sufficiently large

$$\mathbf{P}\left\{Y \neq \hat{C}_n(X)\right\} - \mathbf{P}\left\{Y \neq f^*(X)\right\} \leq c_9 \cdot (\log n)^4 \cdot n^{-\min\{\frac{p}{2p+4}, \frac{1}{4}\}}.$$

Remark 2. The rates of convergence above do not depend on the dimension $d_1 \cdot d_2$ of the image, hence in case that the a posteriori distribution satisfies a hierarchical composition model, our estimate is able to circumvent the curse of dimensionality.

4. Proofs

4.1. Proof of Theorem 1

In the proof of Theorem 1 we will need the following auxiliary result.

Lemma 1 *Let $l_1, l_2, t_n \in \mathbb{N}$, let $D_n \geq 0$, let $A \subset \mathbb{R}^{l_1}$ be closed and convex, let $B \subseteq \mathbb{R}^{l_2}$ and let $F_t, F : \mathbb{R}^{l_1} \times \mathbb{R}^{l_2} \to \mathbb{R}_+$ $(t = 0, \ldots, t_n - 1)$ be functions such that for all $t \in \{0, \ldots, t_n - 1\}$*

$$u \mapsto F(u, v) \quad \textit{is differentiable and convex for all } v \in \mathbb{R}^{l_2},$$

$$u \mapsto F_t(u, v) \quad \textit{is differentiable for all } v \in \mathbb{R}^{l_2},$$

and

$$\|(\nabla_u F_t)(u, v)\| \leq D_n \tag{19}$$

for all $(u, v) \in A \times B$. Choose $(u_0, v_0) \in A \times B$, let $v_1, \ldots, v_{t_n} \in B$ and set

$$u_{t+1} = Proj_A\left(u_t - \lambda \cdot (\nabla_u F_t)\left(u_t, v_t\right)\right) \quad (t = 0, \ldots, t_n - 1),$$

where

$$\lambda = \frac{1}{t_n}.$$

Let $u^* \in A$. Then it holds:

$$\begin{aligned}\frac{1}{t_n}\sum_{t=0}^{t_n-1} F(u_t, v_t) \quad \leq \quad & F(u^*, v_0) + \frac{1}{t_n}\sum_{t=1}^{t_n-1} |F(u^*, v_t) - F(u^*, v_0)| + \frac{\|u^* - u_0\|^2}{2} + \frac{D_n^2}{2 \cdot t_n} \\ & + \frac{1}{t_n}\sum_{t=0}^{t_n-1} < (\nabla_u F)\left(u_t, v_t\right) - (\nabla_u F_t)\left(u_t, v_t\right), u_t - u^* > .\end{aligned}$$

Proof. By convexity of $u \mapsto F(u, v_t)$ and because of $u^* \in A$ we have

$$\begin{aligned}
& F(u_t, v_t) - F(u^*, v_t) \\
\leq & < (\nabla_u F)(u_t, v_t), u_t - u^* > \\
= & \frac{1}{2 \cdot \lambda} \cdot 2 \cdot < \lambda \cdot (\nabla_u F_t)(u_t, v_t), u_t - u^* > + < (\nabla_u F)\left(u_t, v_t\right) - (\nabla_u F_t)\left(u_t, v_t\right), u_t - u^* > \\
= & \frac{1}{2 \cdot \lambda} \cdot \left(-\|u_t - u^* - \lambda \cdot (\nabla_u F_t)(u_t, v_t)\|^2 + \|u_t - u^*\|^2 + \|\lambda \cdot (\nabla_u F_t)(u_t, v_t)\|^2\right) \\
& + < (\nabla_u F)\left(u_t, v_t\right) - (\nabla_u F_t)\left(u_t, v_t\right), u_t - u^* > \\
\leq & \frac{1}{2 \cdot \lambda} \cdot \left(-\|Proj_A(u_t - \lambda \cdot (\nabla_u F_t)(u_t, v_t)) - u^*\|^2 + \|u_t - u^*\|^2 + \lambda^2 \cdot \|(\nabla_u F_t)(u_t, v_t)\|^2\right) \\
& + < (\nabla_u F)\left(u_t, v_t\right) - (\nabla_u F_t)\left(u_t, v_t\right), u_t - u^* > \\
= & \frac{1}{2 \cdot \lambda} \cdot \left(\|u_t - u^*\|^2 - \|u_{t+1} - u^*\|^2 + \lambda^2 \cdot \|(\nabla_u F_t)(u_t, v_t)\|^2\right) \\
& + < (\nabla_u F)\left(u_t, v_t\right) - (\nabla_u F_t)\left(u_t, v_t\right), u_t - u^* > .
\end{aligned}$$

This implies

$$
\begin{aligned}
&\frac{1}{t_n}\sum_{t=0}^{t_n-1} F(u_t,v_t) - \frac{1}{t_n}\sum_{t=0}^{t_n-1} F(u^*,v_t) \\
&= \frac{1}{t_n}\sum_{t=0}^{t_n-1} \left(F(u_t,v_t) - F(u^*,v_t)\right) \\
&\leq \frac{1}{t_n}\sum_{t=0}^{t_n-1} \frac{1}{2\cdot\lambda}\cdot\left(\|u_t-u^*\|^2 - \|u_{t+1}-u^*\|^2\right) + \frac{1}{t_n}\sum_{t=0}^{t_n-1}\frac{\lambda}{2}\cdot\|(\nabla_u F)(u_t,v_t)\|^2 \\
&\quad + \frac{1}{t_n}\sum_{t=0}^{t_n-1} < \left(\nabla_u F\right)(u_t,v_t) - \left(\nabla_u F_t\right)(u_t,v_t), u_t - u^* > \\
&= \frac{1}{2}\cdot\sum_{t=0}^{t_n-1}\left(\|u_t-u^*\|^2 - \|u_{t+1}-u^*\|^2\right) + \frac{1}{2\cdot t_n^2}\sum_{t=0}^{t_n-1}\|(\nabla_u F)(u_t,v_t)\|^2 \\
&\quad + \frac{1}{t_n}\sum_{t=0}^{t_n-1} < \left(\nabla_u F\right)(u_t,v_t) - \left(\nabla_u F_t\right)(u_t,v_t), u_t - u^* > \\
&\leq \frac{\|u_0-u^*\|^2}{2} + \frac{1}{2\cdot t_n^2}\sum_{t=0}^{t_n-1}\|(\nabla_u F)(u_t,v_t)\|^2 \\
&\quad + \frac{1}{t_n}\sum_{t=0}^{t_n-1} < \left(\nabla_u F\right)(u_t,v_t) - \left(\nabla_u F_t\right)(u_t,v_t), u_t - u^* > .
\end{aligned}
$$

Using the above result and (19) we get

$$
\begin{aligned}
&\frac{1}{t_n}\sum_{t=0}^{t_n-1} F(u_t,v_t) \\
&\leq \frac{1}{t_n}\sum_{t=0}^{t_n-1} F(u^*,v_t) + \frac{\|u^*-u_0\|^2}{2} + \frac{1}{2\cdot t_n^2}\sum_{t=0}^{t_n-1}\|(\nabla_u F)(u_t,v_t)\|^2 \\
&\quad + \frac{1}{t_n}\sum_{t=0}^{t_n-1} < \left(\nabla_u F\right)(u_t,v_t) - \left(\nabla_u F_t\right)(u_t,v_t), u_t - u^* > \\
&\leq F(u^*,v_0) + \frac{1}{t_n}\sum_{t=0}^{t_n-1}|F(u^*,v_t) - F(u^*,v_0)| + \frac{\|u^*-u_0\|^2}{2} + \frac{D_n^2}{2\cdot t_n} \\
&\quad + \frac{1}{t_n}\sum_{t=0}^{t_n-1} < \left(\nabla_u F\right)(u_t,v_t) - \left(\nabla_u F_t\right)(u_t,v_t), u_t - u^* > .
\end{aligned}
$$

□

Proof of Theorem 1. Let E_n be the event that there exist pairwise distinct $j_1,\ldots,j_{N_n} \in \{1,\ldots,K_n\}$ such that

$$\vartheta_{j_i}^{(0)} \in \mathbf{\Theta}^*$$

holds for all $i = 1, \dots, N_n$. If E_n holds set

$$w^*_{j_i} = \frac{1}{N_n} \quad (i = 1, \dots, N_n) \quad \text{and} \quad w^*_k = 0 \quad (k \in \{1, \dots, K_n\} \setminus \{j_1, \dots, j_{N_n}\})$$

and $\mathbf{w}^* = (w^*_k)_{k=1,\dots,K_n}$, otherwise set $\mathbf{w}^* = 0$.

We will use the following error decomposition:

$$\begin{aligned}
&\mathbf{E}\left\{\varphi(Y \cdot f_n(X))\right\} - \min_{f:[0,1]^{d_1 \times d_2} \to \bar{\mathbb{R}}} \mathbf{E}\left\{\varphi(Y \cdot f(X))\right\} \\
&= \mathbf{E}\left\{\varphi(Y \cdot f_n(X)) \cdot 1_{E_n^c}\right\} \\
&\quad + \mathbf{E}\left\{\mathbf{E}\left\{\varphi(Y \cdot f_{(\hat{\mathbf{w}}, \vartheta^{(t_n)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \cdot 1_{E_n}\right\} \\
&\qquad\qquad - \mathbf{E}\left\{\frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t_n)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \cdot 1_{E_n}\right\} \\
&\quad + \mathbf{E}\left\{\frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t_n)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \cdot 1_{E_n}\right\} \\
&\qquad\qquad - \mathbf{E}\left\{\frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \cdot 1_{E_n}\right\} \\
&\quad + \mathbf{E}\left\{\frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \cdot 1_{E_n}\right\} - \min_{f:[0,1]^{d_1 \times d_2} \to \bar{\mathbb{R}}} \mathbf{E}\left\{\varphi(Y \cdot f(X))\right\} \\
&=: T_{1,n} + T_{2,n} + T_{3,n} + T_{4,n}.
\end{aligned}$$

In the *first step of the proof* we show

$$\mathbf{P}\{E_n^c\} \leq \frac{1}{n}. \tag{20}$$

To do this we consider a sequential choice of the initial weights $\vartheta^{(0)}_1, \dots, \vartheta^{(0)}_{K_n}$. By definition of ϵ_n we know that the probability that none of $\vartheta^{(0)}_1, \dots, \vartheta^{(0)}_{I_n}$ is contained in $\mathbf{\Theta}^*$ is given by

$$(1 - \epsilon_n)^{I_n}.$$

This implies that the probability that there exists $l \in \{1, \dots, N_n\}$ such that none of $\vartheta^{(0)}_{(l-1)\cdot I_n + 1}, \dots, \vartheta^{(0)}_{l \cdot I_n}$ is contained in $\mathbf{\Theta}^*$ is upper bounded by

$$N_n \cdot (1 - \epsilon_n)^{I_n}.$$

(10) implies

$$\mathbf{P}\{E_n^c\} \leq N_n \cdot (1 - \epsilon_n)^{I_n} \leq \frac{1}{n}.$$

In the *second step of the proof* we show

$$T_{1,n} \leq c_{10} \cdot \frac{(\log n)}{n}.$$

To do this, we observe that for $|z| \leq \beta_n$ we have

$$\varphi(z) = \log(1+\exp(-z)) \leq (\log 4) \cdot I_{\{z>-1\}} + \log(2 \cdot \exp(-z)) \cdot I_{\{z \leq -1\}} \leq 3 + |z| \leq c_{11} \cdot \log n,$$

from which we can conclude by the first step of the proof

$$T_{1,n} \leq c_{11} \cdot (\log n) \cdot \mathbf{P}\{E_n^c\} \leq c_{11} \cdot \frac{\log n}{n}.$$

In the *third step of the proof* we show

$$T_{2,n} \quad \leq \quad 0.$$

This follows from the convexity of the logistic loss, which implies

$$\begin{aligned}
&\mathbf{E}\left\{\varphi(Y \cdot f_{(\hat{\mathbf{w}},\vartheta^{(t_n)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \\
&= \mathbf{E}\left\{\varphi(Y \cdot \frac{1}{t_n} \sum_{t=0}^{t_n-1} f_{(\mathbf{w}^{(t)},\vartheta^{(t_n)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \\
&\leq \mathbf{E}\left\{\frac{1}{t_n} \sum_{t=0}^{t_n-1} \varphi(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t_n)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\} \\
&= \frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t_n)})}(X)) \middle| \vartheta^{(0)}, \mathcal{D}_n\right\}.
\end{aligned}$$

In the *fourth step of the proof* we show

$$T_{3,n} \leq 2 \cdot \frac{C_n}{\sqrt{N}_n}.$$

Due to the fact that the logistic loss is Lipschitz continuous with Lipschitz constant 1 and by assumptions (4) and (8) we have

$$\begin{aligned}
&T_{3,n} \\
&\leq \frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\left|\varphi(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t_n)})}(X)) - \varphi(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X))\right|\right\} \\
&\leq \frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{|f_{(\mathbf{w}^{(t)},\vartheta^{(t_n)})}(X) - f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X)|\right\} \\
&\leq \frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\sum_{k=1}^{K_n} w_k^{(t)} \cdot |T_{\beta_n} f_{\vartheta_k^{(t_n)}}(X) - T_{\beta_n} f_{\vartheta_k^{(t)}}(X)|\right\} \\
&\leq \frac{1}{t_n} \sum_{t=0}^{t_n-1} \mathbf{E}\left\{\sum_{k=1}^{K_n} w_k^{(t)} \cdot |f_{\vartheta_k^{(t_n)}}(X) - f_{\vartheta_k^{(t)}}(X)|\right\}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{t_n}\sum_{t=0}^{t_n-1}\mathbf{E}\left\{\sqrt{\sum_{k=1}^{K_n}(w_k^{(t)})^2}\cdot\sqrt{\sum_{k=1}^{K_n}|f_{\vartheta_k^{(t_n)}}(X)-f_{\vartheta_k^{(t)}}(X)|^2}\right\}\\
&\leq \frac{1}{t_n}\sum_{t=0}^{t_n-1}\mathbf{E}\left\{\sqrt{\alpha_n}\cdot\sqrt{\sum_{k=1}^{K_n}C_n^2\cdot\|\vartheta_k^{(t_n)}-\vartheta_k^{(t)}\|^2}\right\}\\
&= \frac{C_n}{\sqrt{N_n}}\cdot\frac{1}{t_n}\sum_{t=0}^{t_n-1}\mathbf{E}\left\{\sqrt{\|\vartheta^{(t_n)}-\vartheta^{(t)}\|^2}\right\}\\
&\leq 2\cdot\frac{C_n}{\sqrt{N_n}}.
\end{aligned}
$$

In the *fifth step of the proof* we apply Lemma 1 to $T_{4,n}$. Set

$$
F((\mathbf{w},\vartheta))=\mathbf{E}\{\varphi(Y\cdot f_{(\mathbf{w},\vartheta)}(X))\}\text{ and }F_t((\mathbf{w},\vartheta))=\varphi(Y_{j_t}\cdot f_{(\mathbf{w},\vartheta)}(X_{j_t})).
$$

Then Lemma 1 implies

$$
\begin{aligned}
T_{4,n} \;\leq\;& \mathbf{E}\left\{\mathbf{E}\left\{\varphi(Y\cdot f_{(\mathbf{w}^*,\vartheta^{(0)})}(X))\big|\vartheta^{(0)},\mathcal{D}_n\right\}\cdot 1_{E_n}\right\}-\min_{f:[0,1]^{d_1\times d_2}\to\bar{\mathbb{R}}}\mathbf{E}\{\varphi(Y\cdot f(X))\}\\
&+\frac{1}{t_n}\sum_{t=1}^{t_n-1}\mathbf{E}\Bigg\{\Bigg|\mathbf{E}\left\{\varphi(Y\cdot f_{(\mathbf{w}^*,\vartheta^{(t)})}(X))\big|\mathcal{D}_n,\vartheta^{(0)}\right\}\\
&\qquad\qquad-\mathbf{E}\left\{\varphi(Y\cdot f_{(\mathbf{w}^*,\vartheta^{(0)})}(X))\big|\mathcal{D}_n,\vartheta^{(0)}\right\}\Bigg|\Bigg\}+\frac{1}{2}\cdot\frac{1}{N_n}+\frac{D_n^2}{2\cdot t_n}\\
&+\frac{1}{t_n}\sum_{t=0}^{t_n-1}\mathbf{E}\left\{<(\nabla_{\mathbf{w}}F)(\mathbf{w}^{(t)},\vartheta^{(t)})-(\nabla_{\mathbf{w}}F_t)(\mathbf{w}^{(t)},\vartheta^{(t)}),\mathbf{w}^{(t)}-\mathbf{w}^*>\right\}.
\end{aligned}
$$

In the *sixth step of the proof* we show

$$
\begin{aligned}
&\mathbf{E}\left\{\mathbf{E}\left\{\varphi(Y\cdot f_{(\mathbf{w}^*,\vartheta^{(0)})}(X))\big|\vartheta^{(0)},\mathcal{D}_n\right\}\cdot 1_{E_n}\right\}-\min_{f:[0,1]^{d_1\times d_2}\to\bar{\mathbb{R}}}\mathbf{E}\{\varphi(Y\cdot f(X))\}\\
&\leq \sup_{\vartheta\in\mathbf{\Theta}^*}\mathbf{E}\{\varphi(Y\cdot T_{\beta_n}f_\vartheta(X))\}-\min_{f:[0,1]^{d_1\times d_2}\to\bar{\mathbb{R}}}\mathbf{E}\{\varphi(Y\cdot f(X))\}\Bigg).
\end{aligned}
$$

This follows from the convexity of the logistic loss, which implies

$$
\begin{aligned}
&\mathbf{E}\left\{\mathbf{E}\left\{\varphi(Y\cdot f_{(\mathbf{w}^*,\vartheta^{(0)})}(X))\big|\vartheta^{(0)},\mathcal{D}_n\right\}\cdot 1_{E_n}\right\}\\
&=\mathbf{E}\left\{\mathbf{E}\left\{\varphi(\frac{1}{N_n}\sum_{k=1}^{N_n}Y\cdot T_{\beta_n}f_{\vartheta_{j_k}^{(0)}}(X))\big|\vartheta^{(0)},\mathcal{D}_n\right\}\cdot 1_{E_n}\right\}\\
&\leq\frac{1}{N_n}\sum_{k=1}^{N_n}\mathbf{E}\left\{\mathbf{E}\left\{\varphi(Y\cdot T_{\beta_n}f_{\vartheta_{j_k}^{(0)}})(X))\big|\vartheta^{(0)},\mathcal{D}_n\right\}\cdot 1_{E_n}\right\}
\end{aligned}
$$

$$\leq \sup_{\vartheta \in \bar{\Theta}^*} \mathbf{E}\left\{\varphi(Y \cdot T_{\beta_n} f_\vartheta(X))\right\}.$$

In the *seventh step of the proof* we show

$$\frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\left|\mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^*,\vartheta^{(t)})}(X))\middle|\mathcal{D}_n, \vartheta^{(0)}\right\} - \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^*,\vartheta^{(0)})}(X))\middle|\mathcal{D}_n, \vartheta^{(0)}\right\}\right|\right\}$$
$$\leq \frac{C_n}{\sqrt{N_n}}.$$

The Lipschitz continuity of the logistic loss and assumption (8) imply

$$\frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\left|\mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^*,\vartheta^{(t)})}(X))\middle|\mathcal{D}_n, \vartheta^{(0)}\right\} - \mathbf{E}\left\{\varphi(Y \cdot f_{(\mathbf{w}^*,\vartheta^{(0)})}(X))\middle|\mathcal{D}_n, \vartheta^{(0)}\right\}\right|\right\}$$
$$\leq \frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\left|\varphi(Y \cdot f_{(\mathbf{w}^*,\vartheta^{(t)})}(X)) - \varphi(Y \cdot f_{(\mathbf{w}^*,\vartheta^{(0)})}(X))\right|\right\}$$
$$\leq \frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\left|f_{(\mathbf{w}^*,\vartheta^{(t)})}(X) - f_{(\mathbf{w}^*,\vartheta^{(0)})}(X)\right|\right\}$$
$$= \frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\left|\sum_{k=1}^{K_n} w_k^* \cdot (T_{\beta_n} f_{\vartheta_k^{(t)}}(X) - T_{\beta_n} f_{\vartheta_k^{(0)}}(X))\right|\right\}$$
$$\leq \frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\sqrt{\sum_{k=1}^{K_n}|w_k^*|^2} \cdot \sqrt{\sum_{k=1}^{K_n}(T_{\beta_n} f_{\vartheta_k^{(t)}}(X) - T_{\beta_n} f_{\vartheta_k^{(0)}}(X))^2}\right\}$$
$$\leq \frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\sqrt{\sum_{k=1}^{K_n}|w_k^*|^2} \cdot \sqrt{\sum_{k=1}^{K_n}(f_{\vartheta_k^{(t)}}(X) - f_{\vartheta_k^{(0)}}(X))^2}\right\}$$
$$\leq \frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\frac{1}{\sqrt{N_n}}\sqrt{\sum_{k=1}^{K_n} C_n^2 \cdot \|\vartheta_k^{(t)} - \vartheta_k^{(0)}\|^2}\right\}$$
$$= \frac{1}{t_n}\sum_{t=1}^{t_n-1} \mathbf{E}\left\{\frac{C_n}{\sqrt{N_n}} \cdot \|\vartheta^{(t)} - \vartheta^{(0)}\|\right\} \leq \frac{C_n}{\sqrt{N_n}}.$$

Let $\mathcal{W}$ be the set of all weight vectors $\mathbf{w} = ((w_k)_{k=1,\ldots,K_n}, (\vartheta_k)_{k=1,\ldots,K_n})$ which satisfy $\vartheta = (\vartheta_k)_{k=1,\ldots,K_n} \in \bar{\Theta}^{K_n}$ and (4). Let $(X_1', Y_1'), \ldots, (X_n', Y_n')$, $\epsilon_1, \ldots, \epsilon_n$ be independent random variables such that (X_j', Y_j') has the same distribution as (X, Y) and such that $\mathbf{P}\{\epsilon_j = 1\} = 1/2 = \mathbf{P}\{\epsilon_j = -1\}$ $(j = 1, \ldots, n)$. In the *eighth step of the proof* we show

$$\frac{1}{t_n}\sum_{t=0}^{t_n-1} \mathbf{E}\left\{< (\nabla_{\mathbf{w}} F)(\mathbf{w}^{(t)}, \vartheta^{(t)}) - (\nabla_{\mathbf{w}} F_t)(\mathbf{w}^{(t)}, \vartheta^{(t)}), \mathbf{w}^{(t)} - \mathbf{w}^* >\right\}$$
$$\leq 4 \cdot \mathbf{E}\left\{\sup_{\substack{\mathbf{w}\in\mathcal{W},\\ k\in\{1,\ldots,K_n\}}} \left(\frac{1}{n}\sum_{i=1}^{n} \epsilon_i \cdot \frac{1}{1+\exp(Y_i \cdot f_{(\mathbf{w},\vartheta)}(X_i))} \cdot Y_i \cdot T_{\beta_n} f_{\vartheta_k}(X_i)\right)\right\}$$

$$+\frac{n\cdot\left(K_n\cdot 2\cdot\beta_n+2\cdot(\beta_n+1)\cdot C_n\cdot\sup_{\mathbf{w}\in\mathcal{W},y\in\{-1,1\},x\in[0,1]^{d_1\times d_2}}\|\nabla_\vartheta\varphi(y\cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty\right)}{t_n}.$$

We have

$$\varphi'(z)=\frac{1}{1+\exp(-z)}\cdot(-\exp(-z))=\frac{-1}{1+\exp(z)},$$

which implies

$$\frac{\partial}{\partial w_j}\varphi\left(Y\cdot\sum_{k=1}^{K_n}w_k\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)=\frac{-Y\cdot T_{\beta_n}f_{\vartheta_j}(X)}{1+\exp\left(Y\cdot\sum_{k=1}^{K_n}w_k\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)}$$

and

$$\begin{aligned}
&\frac{\partial}{\partial w_j}\mathbf{E}\left\{\varphi\left(Y\cdot\sum_{k=1}^{K_n}w_k\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)\right\}\\
&=\lim_{h\to 0}\mathbf{E}\left\{\frac{\varphi\left(Y\cdot\sum_{k=1}^{K_n}(w_k+h\cdot I_{\{k=j\}})\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)-\varphi\left(Y\cdot\sum_{k=1}^{K_n}w_k\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)}{h}\right\}\\
&=\mathbf{E}\left\{\lim_{h\to 0}\frac{\varphi\left(Y\cdot\sum_{k=1}^{K_n}(w_k+h\cdot I_{\{k=j\}})\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)-\varphi\left(Y\cdot\sum_{k=1}^{K_n}w_k\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)}{h}\right\}\\
&=\mathbf{E}\left\{\frac{-Y\cdot T_{\beta_n}f_{\vartheta_j}(X)}{1+\exp\left(Y\cdot\sum_{k=1}^{K_n}w_k\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)}\right\},
\end{aligned}$$

where we have used

$$\begin{aligned}
&\left|\frac{\varphi\left(Y\cdot\sum_{k=1}^{K_n}(w_k+h\cdot I_{\{k=j\}})\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)-\varphi\left(Y\cdot\sum_{k=1}^{K_n}w_k\cdot T_{\beta_n}f_{\vartheta_k}(X)\right)}{h}\right|\\
&=|\varphi'(\xi)|\le 1
\end{aligned}$$

and the dominated convergence theorem in order to interchange limites and expectations above.

Consequently,

$$\begin{aligned}
&\frac{1}{t_n}\sum_{t=0}^{t_n-1}\mathbf{E}\left\{<\left(\nabla_{\mathbf{w}}F\right)(\mathbf{w}^{(t)},\vartheta^{(t)})-\left(\nabla_{\mathbf{w}}F_t\right)(\mathbf{w}^{(t)},\vartheta^{(t)}),\mathbf{w}^{(t)}-\mathbf{w}^*>\right\}\\
&=\mathbf{E}\Bigg\{\frac{1}{t_n}\sum_{t=0}^{t_n-1}\sum_{k=1}^{K_n}\Bigg(\mathbf{E}\left\{\frac{-Y\cdot T_{\beta_n}f_{\vartheta_k^{(t)}}(X)}{1+\exp\left(Y\cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X)\right)}\middle|\mathcal{D}_n,\vartheta^{(0)}\right\}\\
&\qquad\qquad-\frac{-Y_{j_t}\cdot T_{\beta_n}f_{\vartheta_k^{(t)}}(X_{j_t})}{1+\exp\left(Y\cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X_{j_t})\right)}\Bigg)\cdot(w_k^{(t)}-w_k^*)\Bigg\}
\end{aligned}$$

$$\leq \frac{1}{t_n/n} \cdot \sum_{s=1}^{t_n/n} \mathbf{E}\Bigg\{ \frac{1}{n} \sum_{t=(s-1)\cdot n}^{s\cdot n-1} \sum_{k=1}^{K_n} \Bigg(\mathbf{E}\left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k^{(t)}}(X)}{1+\exp\left(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X)\right)} \middle| \mathcal{D}_n, \vartheta^{(0)} \right\}$$
$$- \frac{-Y_{j_t} \cdot T_{\beta_n} f_{\vartheta_k^{(t)}}(X_{j_t})}{1+\exp\left(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X_{j_t})\right)} \Bigg) \cdot (w_k^{(t)} - w_k^*) \Bigg\}.$$

During n gradient descent steps the parameter $(\mathbf{w}, \vartheta)$ changes in supremum norm at most by

$$\begin{aligned}
& n \cdot \lambda_n \cdot \max \Bigg\{ \sup_{\mathbf{w}\in\mathcal{W}, y\in\{-1,1\}, x\in[0,1]^{d_1\times d_2}} \|\nabla_{\mathbf{w}} \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty, \\
& \qquad \sup_{\mathbf{w}\in\mathcal{W}, y\in\{-1,1\}, x\in[0,1]^{d_1\times d_2}} \|\nabla_{\vartheta} \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty \Bigg\} \\
& \leq n \cdot \lambda_n \cdot \left(\beta_n + \sup_{\mathbf{w}\in\mathcal{W}, y\in\{-1,1\}, x\in[0,1]^{d_1\times d_2}} \|\nabla_{\vartheta} \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty \right) \\
& = \frac{n \cdot \left(\beta_n + \sup_{\mathbf{w}\in\mathcal{W}, y\in\{-1,1\}, x\in[0,1]^{d_1\times d_2}} \|\nabla_{\vartheta} \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty \right)}{t_n}.
\end{aligned}$$

Using

$$\begin{aligned}
& \sum_{k=1}^{K_n} \Bigg(\mathbf{E}\left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k^{(t)}}(X)}{1+\exp\left(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X)\right)} \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \\
& \qquad - \mathbf{E}\left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k^{(s\cdot n-1)}}(X)}{1+\exp\left(Y \cdot f_{(\mathbf{w}^{(s\cdot n-1)},\vartheta^{(s\cdot n-1)})}(X)\right)} \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \\
& \qquad - \frac{-Y_{j_t} \cdot T_{\beta_n} f_{\vartheta_k^{(t)}}(X_{j_t})}{1+\exp\left(Y \cdot f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X_{j_t})\right)} \\
& \qquad + \frac{-Y_{j_t} \cdot T_{\beta_n} f_{\vartheta_k^{(s\cdot n-1)}}(X_{j_t})}{1+\exp\left(Y \cdot f_{(\mathbf{w}^{(s\cdot n-1)},\vartheta^{(s\cdot n-1)})}(X_{j_t})\right)} \Bigg) \cdot (w_k^{(t)} - w_k^*) \\
& \leq \sum_{k=1}^{K_n} \Bigg(\beta_n \cdot \mathbf{E}\left\{ |f_{(\mathbf{w}^{(t)},\vartheta^{(t)})}(X) - f_{(\mathbf{w}^{(t)},\vartheta^{(s\cdot n-1)})}(X)| \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \\
& \qquad - \mathbf{E}\left\{ |f_{\vartheta_k^{(t)}}(X) - f_{\vartheta_k^{(s\cdot n-1)}}(X)| \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \Bigg) \cdot |w_k^{(t)} - w_k^*| \\
& \leq \sum_{k=1}^{K_n} \left(\beta_n \cdot C_n \cdot \max_j \|\vartheta_j^{(t)} - \vartheta_j^{(s\cdot n-1)}\|_\infty + C_n \cdot \|\vartheta_k^{(t)} - \vartheta_k^{(s\cdot n-1)}\|_\infty \right) \cdot |w_k^{(t)} - w_k^*|
\end{aligned}$$

$$\leq \frac{2 \cdot (\beta_n + 1) \cdot C_n \cdot n \cdot \sup_{\mathbf{w} \in \mathcal{W}, y \in \{-1,1\}, x \in [0,1]^{d_1 \times d_2}} \|\nabla_\vartheta \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty}{t_n}$$

and

$$\sum_{k=1}^{K_n} \left(\mathbf{E} \left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k^{(s \cdot n-1)}}(X)}{1 + \exp\left(Y \cdot f_{(\mathbf{w}^{(s \cdot n-1)}, \vartheta^{(s \cdot n-1)})}(X)\right)} \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \right.$$
$$\left. - \frac{-Y_{j_t} \cdot T_{\beta_n} f_{\vartheta_k^{(s \cdot n-1)}}(X_{j_t})}{1 + \exp\left(Y_{j_t} \cdot f_{(\mathbf{w}^{(s \cdot n-1)}, \vartheta^{(s \cdot n-1)})}(X_{j_t})\right)} \right) \cdot (w_k^{(t)} - w_k^{(s \cdot n-1)})$$
$$\leq 2 \cdot \sum_{k=1}^{K_n} |w_k^{(t)} - w_k^{(s \cdot n-1)}| \leq 2 \cdot K_n \cdot n \cdot \frac{\beta_n}{t_n}$$

we can conclude

$$\frac{1}{t_n/n} \cdot \sum_{s=1}^{t_n/n} \mathbf{E} \left\{ \frac{1}{n} \sum_{t=(s-1) \cdot n}^{s \cdot n-1} \sum_{k=1}^{K_n} \left(\mathbf{E} \left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k^{(t)}}(X)}{1 + \exp\left(Y \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t)})}(X)\right)} \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \right. \right.$$
$$\left. \left. - \frac{-Y_{j_t} \cdot T_{\beta_n} f_{\vartheta_k^{(t)}}(X_{j_t})}{1 + \exp\left(Y \cdot f_{(\mathbf{w}^{(t)}, \vartheta^{(t)})}(X_{j_t})\right)} \right) \cdot (w_k^{(t)} - w_k^*) \right\}$$
$$\leq \frac{1}{t_n/n} \cdot \sum_{s=1}^{t_n/n} \mathbf{E} \left\{ \frac{1}{n} \sum_{t=(s-1) \cdot n}^{s \cdot n-1} \sum_{k=1}^{K_n} \left(\mathbf{E} \left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k^{(s \cdot n-1)}}(X)}{1 + \exp\left(Y \cdot f_{(\mathbf{w}^{(s \cdot n-1)}, \vartheta^{(s \cdot n-1)})}(X)\right)} \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \right. \right.$$
$$\left. \left. - \frac{-Y_{j_t} \cdot T_{\beta_n} f_{\vartheta_k^{(s \cdot n-1)}}(X_{j_t})}{1 + \exp\left(Y \cdot f_{(\mathbf{w}^{(s \cdot n-1)}, \vartheta^{(s \cdot n-1)})}(X_{j_t})\right)} \right) \cdot (w_k^{(s \cdot n-1)} - w_k^*) \right\}$$
$$+ \frac{n \cdot \left(K_n \cdot 2 \cdot \beta_n + 2 \cdot (\beta_n + 1) \cdot C_n \cdot \sup_{\mathbf{w} \in \mathcal{W}, y \in \{-1,1\}, x \in [0,1]^{d_1 \times d_2}} \|\nabla_\vartheta \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty \right)}{t_n}.$$

We continue by deriving an upper bound on the first term of the sum of the right-hand side above. We have

$$\frac{1}{t_n/n} \cdot \sum_{s=1}^{t_n/n} \mathbf{E} \left\{ \frac{1}{n} \sum_{t=(s-1) \cdot n}^{s \cdot n-1} \sum_{k=1}^{K_n} \left(\mathbf{E} \left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k^{(s \cdot n-1)}}(X)}{1 + \exp\left(Y \cdot f_{(\mathbf{w}^{(s \cdot n-1)}, \vartheta^{(s \cdot n-1)})}(X)\right)} \middle| \mathcal{D}_n, \vartheta^{(0)} \right\} \right. \right.$$
$$\left. \left. - \frac{-Y_{j_t} \cdot T_{\beta_n} f_{\vartheta_k^{(s \cdot n-1)}}(X_{j_t})}{1 + \exp\left(Y_{j_t} \cdot f_{(\mathbf{w}^{(s \cdot n-1)}, \vartheta^{(s \cdot n-1)})}(X_{j_t})\right)} \right) \cdot (w_k^{(s \cdot n-1)} - w_k^*) \right\}$$
$$\leq \frac{1}{t_n/n} \cdot \sum_{s=1}^{t_n/n} \mathbf{E} \left\{ \sup_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{t=(s-1) \cdot n}^{s \cdot n-1} \sum_{k=1}^{K_n} \left(\mathbf{E} \left\{ \frac{-Y \cdot T_{\beta_n} f_{\vartheta_k}(X)}{1 + \exp\left(Y \cdot f_{(\mathbf{w},\vartheta)}(X)\right)} \right\} \right.$$

$$
\begin{aligned}
&\qquad\qquad -\frac{-Y_{j_t}\cdot T_{\beta_n} f_{\vartheta_k}(X_{j_t})}{1+\exp\left(Y_{j_t}\cdot f_{(\mathbf{w},\vartheta)}(X_{j_t})\right)}\Bigg)\cdot(w_k-w_k^*)\Bigg\}\\
&=\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\Bigg(\mathbf{E}\left\{\frac{-Y\cdot T_{\beta_n} f_{\vartheta_k}(X)}{1+\exp\left(Y\cdot f_{(\mathbf{w},\vartheta)}(X)\right)}\right\}\\
&\qquad\qquad -\frac{-Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\Bigg)\cdot(w_k-w_k^*)\Bigg\}\\
&=\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\Bigg(\mathbf{E}\left\{\frac{-Y_j'\cdot T_{\beta_n} f_{\vartheta_k}(X_j')}{1+\exp\left(Y_j'\cdot f_{(\mathbf{w},\vartheta)}(X_j')\right)}\middle|\mathcal{D}_n\right\}\\
&\qquad\qquad -\frac{-Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\Bigg)\cdot(w_k-w_k^*)\Bigg\}\\
&\le\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\Bigg(\frac{-Y_j'\cdot T_{\beta_n} f_{\vartheta_k}(X_j')}{1+\exp\left(Y_j'\cdot f_{(\mathbf{w},\vartheta)}(X_j')\right)}\\
&\qquad\qquad -\frac{-Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\Bigg)\cdot(w_k-w_k^*)\Bigg\}\\
&=\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\epsilon_j\cdot\Bigg(\frac{-Y_j'\cdot T_{\beta_n} f_{\vartheta_k}(X_j')}{1+\exp\left(Y_j'\cdot f_{(\mathbf{w},\vartheta)}(X_j')\right)}\\
&\qquad\qquad -\frac{-Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\Bigg)\cdot(w_k-w_k^*)\Bigg\}\\
&\le\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\epsilon_j\cdot\left(\frac{-Y_j'\cdot T_{\beta_n} f_{\vartheta_k}(X_j')}{1+\exp\left(Y_j'\cdot f_{(\mathbf{w},\vartheta)}(X_j')\right)}\right)\cdot(w_k-w_k^*)\Bigg\}\\
&\quad+\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\epsilon_j\cdot\left(\frac{Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\right)\cdot(w_k-w_k^*)\Bigg\}\\
&=2\cdot\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\epsilon_j\cdot\left(\frac{Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\right)\cdot(w_k-w_k^*)\Bigg\}\\
&\le 2\cdot\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}\epsilon_j\cdot\frac{Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\cdot(w_k-w_k^*)_+\Bigg\}\\
&\quad+2\cdot\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{K_n}(-\epsilon_j)\cdot\frac{Y_j\cdot T_{\beta_n} f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\cdot(w_k^*-w_k)_+\Bigg\}\\
&\le 2\cdot\mathbf{E}\Bigg\{\sup_{\mathbf{w}\in\mathcal{W}}\sum_{k=1}^{K_n}\sup_{\bar{\mathbf{w}}\in\mathcal{W},\bar{k}\in\{1,\ldots,K_n\}}\frac{1}{n}\sum_{j=1}^{n}\epsilon_j\cdot\frac{Y_j\cdot T_{\beta_n} f_{\bar{\vartheta}_{\bar{k}}}(X_j)}{1+\exp\left(Y_j\cdot f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_j)\right)}\cdot(w_k-w_k^*)_+\Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
&+2\cdot\mathbf{E}\left\{\sup_{\mathbf{w}\in\mathcal{W}}\sum_{k=1}^{K_n}\sup_{\bar{\mathbf{w}}\in\mathcal{W},\bar{k}\in\{1,\dots,K_n\}}\frac{1}{n}\sum_{j=1}^{n}(-\epsilon_j)\cdot\frac{Y_j\cdot T_{\beta_n}f_{\bar{\vartheta}_{\bar{k}}}(X_j)}{1+\exp\left(Y_j\cdot f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_j)\right)}\cdot(w_k^*-w_k)_+\right\}\\
&\le 2\cdot\mathbf{E}\left\{\sup_{\mathbf{w}\in\mathcal{W},k\in\{1,\dots,K_n\}}\frac{1}{n}\sum_{j=1}^{n}\epsilon_j\cdot\frac{Y_j\cdot T_{\beta_n}f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\cdot\sup_{\mathbf{w}\in\mathcal{W}}\sum_{k=1}^{K_n}(w_k-w_k^*)_+\right\}\\
&\quad+2\cdot\mathbf{E}\left\{\sup_{\mathbf{w}\in\mathcal{W},k\in\{1,\dots,K_n\}}\frac{1}{n}\sum_{j=1}^{n}(-\epsilon_j)\cdot\frac{Y_j\cdot T_{\beta_n}f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\cdot\sup_{\mathbf{w}\in\mathcal{W}}\sum_{k=1}^{K_n}(w_k^*-w_k)_+\right\}\\
&\le 4\cdot\mathbf{E}\left\{\sup_{\mathbf{w}\in\mathcal{W},k\in\{1,\dots,K_n\}}\frac{1}{n}\sum_{j=1}^{n}\epsilon_j\cdot\frac{Y_j\cdot T_{\beta_n}f_{\vartheta_k}(X_j)}{1+\exp\left(Y_j\cdot f_{(\mathbf{w},\vartheta)}(X_j)\right)}\right\}.
\end{aligned}
$$

In the *ninth step of the proof* we derive an upper bound on

$$
\mathbf{E}\left\{\sup_{\substack{\mathbf{w}\in\mathcal{W},\\ k\in\{1,\dots,K_n\}}}\left(\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\cdot\frac{1}{1+\exp(Y_i\cdot f_{(\mathbf{w},\vartheta)}(X_i))}\cdot Y_i\cdot T_{\beta_n}f_{\vartheta_k}(X_i)\right)\right\}.
$$

To do this we use a contraction style argument. Because of the independence of the random variables we can compute the expectation by first computing the expectation with respect to ϵ_1 and then by computing the expectation with respect to all other random variables. This yields that the last term above is equal to

$$
\begin{aligned}
&\frac{1}{2}\cdot\mathbf{E}\left\{\sup_{\substack{\mathbf{w}\in\mathcal{W},\\ k\in\{1,\dots,K_n\}}}\left(\frac{1}{n}\sum_{i=2}^{n}\epsilon_i\cdot\frac{1}{1+\exp(Y_i\cdot f_{(\mathbf{w},\vartheta)}(X_i))}\cdot Y_i\cdot T_{\beta_n}f_{\vartheta_k}(X_i)\right.\right.\\
&\qquad\left.\left.+\frac{1}{1+\exp(Y_1\cdot f_{(\mathbf{w},\vartheta)}(X_1))}\cdot Y_1\cdot T_{\beta_n}f_{\vartheta_k}(X_1)\right)\right\}\\
&\quad+\frac{1}{2}\cdot\mathbf{E}\left\{\sup_{\substack{\mathbf{w}\in\mathcal{W},\\ k\in\{1,\dots,K_n\}}}\left(\frac{1}{n}\sum_{i=2}^{n}\epsilon_i\cdot\frac{1}{1+\exp(Y_i\cdot f_{(\mathbf{w},\vartheta)}(X_i))}\cdot Y_i\cdot T_{\beta_n}f_{\vartheta_k}(X_i)\right.\right.\\
&\qquad\left.\left.-\frac{1}{1+\exp(Y_1\cdot f_{(\mathbf{w},\vartheta)}(X_1))}\cdot Y_1\cdot T_{\beta_n}f_{\vartheta_k}(X_1)\right)\right\}\\
&=\frac{1}{2}\cdot\mathbf{E}\left\{\sup_{\substack{\mathbf{w},\bar{\mathbf{w}}\in\mathcal{W},\\ k,\bar{k}\in\{1,\dots,K_n\}}}\left(\frac{1}{n}\sum_{i=2}^{n}\epsilon_i\cdot\frac{1}{1+\exp(Y_i\cdot f_{(\mathbf{w},\vartheta)}(X_i))}\cdot Y_i\cdot T_{\beta_n}f_{\vartheta_k}(X_i)\right.\right.\\
&\qquad+\frac{1}{n}\sum_{i=2}^{n}\epsilon_i\cdot\frac{1}{1+\exp(Y_i\cdot f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_i))}\cdot Y_i\cdot T_{\beta_n}f_{\bar{\vartheta}_{\bar{k}}}(X_i)\\
&\qquad+\frac{1}{1+\exp(Y_1\cdot f_{(\mathbf{w},\vartheta)}(X_1))}\cdot Y_1\cdot T_{\beta_n}f_{\vartheta_k}(X_1)\\
&\qquad\left.\left.-\frac{1}{1+\exp(Y_1\cdot f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_1))}\cdot Y_1\cdot T_{\beta_n}f_{\bar{\vartheta}_{\bar{k}}}(X_1)\right)\right\}
\end{aligned}
$$

$$\leq \frac{1}{2}\cdot \mathbf{E}\Bigg\{ \sup_{\substack{\mathbf{w},\bar{\mathbf{w}}\in\mathcal{W},\\ k,\bar{k}\in\{1,\dots,K_n\}}} \Bigg(\frac{1}{n}\sum_{i=2}^{n}\epsilon_i \cdot \frac{1}{1+\exp(Y_i\cdot f_{(\mathbf{w},\vartheta)}(X_i))}\cdot Y_i \cdot T_{\beta_n} f_{\vartheta_k}(X_i) \\ + \frac{1}{n}\sum_{i=2}^{n}\epsilon_i \cdot \frac{1}{1+\exp(Y_i\cdot f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_i))}\cdot Y_i \cdot T_{\beta_n} f_{\bar{\vartheta}_{\bar{k}}}(X_i) \\ + \beta_n \cdot |f_{(\mathbf{w},\vartheta)}(X_1) - f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_1)| + |T_{\beta_n} f_{\vartheta_k}(X_1) - T_{\beta_n} f_{\bar{\vartheta}_{\bar{k}}}(X_1)| \Bigg) \Bigg\}.$$

The sum inside the supremum above does not change its value if $(\mathbf{w}, k)$ is interchanged with $(\bar{\mathbf{w}}, \bar{k})$. Consequently we can assume without loss of generality that $f_{(\mathbf{w},\vartheta)}(X_1) - f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_1)$ is positive or that it is negative. Set $\bar{\epsilon}_1 = \bar{\epsilon}_1(\epsilon_1, X_1, Y_1, \dots, X_n, Y_n) = \epsilon_1$ if the functions $f_{(\mathbf{w},\vartheta)}(X_1) - f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_1)$ and $T_{\beta_n} f_{\vartheta_k}(X_1) - T_{\beta_n} f_{\bar{\vartheta}_{\bar{k}}}(X_1)$ which "attain" the above supremum have the same sign, and set it equal to $-\epsilon_1$ otherwise. Then the right-hand side above is equal to

$$\frac{1}{2}\cdot \mathbf{E}\Bigg\{ \sup_{\substack{\mathbf{w},\bar{\mathbf{w}}\in\mathcal{W},\\ k,\bar{k}\in\{1,\dots,K_n\}}} \Bigg(\frac{1}{n}\sum_{i=2}^{n}\epsilon_i \cdot \frac{1}{1+\exp(Y_i\cdot f_{(\mathbf{w},\vartheta)}(X_i))}\cdot Y_i \cdot T_{\beta_n} f_{\vartheta_k}(X_i) \\ + \frac{1}{n}\sum_{i=2}^{n}\epsilon_i \cdot \frac{1}{1+\exp(Y_i\cdot f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_i))}\cdot Y_i \cdot T_{\beta_n} f_{\bar{\vartheta}_{\bar{k}}}(X_i) \\ + \beta_n \cdot \epsilon_1 \cdot (f_{(\mathbf{w},\vartheta)}(X_1) - f_{(\bar{\mathbf{w}},\bar{\vartheta})}(X_1)) + \bar{\epsilon}_1 \cdot (T_{\beta_n} f_{\vartheta_k}(X_1) - T_{\beta_n} f_{\bar{\vartheta}_{\bar{k}}}(X_1)) \Bigg) \Bigg\}$$

$$\leq \mathbf{E}\Bigg\{ \sup_{\substack{\mathbf{w},\in\mathcal{W},\\ k\in\{1,\dots,K_n\}}} \Bigg(\frac{1}{n}\sum_{i=2}^{n}\epsilon_i \cdot \frac{1}{1+\exp(Y_i\cdot f_{(\mathbf{w},\vartheta)}(X_i))}\cdot Y_i \cdot T_{\beta_n} f_{\vartheta_k}(X_i) \\ + \beta_n \cdot \epsilon_1 \cdot f_{(\mathbf{w},\vartheta)}(X_1) + \bar{\epsilon}_1 \cdot T_{\beta_n} f_{\vartheta_k}(X_1) \Bigg) \Bigg\},$$

where we have used that conditioned on $(X_1, Y_1), \dots, (X_n, Y_n)$ the random vector $(\epsilon_1, \bar{\epsilon}_1)$ has the same distribution as the random vector $(-\epsilon_1, -\bar{\epsilon}_1)$.

Arguing in the same way for $k = 2, \dots, n$ we see that we can upper bound the term on the right-hand side above by

$$\mathbf{E}\Bigg\{ \sup_{\substack{\mathbf{w}\in\mathcal{W},\\ k\in\{1,\dots,K_n\}}} \frac{1}{n}\sum_{i=1}^{n}\Bigg(\beta_n \cdot \epsilon_i \cdot f_{(\mathbf{w},\vartheta)}(X_i) + \bar{\epsilon}_i \cdot T_{\beta_n} f_{\vartheta_k}(X_i) \Bigg) \Bigg\}$$

$$\leq \beta_n \cdot \mathbf{E}\Bigg\{ \sup_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n}\epsilon_i \cdot f_{(\mathbf{w},\vartheta)}(X_i) \Bigg\} + \mathbf{E}\Bigg\{ \sup_{\substack{\mathbf{w}\in\mathcal{W},\\ k\in\{1,\dots,K_n\}}} \frac{1}{n}\sum_{i=1}^{n}\bar{\epsilon}_i \cdot T_{\beta_n} f_{\vartheta_k}(X_i) \Bigg\}$$

$$
\begin{aligned}
&\leq \beta_n \cdot \sup_{x_1,\ldots,x_n \in \mathbb{R}^{d_1 \times d_2}} \mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot f_{(\mathbf{w},\vartheta)}(x_i) \right\} \\
&\quad + \sup_{x_1,\ldots,x_n \in \mathbb{R}^{d_1 \times d_2}} \mathbf{E}\left\{ \sup_{\substack{\mathbf{w} \in \mathcal{W}, \\ k \in \{1,\ldots,K_n\}}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot T_{\beta_n} f_{\vartheta_k}(x_i) \right\} \\
&= \beta_n \cdot \sup_{x_1,\ldots,x_n \in \mathbb{R}^{d_1 \times d_2}} \mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot f_{(\mathbf{w},\vartheta)}(x_i) \right\} \\
&\quad + \sup_{x_1,\ldots,x_n \in \mathbb{R}^{d_1 \times d_2}} \mathbf{E}\left\{ \sup_{\vartheta \in \bar{\Theta}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot T_{\beta_n} f_{\vartheta}(x_i) \right\},
\end{aligned}
$$

where the last equality follows from

$$
\{T_{\beta_n} f_{\vartheta_k} \,:\, \mathbf{w} \in \mathcal{W}, k \in \{1,\ldots,K_n\}\} = \{T_{\beta_n} f_{\vartheta_1} \,:\, \mathbf{w} \in \mathcal{W}\}. \tag{21}
$$

In the *tenth step of the proof* we show for $x_1, \ldots, x_n \in \mathbb{R}^{d_1 \times d_2}$ arbitrary

$$
\mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot f_{(\mathbf{w},\vartheta)}(x_i) \right\} \leq \mathbf{E}\left\{ \sup_{\vartheta \in \bar{\Theta}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot (T_{\beta_n} f_{\vartheta}(x_i)) \right| \right\}.
$$

We have

$$
\begin{aligned}
&\mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot f_{(\mathbf{w},\vartheta)}(x_i) \right\} \\
&= \mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot \sum_{j=1}^{K_n} w_j \cdot (T_{\beta_n} f_{\vartheta_j}(x_i)) \right\} \\
&= \mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \sum_{j=1}^{K_n} w_j \cdot \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot (T_{\beta_n} f_{\vartheta_j}(x_i)) \right\} \\
&\leq \mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \sum_{j=1}^{K_n} |w_j| \cdot \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot (T_{\beta_n} f_{\vartheta_j}(x_i)) \right| \right\} \\
&\leq \mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}} \sum_{j=1}^{K_n} |w_j| \cdot \sup_{\mathbf{w} \in \mathcal{W}, k \in \{1,\ldots,K_n\}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot (T_{\beta_n} f_{\vartheta_k}(x_i)) \right| \right\} \\
&\leq 1 \cdot \mathbf{E}\left\{ \sup_{\mathbf{w} \in \mathcal{W}, k \in \{1,\ldots,K_n\}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot (T_{\beta_n} f_{\vartheta_k}(x_i)) \right| \right\} \\
&= \mathbf{E}\left\{ \sup_{\vartheta \in \bar{\Theta}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot (T_{\beta_n} f_{\vartheta}(x_i)) \right| \right\},
\end{aligned}
$$

where the last equality followed from (21).

Summarizing the above results, the proof is complete. □

4.2. Proof of Theorem 2

In the proof of Theorem 2 we will need the following auxiliary results.

4.2.1. Using the logistic risk for classification

Lemma 2 *Let φ be the logistic loss. Let $(X,Y),(X_1,Y_1),\dots,(X_n,Y_n)$ and f^*, $\mathcal{D}_n, f_n$ and $\hat{C}_n$ as in Sections 1 and 3, and set*

$$f_{\varphi^*} = \arg \min_{f:[0,1]^{d_1\times d_2}\to\bar{\mathbb{R}}} \mathbf{E}\left\{\varphi(Y\cdot f(X))\right\}.$$

a) *Then*

$$\begin{aligned}&\mathbf{P}\left\{Y \neq \hat{C}_n(X)|\mathcal{D}_n\right\} - \mathbf{P}\left\{Y\neq f^*(X)\right\}\\ &\leq \frac{1}{\sqrt{2}}\cdot\left(\mathbf{E}\left\{\varphi(Y\cdot f_n(X))|\mathcal{D}_n\right\} - \mathbf{E}\left\{\varphi(Y\cdot f_{\varphi^*}(X))\right\}\right)^{1/2}\end{aligned}$$

holds.
b) *Then*

$$\begin{aligned}&\mathbf{P}\left\{Y \neq \hat{C}_n(X)|\mathcal{D}_n\right\} - \mathbf{P}\left\{Y\neq f^*(X)\right\}\\ &\leq 2\cdot\left(\mathbf{E}\left\{\varphi(Y\cdot f_n(X))|\mathcal{D}_n\right\} - \mathbf{E}\left\{\varphi(Y\cdot f_{\varphi^*}(X))\right\}\right) + 4\cdot\mathbf{E}\left\{\varphi(Y\cdot f_{\varphi^*}(X))\right\}.\end{aligned}$$

holds.
c) *Assume that*

$$\mathbf{P}\left\{|f_{\varphi^*}(X)| > \tilde{F}_n\right\} \geq 1 - e^{-\tilde{F}_n}$$

for a given sequence $\{\tilde{F}_n\}_{n\in\mathbb{N}}$ with $\tilde{F}_n\to\infty$. Then

$$\mathbf{E}\left\{\varphi(Y\cdot f_{\varphi^*}(X))\right\} \leq c_{12}\cdot\tilde{F}_n\cdot e^{-\tilde{F}_n}$$

holds.

Proof. **a)** This result follows from Theorem 2.1 in Zhang (2004), where we choose $s=2$ and $c=2^{-1/2}$.
b) This result follows from Lemma 1 b) in Kohler and Langer (2020).
c) This result follows from Lemma 3 in Kim, Ohn and Kim (2019). □

4.2.2. Lipschitz property of the networks

Lemma 3 *Let $\mathcal{F}_n = \{f_\vartheta : \vartheta\in\Theta\}$ be the class of deep convolutional neural networks introduced in Subsection 3.1 (cf., (14)). Set*

$$M_{\max} = \max\{M_1,\dots,M_{L_n^{(1)}}\} \quad \textit{and} \quad k_{max} = \max\{k_1,\dots,k_{L_n^{(1)}}\}.$$

Let $\vartheta, \bar{\vartheta} \in \Theta$ such that

$$\left\|\vartheta - \bar{\vartheta}\right\|_\infty \leq 1 \tag{22}$$

holds and all weights in f_ϑ are bounded in absolute value by $B_n \geq 0$. Then

$$\|f_\vartheta - f_{\bar{\vartheta}}\|_{\infty,[0,1]^{d_1 \times d_2}} \leq 7 \cdot L_n^{(2)} \cdot L_n^{(1)} \cdot k_{max}^{L_n^{(1)}+1} \cdot (M_{max}^2+1)^{L_n^{(1)}} \cdot (B_n+1)^{L_n^{(1)}+3} \cdot \|\vartheta - \bar{\vartheta}\|_\infty.$$

Proof. Let $o^{(l)}$, g and $\bar{o}^{(l)}$, $\bar{g}$ be defined as in Section 3.1 using the weights in ϑ and $\bar{\vartheta}$, resp., and set

$$\|o^{(l)}(x)\|_\infty = \max_{(i,j),s_2} |o^{(l)}_{(i,j),s_2}(x)|.$$

Then $|\sigma(z)| \leq |z|$ implies

$$\begin{aligned} |o^{(r)}_{(i,j),s_2}(x)| &\leq k_{max} \cdot (M_{max}^2+1) \cdot B_n \cdot \max\{\|o^{(r-1)}(x)\|_\infty, 1\} \\ &\leq k_{max}^r \cdot (M_{max}^2+1)^r \cdot B_n^r. \end{aligned}$$

Using this together with $|\sigma(z_1) - \sigma(z_2)| \leq |z_1 - z_2|$ we conclude

$$\begin{aligned} &|o^{(r)}_{(i,j),s_2}(x) - \bar{o}^{(r)}_{(i,j),s_2}(x)| \\ &\leq \left| \sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\}, \\ (i+t_1-1, j+t_2-1)}} \left(w^{(r)}_{t_1,t_2,s_1,s_2} \cdot o^{(r-1)}_{i+t_1-1,j+t_2-1,s_1}(x) - \bar{w}^{(r)}_{t_1,t_2,s_1,s_2} \cdot \bar{o}^{(r-1)}_{i+t_1-1,j+t_2-1,s_1}(x) \right) \right. \\ &\qquad \left. + w^{(r)}_{s_2} - \bar{w}^{(r)}_{s_2} \right| \\ &\leq \sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\}, \\ (i+t_1-1, j+t_2-1)}} \left| w^{(r)}_{t_1,t_2,s_1,s_2} - \bar{w}^{(r)}_{t_1,t_2,s_1,s_2} \right| \cdot \left| o^{(r-1)}_{i+t_1-1,j+t_2-1,s_1}(x) \right| + |w^{(r)}_{s_2} - \bar{w}^{(r)}_{s_2}| \\ &\quad + \sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\}, \\ (i+t_1-1, j+t_2-1)}} |\bar{w}^{(r)}_{t_1,t_2,s_1,s_2}| \cdot \left| o^{(r-1)}_{i+t_1-1,j+t_2-1,s_1}(x) - \bar{o}^{(r-1)}_{i+t_1-1,j+t_2-1,s_1}(x) \right| \\ &\leq k_{max} \cdot (M_{max}^2+1) \cdot \left(\|\vartheta - \bar{\vartheta}\|_\infty \cdot k_{max}^{r-1} \cdot (M_{max}^2+1)^{r-1} \cdot B_n^{r-1} + (B_n+1) \cdot \|o^{(r-1)} - \bar{o}^{(r-1)}\|_\infty \right) \\ &\leq r \cdot k_{max}^r \cdot (M_{max}^2+1)^r \cdot (B_n+1)^{r-1} \cdot \|\vartheta - \bar{\vartheta}\|_\infty. \end{aligned}$$

From this and

$$|\max\{a_1, \dots, a_n\} - \max\{b_1, \dots, b_n\}| \leq \max\{|a_1 - b_1|, \dots, |a_n - b_n|\}$$

we conclude

$$\begin{aligned} &|f_{\mathbf{w},\mathbf{w}_{bias},\mathbf{w}_{out}}(x) - f_{\bar{\mathbf{w}},\bar{\mathbf{w}}_{bias},\bar{\mathbf{w}}_{out}}(x)| \\ &\leq k_{max} \cdot \|\vartheta - \bar{\vartheta}\|_\infty \cdot B_n \cdot k_{max}^{L_n^{(1)}} \cdot (M_{max}^2+1)^{L_n^{(1)}} \cdot B_n^{L_n^{(1)}} \end{aligned}$$

$$
\begin{aligned}
&+k_{max}\cdot(B_n+1)\cdot L_n^{(1)}\cdot k_{max}^{L_n^{(1)}}\cdot(M_{max}^2+1)^{L_n^{(1)}}\cdot(B_n+1)^{L_n^{(1)}-1}\cdot\|\vartheta-\bar\vartheta\|_\infty\\
&\le (L_n^{(1)}+1)\cdot k_{max}^{L_n^{(1)}+1}\cdot(M_{max}^2+1)^{L_n^{(1)}}\cdot(B_n+1)^{L_n^{(1)}+1}\cdot\|\vartheta-\bar\vartheta\|_\infty.
\end{aligned}
$$

This implies

$$
\begin{aligned}
&|g(x)-\bar g(x)|\\
&\le (L_n^{(2)}+1)\cdot\|\vartheta-\bar\vartheta\|_\infty\cdot 2\cdot B_n\cdot k_{max}\cdot B_n\cdot k_{max}^{L_n^{(1)}}\cdot(M_{max}^2+1)^{L_n^{(1)}}\cdot(B_n+1)^{L_n^{(1)}}\\
&\quad +L_n^{(2)}\cdot(B_n+1)\cdot\Bigg(2\cdot\|\vartheta-\bar\vartheta\|_\infty\cdot k_{max}\cdot B_n\cdot k_{max}^{L_n^{(1)}}\cdot(M_{max}^2+1)^{L_n^{(1)}}\cdot(B_n+1)^{L_n^{(1)}}\\
&\qquad\qquad +(B_n+1)\cdot(L_n^{(1)}+1)\cdot k_{max}^{L_n^{(1)}+1}\cdot(M_{max}^2+1)^{L_n^{(1)}}\cdot(B_n+1)^{L_n^{(1)}+1}\cdot\|\vartheta-\bar\vartheta\|_\infty\Bigg)\\
&\le 7\cdot L_n^{(2)}\cdot L_n^{(1)}\cdot k_{max}^{L_n^{(1)}+1}\cdot(M_{max}^2+1)^{L_n^{(1)}}\cdot(B_n+1)^{L_n^{(1)}+3}\cdot\|\vartheta-\bar\vartheta\|_\infty.
\end{aligned}
$$

□

4.2.3. Approximation error

In our next lemma we present a bound on the error we make in case that we replace the functions $g_{k,s}$ in a hierarchical model by some approximations of them.

Lemma 4 *Let $d_1,d_2,t\in\mathbb{N}$ and $l\in\mathbb{N}$ with $2^l\le\min\{d_1,d_2\}$. For $a\in\{1,\dots,t\}$, set $I=\{0,1,\dots,2^l-1\}\times\{0,1,\dots,2^l-1\}$ and define*

$$
m_a(x)=\max_{(i,j)\in\mathbb{Z}^2\,:\,(i,j)+I\subseteq\{1,\dots,d_1\}\times\{1,\dots,d_2\}} f_a\left(x_{(i,j)+I}\right)
$$

and

$$
\bar m_a(x)=\max_{(i,j)\in\mathbb{Z}^2\,:\,(i,j)+I\subseteq\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \bar f_a\left(x_{(i,j)+I}\right),
$$

where f_a and $\bar f_a$ satisfy

$$
f_a=f_{l,1}^{(a)}\quad\text{and}\quad\bar f_a=\bar f_{l,1}^{(a)}
$$

for some $f_{k,s}^{(a)},\bar f_{k,s}^{(a)}:\mathbb{R}^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}}\to\mathbb{R}$ recursively defined by

$$
\begin{aligned}
f_{k,s}^{(a)}(x) &= g_{k,s}^{(a)}\big(f_{k-1,4\cdot(s-1)+1}^{(a)}(x_{\{1,\dots,2^{k-1}\}\times\{1,\dots,2^{k-1}\}}),\\
&\qquad f_{k-1,4\cdot(s-1)+2}^{(a)}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{1,\dots,2^{k-1}\}}),\\
&\qquad f_{k-1,4\cdot(s-1)+3}^{(a)}(x_{\{1,\dots,2^{k-1}\}\times\{2^{k-1}+1,\dots,2^k\}}),\\
&\qquad f_{k-1,4\cdot s}^{(a)}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{2^{k-1}+1,\dots,2^k\}})\big)
\end{aligned}
$$

and

$$
\begin{aligned}
\bar f_{k,s}^{(a)}(x) &= \bar g_{k,s}^{(a)}\big(\bar f_{k-1,4\cdot(s-1)+1}^{(a)}(x_{\{1,\dots,2^{k-1}\}\times\{1,\dots,2^{k-1}\}}),
\end{aligned}
$$

$$\bar{f}^{(a)}_{k-1,4\cdot(s-1)+2}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{1,\dots,2^{k-1}\}}),$$
$$\bar{f}^{(a)}_{k-1,4\cdot(s-1)+3}(x_{\{1,\dots,2^{k-1}\}\times\{2^{k-1}+1,\dots,2^k\}}),$$
$$\bar{f}^{(a)}_{k-1,4\cdot s}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{2^{k-1}+1,\dots,2^k\}})\big)$$

for $k = 2,\dots,l, s = 1,\dots,4^{l-k}$, *and*

$$f^{(a)}_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2}) = g^{(a)}_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2})$$

and

$$\bar{f}^{(a)}_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2}) = \bar{g}^{(a)}_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2})$$

for $s = 1,\dots,4^{l-1}$, *where*

$$g^{(a)}_{k,s} : \mathbb{R}^4 \to [0,1] \text{ and } \bar{g}^{(a)}_{k,s} : \mathbb{R}^4 \to \mathbb{R}$$

are functions for $a \in \{1,\dots,t\}$, $k \in \{1,\dots,l\}$ *and* $s \in \{1,\dots,4^{l-k}\}$. *Furthermore, let* $g : \mathbb{R}^t \to [0,1]$ *and* $\bar{g} : \mathbb{R}^t \to \mathbb{R}$ *be functions. Assume that all restrictions* $g^{(a)}_{k,s}|_{[-2,2]^4} : [-2,2]^4 \to [0,1]$ *and* $g|_{[-2,2]^t} : [-2,2]^t \to [0,1]$ *are Lipschitz continuous with respect to the Euclidean distance with Lipschitz constant* $C > 0$ *and for all* $a \in \{1,\dots,t\}$, $k \in \{1,\dots,l\}$ *and* $s \in \{1,\dots,4^{l-k}\}$ *we assume that*

$$\left\|\bar{g}^{(a)}_{k,s}\right\|_{[-2,2]^4,\infty} \le 2. \tag{23}$$

Then for any $x \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ *it holds:*

$$\begin{aligned}
&|g(m_1(x),\dots,m_t(x)) - \bar{g}(\bar{m}_1(x),\dots,\bar{m}_t(x))| \\
&\le \sqrt{t}\cdot(2C+1)^l \\
&\quad\cdot \max_{a\in\{1,\dots,t\},j\in\{1,\dots,l\},s\in\{1,\dots,4^{l-j}\}} \left\{\|g^{(a)}_{j,s} - \bar{g}^{(a)}_{j,s}\|_{[-2,2]^4,\infty}, \|g-\bar{g}\|_{[-2,2]^t,\infty}\right\}.
\end{aligned}$$

Proof. See Lemma 1 in Kohler, Krzyżak and Walter (2022). □

Lemma 5 *Let* $d \in \mathbb{N}$, *let* $f : \mathbb{R}^d \to \mathbb{R}$ *be* (p,C)*-smooth for some* $p = q + s$, $q \in \mathbb{N}_0$ *and* $s \in (0,1]$, *and* $C > 0$. *Let* $A \ge 1$ *and* $M \in \mathbb{N}$ *sufficiently large (independent of the size of* A, *but*

$$M \ge 2 \text{ and } M^{2p} \ge c_{13}\cdot\left(\max\left\{A, \|f\|_{C^q([-A,A]^d)}\right\}\right)^{4(q+1)},$$

where

$$\|f\|_{C^q([-A,A]^d)} = \max_{\substack{\alpha_1,\dots,\alpha_d\in\mathbb{N}_0,\\ \alpha_1+\dots+\alpha_d\le q}} \left\|\frac{\partial^q f}{\partial x_1^{\alpha_1}\dots\partial x_d^{\alpha_d}}\right\|_{\infty,[-A,A]^d},$$

must hold for some sufficiently large constant $c_{13} \ge 1$*).*
Let $L, r \in \mathbb{N}$ *be such that*

1. $L \geq 5M^d + \left\lceil \log_4 \left(M^{2p+4\cdot d\cdot(q+1)} \cdot e^{4\cdot(q+1)\cdot(M^d-1)} \right) \right\rceil$
$\cdot \lceil \log_2(\max\{q,d\}+1) \rceil + \lceil \log_4(M^{2p}) \rceil$

2. $r \geq 132 \cdot 2^d \cdot \lceil e^d \rceil \cdot \binom{d+q}{d} \cdot \max\{q+1, d^2\}$

hold. There exists a feedforward neural network $f_{net,deep}$ with ReLU activation function, L hidden layers and r neurons per hidden layer where all weights are bounded in absolute value by

$$e^{c_{14}\cdot(p+1)\cdot M^d}$$

for some $c_{14} = c_{14}(f) > 0$, such that

$$\|f - f_{net,deep}\|_{\infty,[-A,A]^d} \leq c_{15} \cdot \left(\max\left\{ A, \|f\|_{C^q([-A,A]^d)} \right\} \right)^{4(q+1)} \cdot M^{-2p}. \tag{24}$$

holds.

Proof. This theorem is proven without the upper bound on the absolute values of the weights in Theorem 2 in Kohler and Langer (2021). It is explained in the supplement how the above upper bound on the absolute value of the weights follows from the proof given there. □

Lemma 6 *Let $d_1, d_2, l \in \mathbb{N}$ with $2^l \leq \min\{d_1, d_2\}$. For $k \in \{1, \ldots, l\}$ and $s \in \{1, \ldots, 4^{l-k}\}$ let*

$$\bar{g}_{net,k,s} : \mathbb{R}^4 \to \mathbb{R}$$

be defined by a feedforward neural network with $L_{net} \in \mathbb{N}$ hidden layers and $r_{net} \in \mathbb{N}$ neurons per hidden layer and ReLU activation function, where all the weights are bounded in absolute value by some $B_n \geq 1$. Set

$$I = \left\{0, \ldots, 2^l - 1\right\} \times \left\{0, \ldots, 2^l - 1\right\}$$

and define $\bar{m} : [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to \mathbb{R}$ by

$$\bar{m}(x) = \max_{(i,j)\in\mathbb{Z}^2 \,:\, (i,j)+I \subseteq \{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \bar{f}\left(x_{(i,j)+I}\right),$$

where $\bar{f}$ satisfies

$$\bar{f} = \bar{f}_{l,1}$$

for some $\bar{f}_{k,s} : [-2,2]^{\{1,\ldots,2^k\}\times\{1,\ldots,2^k\}} \to \mathbb{R}$ recursively defined by

$$\begin{aligned} \bar{f}_{k,s}(x) \;=\; & \bar{g}_{net,k,s}\big(\bar{f}_{k-1,4\cdot(s-1)+1}(x_{\{1,\ldots,2^{k-1}\}\times\{1,\ldots,2^{k-1}\}}), \\ & \bar{f}_{k-1,4\cdot(s-1)+2}(x_{\{2^{k-1}+1,\ldots,2^k\}\times\{1,\ldots,2^{k-1}\}}), \\ & \bar{f}_{k-1,4\cdot(s-1)+3}(x_{\{1,\ldots,2^{k-1}\}\times\{2^{k-1}+1,\ldots,2^k\}}), \\ & \bar{f}_{k-1,4\cdot s}(x_{\{2^{k-1}+1,\ldots,2^k\}\times\{2^{k-1}+1,\ldots,2^k\}})\big) \end{aligned}$$

for $k = 2, \ldots, l, s = 1, \ldots, 4^{l-k}$, *and*

$$\bar{f}_{1,s}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}) = \bar{g}_{net,1,s}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2})$$

for $s = 1, \ldots, 4^{l-1}$. *Set*

$$l_{net} = \frac{4^l - 1}{3} \cdot L_{net} + l,$$

$$k_s = \frac{2 \cdot 4^l + 4}{3} + r_{net} \quad (s = 1, \ldots, l_{net}),$$

and set

$$M_s = 2^{\pi(s)} \quad \text{for } s \in \{1, \ldots, l_{net}\},$$

where the function $\pi : \{1, \ldots, l_{net}\} \to \{1, \ldots, l\}$ *is defined by*

$$\pi(s) = \sum_{i=1}^{l} \mathbb{I}_{\{s \geq i + \sum_{r=l-i+1}^{l-1} 4^r \cdot L_{net}\}}.$$

Then there exists some $m_{net} \in \mathcal{F}^{CNN}_{l_{net},\mathbf{k},\mathbf{M}}$, *where all the weights are bounded in absolute value by* B_n, *such that*

$$\bar{m}(x) = m_{net}(x)$$

holds for all $x \in [-2, 2]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}}$.

Proof. This theorem is proven without the upper bound on the absolute values of the weights in Lemma 2 in Kohler, Krzyżak and Walter (2022). In its proof the weights from the feedforward neural networks are copied in the convolutional neural network, and all other weights occuring are bounded in absolute value by 1, therefore the bound on the absolute values of the weights holds. □

Lemma 7 *Set*

$$f(z) = \begin{cases} \infty & , z = 1 \\ \log \frac{z}{1-z} & , 0 < z < 1 \\ -\infty & , z = 0, \end{cases}$$

let $K \in \mathbb{N}$ *with* $K \geq 6$, *let* $m : \mathbb{R}^{d \cdot l} \to [0, 1]$ *and let* $\bar{g} : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}$ *such that* $\|\bar{g} - m\|_{\infty,[0,1]^{d_1 \times d_2}} \leq \epsilon$ *for some*

$$0 \leq \epsilon \leq \frac{1}{K}.$$

Then there exists a neural network $\bar{f} : \mathbb{R} \to \mathbb{R}$ *with ReLU activation function, and one hidden layer with* $3 \cdot K + 9$ *neurons, where all the weights are bounded in absolute value by* K, *such that for each network* $\tilde{f} : \mathbb{R} \to \mathbb{R}$ *which has the same structure and which has weights which are in supremum norm not more than*

$$0 \leq \bar{\epsilon} \leq 1$$

away from the weights of the above network we have that $\tilde{f} \circ \bar{g}$ *satisfies*

$$\|\tilde{f} \circ \bar{g}\|_{\infty,[0,1]^{d_1 \times d_2}} \leq 132 \cdot K^2 \cdot \bar{\epsilon} + \log K$$

and

$$\begin{aligned}
&\sup_{x \in [0,1]^{d_1 \times d_2}} \left(\left| m(x) \cdot \left(\varphi(\tilde{f}(\bar{g}(x))) - \varphi(f(m(x))) \right) \right| \right. \\
&\qquad \left. + \left| (1 - m(x)) \cdot \left(\varphi(-\tilde{f}(\bar{g}(x))) - \varphi(-f(m(x))) \right) \right| \right) \\
&\leq c_{16} \cdot \left(\frac{\log K}{K} + \epsilon \right) + 132 \cdot K^2 \cdot \bar{\epsilon}.
\end{aligned}$$

Proof. See Lemma 13 in Kohler and Krzyżak (2023). □

Lemma 8 *Let* $p \geq 1$ *and* $C > 0$ *be arbitrary. Assume that* $\eta : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}$ *satisfies a* (p, C)*–smooth hierarchical max-pooling model. Let* $\mathcal{F}_n$ *be the set of all CNNs with ReLU activation function, which have* $L_n^{(1)}$ *convolutional layers with* c_7 *neurons in each layer, where* c_7 *is sufficiently large, one max pooling layer and one additional layer with* $L_n^{(2)}$ *neurons. Furthermore assume* $(L_n^{(1)})^{2p/4} \geq c_{17} \cdot L_n^{(2)}$*. Then there exists a network* $f \in \mathcal{F}_n$ *where all the weights are bounded in absolute value by*

$$\max \left\{ L_n^{(2)}, e^{c_{18}(\eta) \cdot (p+1) \cdot L_n^{(1)}} \right\}$$

such that

$$\mathbf{E} \left\{ \varphi(Y \cdot f(X)) \right\} - \mathbf{E} \left\{ \varphi(Y \cdot f_\varphi^*(X)) \right\} \leq c_{19} \cdot \left(\frac{\log L_n^{(2)}}{L_n^{(2)}} + \frac{1}{(L_n^{(1)})^{2p/4}} \right)$$

and such that f *is bounded in absolute value by* $\log L_n^{(2)}$*.*

Proof. Use Lemma 5 (applied with $d = 4$ and

$$M = \left\lfloor \left(\frac{3}{4^l - 1} \cdot (L_n^{(1)} - l) \right)^{\frac{1}{4}} \right\rfloor,$$

where l is the level of the hierarchical max–pooling model for η) and Lemma 6 to construct a convolutional neural network $\bar{g}_{NN}$ built on the basis of feedforward neural networks which approximate the functions in the hierarchical model of the a posteriori probability η in supremum norm up to an error of order

$$\frac{1}{(L_n^{(1)})^{2p/4}}. \tag{25}$$

Application of Lemma 4 with $t = 1$ and $g(x) = \bar{g}(x) = x$ yields that $\bar{g}_{NN}$ approximates η in supremum norm by an error of order (25). Next apply Lemma 7 (with $\epsilon = c_{20} \cdot \frac{1}{(L_n^{(1)})^{2p/4}}$ and $\bar{\epsilon} = 0$) to construct a neural network $\tilde{f}$ with one hidden layer and $L_n^{(2)}$ neurons which takes on function values bounded in absolute value by $\log L_n^{(2)}$ and which satisfies

$$\begin{aligned}
&\sup_{x \in [0,1]^{d_1 \times d_2}} \Bigg(\Big| \eta(x) \cdot \Big(\varphi(\tilde{f}(\bar{g}_{NN}(x))) - \varphi(f(\eta(x))) \Big) \Big| \\
&\qquad\qquad + \Big| (1 - \eta(x)) \cdot \Big(\varphi(-\tilde{f}(\bar{g}_{NN}(x))) - \varphi(-f(\eta(x))) \Big) \Big| \Bigg) \\
&\leq c_{21} \cdot \left(\frac{\log L_n^{(2)}}{L_n^{(2)}} + \frac{1}{(L_n^{(1)})^{2p/4}} \right),
\end{aligned}$$

where f is the function defined in Lemma 7. Because of

$$f_\varphi^*(x) = f(\eta(x))$$

this implies

$$\begin{aligned}
&\mathbf{E}\left\{ \varphi(Y \cdot \tilde{f}(\bar{g}_{NN}(X))) \right\} - \mathbf{E}\left\{ \varphi(Y \cdot f_\varphi^*(X)) \right\} \\
&= \mathbf{E}\left\{ (1_{\{Y=1\}} + 1_{\{Y=-1\}}) \cdot \Big(\varphi(Y \cdot \tilde{f}(\bar{g}_{NN}(X))) - \varphi(Y \cdot f(\eta(X))) \Big) \right\} \\
&= \mathbf{E}\Bigg\{ \eta(X) \cdot \Big(\varphi(\tilde{f}(\bar{g}_{NN}(X))) - \varphi(f(\eta(X))) \Big) \\
&\qquad + (1 - \eta(X)) \cdot \Big(\varphi(-\tilde{f}(\bar{g}_{NN}(X))) - \varphi(-f(\eta(X))) \Big) \Bigg\} \\
&\leq \sup_{x \in [0,1]^{d_1 \times d_2}} \Bigg(\Big| \eta(x) \cdot \Big(\varphi(\tilde{f}(\bar{g}_{NN}(x))) - \varphi(f(\eta(x))) \Big) \Big| \\
&\qquad\qquad + \Big| (1 - \eta(x)) \cdot \Big(\varphi(-\tilde{f}(\bar{g}_{NN}(x))) - \varphi(-f(\eta(x))) \Big) \Big| \Bigg) \\
&\leq c_{21} \cdot \left(\frac{\log L_n^{(2)}}{L_n^{(2)}} + \frac{1}{(L_n^{(1)})^{2p/4}} \right).
\end{aligned}$$

□

4.2.4. Generalization error

Lemma 9 *Let $\{f_\vartheta : \vartheta \in \bar{\mathbf{\Theta}}\}$ be defined as in Subsection 3.2 and let $\beta_n = c_1 \cdot \log n$. Then we have*

$$\sup_{x_1, \ldots, x_n \in [0,1]^{d_1 \times d_2}} \mathbf{E}\left\{ \left| \sup_{\vartheta \in \bar{\mathbf{\Theta}}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \cdot T_{\beta_n} f_\vartheta(x_i) \right| \right\}$$

$$\leq c_{22} \cdot (\log n)^2 \cdot \frac{\sqrt{((L_n^{(1)})^2 + L_n^{(1)} \cdot L_n^{(2)}) \cdot \log(\max\{L_n^{(1)}, L_n^{(2)}\})}}{\sqrt{n}}.$$

In the proof of Lemma 8 we will need the following bound on the VC dimension of the class $\{f_\vartheta : \vartheta \in \bar{\mathbf{\Theta}}\}$ of functions.

Lemma 10 *The VC dimension of the class $\{f_\vartheta : \vartheta \in \bar{\mathbf{\Theta}}\}$ of functions in Lemma 9 is bounded from above by*

$$c_{23} \cdot ((L_n^{(1)})^2 + L_n^{(1)} \cdot L_n^{(2)}) \cdot \log(\max\{L_n^{(1)}, L_n^{(2)}\}).$$

Proof. The result follows from the proof of Lemma 7 in Kohler, Krzyżak and Walter (2022). □

Proof of Lemma 9. The results follows from Lemma 10 by an easy application of standard techniques from VC theory. For the sake of completeness we present nevertheless a detailed proof here.

Set $\mathcal{F} = \{f_\vartheta : \vartheta \in \bar{\Theta}\}$. For $\delta_n > 0$ and $x_1, \ldots, x_n \in [0,1]^{d_1 \times d_2}$ we have

$$\begin{aligned}
&\mathbf{E}\left\{\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i \cdot T_{\beta_n}(f(x_i)) \right| \right\} \\
&= \int_0^{\beta_n} \mathbf{P}\left\{\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i \cdot T_{\beta_n}(f(x_i)) \right| > t \right\} dt \\
&\leq \delta_n + \int_{\delta_n}^{\beta_n} \mathbf{P}\left\{\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i \cdot T_{\beta_n}(f(x_i)) \right| > t \right\} dt.
\end{aligned}$$

Using a standard covering argument from empirical process theory we see that for any $t \geq \delta_n$ we have

$$\begin{aligned}
&\mathbf{P}\left\{\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i \cdot T_{\beta_n}(f(x_i)) \right| > t \right\} \\
&\leq \mathcal{M}_1\left(\frac{\delta_n}{2}, \{T_{\beta_n} f : f \in \mathcal{F}\}, x_1^n\right) \\
&\qquad \cdot \sup_{f \in \mathcal{F}} \mathbf{P}\left\{\left| \frac{1}{n} \sum_{i=1}^n \epsilon_i \cdot T_{\beta_n}(f(x_i)) \right| > \frac{t}{2} \right\}.
\end{aligned}$$

Application of Lemma 10 and Theorem 9.4 in Györfi et al. (2002) yields

$$\mathcal{M}_1\left(\frac{\delta_n}{2}, \{T_{\beta_n} f : f \in \mathcal{F}\}, x_1^n\right) \leq c_{24} \cdot \left(\frac{c_{25} \cdot \beta_n}{\delta_n}\right)^{c_{26} \cdot ((L_n^{(1)})^2 + L_n^{(1)} \cdot L_n^{(2)}) \cdot \log(\max\{L_n^{(1)}, L_n^{(2)}\})}.$$

By the inequality of Hoeffding (cf., e.g., Lemma A.3 in Györfi et al. (2002)) and

$$|T_{\beta_n}(f(x))| \leq \beta_n \quad (x \in \mathbb{R}^d)$$

we have for any $f \in \mathcal{F}$

$$\mathbf{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i \cdot T_{\beta_n}(f(x_i))\right| > t\right\} \leq 2 \cdot \exp\left(-\frac{2 \cdot n \cdot t^2}{4 \cdot \beta_n^2}\right).$$

Hence we get

$$\begin{aligned}
&\sup_{x_1,\ldots,x_n \in [0,1]^{d_1 \times d_2}} \mathbf{E}\left\{\sup_{f \in \mathcal{G} \circ \mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i \cdot T_{\beta_n}(f(X_i))\right|\right\} \\
&\leq \delta_n + \int_{\delta_n}^{\beta_n} c_{24} \cdot \left(\frac{c_{25} \cdot \beta_n}{\delta_n}\right)^{c_{26} \cdot ((L_n^{(1)})^2 + L_n^{(1)} \cdot L_n^{(2)}) \cdot \log(\max\{L_n^{(1)}, L_n^{(2)}\})} \\
&\qquad\qquad \cdot 2 \cdot \exp\left(-\frac{n \cdot \delta_n \cdot t}{2 \cdot \beta_n^2}\right) dt \\
&\leq \delta_n + c_{24} \cdot \left(\frac{c_{25} \cdot \beta_n}{\delta_n}\right)^{c_{26} \cdot ((L_n^{(1)})^2 + L_n^{(1)} \cdot L_n^{(2)}) \cdot \log(\max\{L_n^{(1)}, L_n^{(2)}\})} \frac{4 \cdot \beta_n^2}{n \cdot \delta_n} \cdot \exp\left(-\frac{n \cdot \delta_n^2}{2 \cdot \beta_n^2}\right).
\end{aligned}$$

With

$$\delta_n = \sqrt{((L_n^{(1)})^2 + L_n^{(1)} \cdot L_n^{(2)}) \cdot \log(\max\{L_n^{(1)}, L_n^{(2)}\})} \cdot \log n \cdot \sqrt{\frac{2 \cdot \beta_n^2}{n}}$$

we get the assertion. □

4.2.5. A bound on the gradient

Lemma 11 *Let A, $\bar{\Theta}$ and $f_{(\mathbf{w},\vartheta)}$ be defined as in Section 3, set*

$$M_{\max} = \max\{M_1, \ldots, M_{L_n^{(1)}}\} \quad \textit{and} \quad k_{max} = \max\{k_1, \ldots, k_{L_n^{(1)}}\}$$

and assume that all weights in $f_{(\mathbf{w},\vartheta)}$ are bounded in absolute value by $B_n \geq 1$. Then

$$\begin{aligned}
&\sup_{\mathbf{w} \in A, \vartheta \in \bar{\Theta}^{K_n}, y \in \{-1,1\}, x \in [0,1]^{d_1 \times d_2}} \|\nabla_\vartheta \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty \\
&\leq L_n^{(2)} \cdot k_{max}^{2 \cdot L_n^{(1)}+1} \cdot (M_{max}^2 + 1)^{2 \cdot L_n^{(1)}+2} \cdot B_n^{2 \cdot L_n^{(1)}+2}.
\end{aligned}$$

Proof. Let

$$f_{(\mathbf{w},\vartheta)}(x) = \sum_{k=1}^{K_n} w_k \cdot T_{\beta_n} f_{\vartheta_k}(x)$$

where

$$f_{\vartheta_k}(x) = g_{\mathbf{w}_k}(f_{\mathbf{w}_k}(x)),$$

$$g_{\mathbf{w}_k}(z) = \sum_{i=1}^{L_n^{(2)}} (\mathbf{w}_k)_i^{(1)} \sigma\left((\mathbf{w}_k)_{i,1}^{(0)} \cdot z + (\mathbf{w}_k)_{i,0}^{(0)}\right) + (\mathbf{w}_k)_0^{(1)},$$

$$f_{\mathbf{w}_k}(x) \quad = \quad \max\left\{ \sum_{s_2=1}^{k_{L_n^{(1)}}} (\mathbf{w}_k)_{s_2} \cdot (o_{(\mathbf{w}_k)})_{(i,j),s_2}^{(L_n^{(1)})} : i \in \{1,\dots,d_1 - M_{L_n^{(1)}} + 1\} \right.$$
$$\left. , j \in \{1,\dots,d_2 - M_{L_n^{(1)}} + 1\} \right\},$$

$$(o_{(\mathbf{w}_k)})_{(i,j),s_2}^{(l)} = \sigma\left(\sum_{s_1=1}^{k_{l-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_l\} \\ (i+t_1-1,j+t_2-1)\in D}} (\mathbf{w}_k)_{t_1,t_2,s_1,s_2}^{(l)} (o_{(\mathbf{w}_k)})_{(i+t_1-1,j+t_2-1),s_1}^{(l-1)} + (\mathbf{w}_k)_{s_2}^{(l)} \right)$$

for $l \in \{1,\dots,L_n^{(1)}\}$, and

$$(o_{(\mathbf{w}_k)})_{(i,j),1}^{(0)} = x_{i,j} \quad \text{for } i \in \{1,\dots,d_1\} \text{ and } j \in \{1,\dots,d_2\}.$$

By the proof of Lemma 3 we know for any $l \in \{1,\dots,L_n^{(1)}\}$

$$|(o_{(\mathbf{w}_k)})_{(i,j),s_2}^{(l)}(x)| \leq k_{max}^{L_n^{(1)}} \cdot (M_{max}^2+1)^{L_n^{(1)}} \cdot B_n^{L_n^{(1)}}. \tag{26}$$

Using the chain rule we get for any $l \in \{1,\dots,L_n^{(1)}\}$ and suitably chosen (random) $(i_0,j_0) \in D$

$$\begin{aligned}
&\frac{\partial}{\partial (\mathbf{w}_k)_{\tilde{t}_1,\tilde{t}_2,\tilde{s}_1,\tilde{s}_2}^{(l)}} f_{(\mathbf{w},\vartheta)}(x) \\
&= w_k \cdot \frac{\partial}{\partial z} T_{\beta_n} z\Big|_{z=f_{\vartheta_k}(x)} \cdot \sum_{i=1}^{L_n^{(2)}} (\mathbf{w}_k)_i^{(1)} \sigma'\left((\mathbf{w}_k)_{i,1}^{(0)} \cdot f_{\mathbf{w}_k}(x) + (\mathbf{w}_k)_{i,0}^{(0)} \right) \cdot (\mathbf{w}_k)_{i,1}^{(0)} \cdot \sum_{s_2=1}^{k_{L_n^{(1)}}} (\mathbf{w}_k)_{s_2} \\
&\quad \cdot \sigma'\left(\sum_{s_1=1}^{k_{L_n^{(1)}-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_{L_n^{(1)}}\} \\ (i_0+t_1-1,j_0+t_2-1)\in D}} (\mathbf{w}_k)_{t_1,t_2,s_1,s_2}^{(L_n^{(1)})} (o_{(\mathbf{w}_k)})_{(i_0+t_1-1,j_0+t_2-1),s_1}^{(L_n^{(1)}-1)} + (\mathbf{w}_k)_{s_2}^{(L_n^{(1)})} \right) \\
&\quad \cdot \sum_{s_1^{(L_n^{(1)})}=1}^{k_{L_n^{(1)}-1}} \sum_{\substack{t_1^{(L_n^{(1)})},t_2^{(L_n^{(1)})} \in \{1,\dots,M_{L_n^{(1)}}\} \\ (i_0+t_1^{(L_n^{(1)})}-1,j_0+t_2^{(L_n^{(1)})}-1)\in D}} (\mathbf{w}_k)_{t_1^{(L_n^{(1)})},t_2^{(L_n^{(1)})},s_1^{(L_n^{(1)})},s_2}^{(L_n^{(1)})} \\
&\quad \cdot \sigma'\left(\sum_{s_1=1}^{k_{L_n^{(1)}-2}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_{L_n^{(1)}-1}\} \\ (i_0+t_1^{(L_n^{(1)})}+t_1-2,j_0+t_2^{(L_n^{(1)})}+t_2-2)\in D}} (\mathbf{w}_k)_{t_1,t_2,s_1,s_1^{(L_n^{(1)})}}^{(L_n^{(1)}-1)} \cdot \right.
\end{aligned}$$

$$\left. \cdot \sum_{s_1^{(L_n^{(1)}-1)}=1}^{k_{L_n^{(1)}-2}} \sum_{\substack{t_1^{(L_n^{(1)}-1)},t_2^{(L_n^{(1)}-1)} \in \{1,\ldots,M_{L_n^{(1)}-1}\} \\ (i_0+t_1^{(L_n^{(1)})}+t_1^{(L_n^{(1)}-1)}-2, j_0+t_2^{(L_n^{(1)})}+t_2^{(L_n^{(1)}-1)}-2) \in D}} (\mathbf{w}_k)^{(L_n^{(1)}-1)}_{t_1^{(L_n^{(1)}-1)},t_2^{(L_n^{(1)}-1)},s_1^{(L_n^{(1)}-1)},s_1^{(L_n^{(1)})}} \right.$$

$$\cdot \ldots \cdot \sigma \left(\sum_{s_1=1}^{k_{l-1}} \sum_{\substack{t_1,t_2 \in \{1,\ldots,M_l\}, (i_0+t_1^{(L_n^{(1)})}+\cdots+t_1^{(l+1)}+\tilde{t}_1+t_1-(L_n-(l-2)), \\ j_0+t_2^{(L_n^{(1)})}+\cdots+t_2^{(l+1)}+\tilde{t}_2+t_2-(L_n^{(1)}-(l-2))) \in D}} (\mathbf{w}_k)^{(l-1)}_{t_1,t_2,s_1,\tilde{s}_1} \cdot (o_{(\mathbf{w}_k)})^{(l-2)}_{\substack{(i_0+t_1^{(L_n^{(1)})}+\cdots+t_1^{(l+1)}+\tilde{t}_1+t_1-(L_n^{(1)}-(l-2)), \\ j_0+t_2^{(L_n^{(1)})}+\cdots+t_2^{(l+1)}+\tilde{t}_2+t_2-(L_n^{(1)}-(l-2))), s_1}} + (\mathbf{w}_k)^{(l-1)}_{\tilde{s}_1} \right).$$

Using (26), $|\sigma'(z)| \leq 1$ and that all weights are bounded in the absolute value by B_n we get

$$\left| \frac{\partial}{\partial (\mathbf{w}_k)^{(l)}_{t_1,t_2,s_1,s_2}} f_{(\mathbf{w},\vartheta)}(x) \right| \leq L_n^{(2)} \cdot k_{max}^{2 \cdot L_n^{(1)}+1} \cdot (M_{max}^2+1)^{2 \cdot L_n^{(1)}+2} \cdot B_n^{2 \cdot L_n^{(1)}+2}.$$

Analogously we can derive bounds on all the other partial derivatives occuring in the assertion. □

4.2.6. Proof of Theorem 2

It suffices to show

$$\mathbf{E}\left\{\varphi(Y \cdot f_n(X))\right\} - \min_{f:[0,1]^{d_1 \times d_2} \to \bar{\mathbb{R}}} \mathbf{E}\left\{\varphi(Y \cdot f(X))\right\} \leq c_{28} \cdot (\log n)^4 \cdot n^{-\min\{\frac{p}{2p+4}, \frac{1}{4}\}} \quad (27)$$

for n sufficiently large.

This implies the assertion, because by Lemma 2 a) we conclude from (27)

$$\begin{aligned}
&\mathbf{P}\left\{Y \neq \hat{C}_n(X)\right\} - \mathbf{P}\left\{Y \neq f^*(X)\right\} \\
&\leq \mathbf{E}\left\{ \frac{1}{\sqrt{2}} \cdot \left(\mathbf{E}\left\{\varphi(Y \cdot f_n(X)) | \mathcal{D}_n\right\} - \mathbf{E}\left\{\varphi(Y \cdot f_{\varphi^*}(X))\right\}\right)^{1/2} \right\} \\
&\leq \frac{1}{\sqrt{2}} \cdot \sqrt{\mathbf{E}\left\{\varphi(Y \cdot f_n(X))\right\} - \mathbf{E}\left\{\varphi(Y \cdot f_{\varphi^*}(X))\right\}} \\
&\leq c_8 \cdot (\log n)^2 \cdot n^{-\min\{\frac{p}{2 \cdot (2p+4)}, \frac{1}{8}\}}
\end{aligned}$$

And from Lemma 2 b), (18) and Lemma 2 c) we conclude from (27)

$$\mathbf{P}\left\{Y \neq \hat{C}_n(X)\right\} - \mathbf{P}\left\{Y \neq f^*(X)\right\}$$

$$\leq 2 \cdot \left(\mathbf{E}\left\{ \varphi(Y \cdot f_n(X)) \right\} - \mathbf{E}\left\{ \varphi(Y \cdot f_{\varphi^*}(X)) \right\} \right) + 4 \cdot \frac{c_{29} \cdot \log n}{n^{1/4}}$$
$$\leq c_9 \cdot (\log n)^4 \cdot n^{-\min\{\frac{p}{2p+4}, \frac{1}{4}\}}.$$

Here we have used the fact that

$$\max\left\{ \frac{\mathbf{P}\{Y=1|X\}}{1-\mathbf{P}\{Y=1|X\}}, \frac{1-\mathbf{P}\{Y=1|X\}}{\mathbf{P}\{Y=1|X\}} \right\} > n^{1/4}$$

is equivalent to

$$|f_{\varphi^*}(X)| = \left| \log \frac{\mathbf{P}\{Y=1|X\}}{1-\mathbf{P}\{Y=1|X\}} \right| > \frac{1}{4} \cdot \log n.$$

In the remainder of the proof we apply Theorem 1 in order to prove (27). Here we assume throughout the proof that n is sufficiently large. Let f_ϑ be the network of Lemma 8 which satisfies

$$\mathbf{E}\left\{ \varphi(Y \cdot f_\vartheta(X)) \right\} - \mathbf{E}\left\{ \varphi(Y \cdot f^*_\varphi(X)) \right\} \leq c_{30} \cdot \left(\frac{\log L_n^{(2)}}{L_n^{(2)}} + \frac{1}{(L_n^{(1)})^{2p/4}} \right).$$

Since f_ϑ is bounded in supremum norm by $\log L_n^{(2)} \leq \beta_n$ (for c_3 sufficiently large) this implies

$$\mathbf{E}\left\{ \varphi(Y \cdot T_{\beta_n} f_\vartheta(X)) \right\} - \mathbf{E}\left\{ \varphi(Y \cdot f^*_\varphi(X)) \right\} \leq c_{30} \cdot \left(\frac{\log L_n^{(2)}}{L_n^{(2)}} + \frac{1}{(L_n^{(1)})^{2p/4}} \right).$$

Set

$$\delta_n = \frac{1}{n \cdot e^n}$$

and choose

$$\Theta^* = \left\{ \vartheta^* \in \Theta : \|\vartheta^* - \vartheta\|_\infty \leq \delta_n \right\}.$$

Then it follows from the Lipschitz continuity of the logistic loss and Lemma 3 that we have for any $\vartheta^* \in \Theta^*$

$$|\varphi(Y \cdot T_{\beta_n} f_{\vartheta^*}(X)) - \varphi(Y \cdot T_{\beta_n} f_\vartheta(X))| \leq c_{31} \cdot e^n \cdot \|\vartheta^* - \vartheta\|_\infty \leq \frac{c_{32}}{n},$$

from which we can conclude

$$\sup_{\vartheta^* \in \Theta^*} \mathbf{E}\left\{ \varphi(Y \cdot T_{\beta_n} f_{\vartheta^*}(X)) \right\} - \mathbf{E}\left\{ \varphi(Y \cdot f^*_\varphi(X)) \right\} \leq c_{30} \cdot \left(\frac{\log L_n^{(2)}}{L_n^{(2)}} + \frac{1}{(L_n^{(1)})^{2p/4}} \right) + \frac{c_{32}}{n}.$$

Furthermore the definition of Θ^* implies that

$$\epsilon_n = \mathbf{P}\left\{ \vartheta_1^{(0)} \in \mathbf{\Theta}^* \right\} \geq \left(\frac{1}{n \cdot e^n} \right)^{c_{33} \cdot (L_n^{(1)} + L_n^{(2)})} \geq e^{-n^{1.5}}.$$

So if we choose

$$N_n = n^2 \cdot e^{2 \cdot n}, \quad I_n = n^2 \cdot e^{n^{1.5}}$$

(which is possible because of $N_n \cdot I_n \leq K_n$) then we have

$$N_n \cdot (1 - \epsilon_n)^{I_n} \leq \frac{1}{n},$$

so (10) holds.

By Lemma 3 we know that (8) is satisfied for

$$C_n = c_{34} \cdot e^n,$$

and because of

$$\|\nabla_{\mathbf{w}} \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|^2 \leq \sum_{k=1}^{K_n} |1 \cdot T_{\beta_n} f_{\vartheta_k}(x)|^2 \leq K_n \cdot \beta_n^2$$

(11) is satisfied for

$$D_n = \sqrt{K}_n \cdot \beta_n.$$

By Lemma 11 we know

$$\sup_{\mathbf{w} \in A, \vartheta \in \bar{\Theta}^{K_n}, y \in \{-1,1\}, x \in [0,1]^{d_1 \times d_2}} \|\nabla_\vartheta \varphi(y \cdot f_{(\mathbf{w},\vartheta)}(x))\|_\infty \leq e^n.$$

Application of Theorem 1 together with Lemma 9 and the above results yields

$$\begin{aligned}
&\mathbf{E}\left\{\varphi(Y \cdot f_n(X))\right\} - \min_{f:\mathbb{R}^{d_1 \times d_2} \to \bar{\mathbb{R}}} \mathbf{E}\left\{\varphi(Y \cdot f(X))\right\} \\
&\leq c_{35} \cdot \left(\frac{\log n}{n} + (\log n)^3 \frac{\sqrt{\left(n^{\frac{4}{2p+4}} + n^{\frac{2}{2p+4}} \cdot n^{\frac{1}{4}}\right) \cdot \log n}}{\sqrt{n}} + \frac{e^n}{n \cdot e^n} + \frac{K_n \cdot \beta_n^2}{t_n} \right. \\
&\quad \left. + \frac{n \cdot \beta_n \cdot (K_n + e^n \cdot e^n)}{t_n} + \frac{\log L_n^{(2)}}{L_n^{(2)}} + \frac{1}{(L_n^{(1)})^{2p/4}} \right) \\
&\leq c_{28} \cdot (\log n)^4 \cdot n^{-\min\{\frac{p}{2p+4}, \frac{1}{4}\}}.
\end{aligned}$$

□

5. Acknowledgment

The second author would like to thank Natural Sciences and Engineering Research Council of Canada for funding this project under Grant RGPIN-2020-06793.

References

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

[2] Andoni, A., Panigraphy, R., Valiant, G., and Zhang, L. (2014). Learning polynomials with neural networks. In *International Conference on Machine Learning*, pp. 1908–1916.

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

[4] 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.

[5] Bartlett, P. L., Montanari, A., and Rakhlin, A. (2021). Deep learning: a statistical viewpoint. *arXiv: 2103.09177v1.*

[6] 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.

[7] Birman, M. S., and Solomjak, M. Z. (1967). Piece-wise polynomial approximations of functions in the classes W_p^α. *Mathematics of the USSR Sbornik* **73**, pp. 295-317.

[8] Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, Second Edition, Springer, pp. 421-436.

[9] Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, **60**, pp. 223-311.

[10] Braun, A., Kohler, M., Langer, S., and Walk, H. (2024). Convergence rates for shallow neural networks learned by gradient descent. *Bernoulli*, **30**, pp. 475-502.

[11] Chen, X., Lee, J.D., Tong, X.T. and Zhang Y. (2020). Statistical Inference For Model Parameters In Stochastic Gradient Descent. *Annals of Statistics*, **48**, pp. 251-273.

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

[13] Cover, T. M. (1968). Rates of convergence of nearest neighbor procedures. In *Proceedings of the Hawaii International Conference on Systems Sciences*, pp. 413-415, Honolulu, HI.

[14] Daniely, A. (2017). SGD learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*, pp. 2422–2430.

[15] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springed, New York, USA.

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

[17] Drews, S. and Kohler, M. (2022). On the universal consistency of an over-parametrized deep neural network estimate learned by gradient descent. arXiv:2208.14283.

[18] 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.

[19] 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*.

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

[21] Gonon, L. (2021). Random feature networks learn Black-Scholes type PDEs without curse of dimensionality. Preprint, *arXiv: 2106.08900*.

[22] Gurevych, I., Kohler, M., and Sahin, G. G. (2022). On the rate of convergence of a classifier based on a Transformer encoder. *IEEE Transactions on Information Theory*, **68**, pp. 8139-8155.

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

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

[25] 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.

[26] 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.

[27] Jacot, A., Gabriel, F., und Hongler, C. (2020). Neural tangent kernel: convergence and generalization in neural networks. *arXiv: 1806.07572v4*.

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

[29] Kim, Y., Ohn, I. and Kim, D. (2021). Fast convergence rates of deep neural networks for classification. *Neural Networks*, **138**, pp.179-197.

[30] Kohler, M. (2014). Optimal global rates of convergence for noiseless regression estimation problems with adaptively chosen design. *Journal of Multivariate Analysis*, **132**, pp. 197-208.

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

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

[33] 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.*

[34] Kohler, M., and Krzyżak, A. (2023). On the rate of convergence of an over-parametrized transformer classifier learned by gradient descent. Preprint, *arXiv: 2312.17007.*

[35] Kohler, M., Krzyżak, A., and Walter, B. (2022). On the rate of convergence of image classifiers based on convolutional neural networks. *Annals of the Institute of Statistical Mathematics*, **74**, pp. 1085-1108.

[36] Kohler, M., Krzyżak, A., and Walter, B. (2023). Analysis of the rate of convergence of an over-parametrized convolutional neural network image classifier learned by gradient descent. *Journal of Statistical Planning and Inference* (to appear).

[37] Kohler, M., and Langer, S. (2020). Statistical theory for image classification using deep convolutional neural networks with cross-entropy loss. arXiv: 2011.13602.

[38] 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.*

[39] Kohler, M., and Walter, B. (2023). Analysis of convolutional neural network image classifiers in a rotationally symmetric model. *IEEE Transaction on Information Theory* **69**, pp. 5203-5218.

[40] 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* **25**, pp. 1097-1105. Red Hook, NY: Curran.

[41] Kushner, J.H. and Yin, G.G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*, Springer.

[42] Langer, S. (2021). Approximating smooth functions by deep neural networks with sigmoid activation function. *Journal of Multivariate Analysis* **182**, pp. 104696.

[43] LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, **521**, pp.436-444.

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

[45] Li, G., Gu, Y. and Ding, J. (2021). The rate of convergence of variation-constrained deep neural networks. arXiv: 2106.12068

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

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

[48] 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.

[49] Nemirovski, A., Juditsky, A., Lan, G. and Shapiro, A. (2008). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, **19**, pp. 1574-1609.

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

[51] Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal of Control and Optimization*, **30**, pp. 838-855.

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

[53] 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.

[54] 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.

[55] Rawat, W., and Wang, Z. (2017). Deep convolutional neural networks for image classification: a comprehensive review. *Neural Computation*, **29**, pp. 2352-2449.

[56] Robbins, H., and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400-407.

[57] Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function (with discussion). *Annals of Statistics* **48**, pp. 1875–1897. Preprint, *arXiv:1708.06633v2.*

[58] Spall, J.C. (2003). *Introduction To Stochastic Search And Optimization: Estimation, Simulation and Control* John Wiley & Sons.

[59] Sun, S., Cao, Z., Zhu, H., and Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE Transactions on Cybernetics*, **50**, pp. 3668-3681.

[60] Suzuki, T. (2018). Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. Preprint, *arXiv: 1810.08033.*

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

[62] Toulis, P. and Airoldi, E.M. (2017). Asymptotic And Finite-Sample Properties Of Estimators Based On Stochastic Gradients. *Annals of Statistics*, **45**, pp. 1694-1727.

[63] Walter, B. (2021). Analysis of convolutional neural network image classifiers in a hierarchical max-pooling model with additional local pooling. arXiv: 2106.05233

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

[65] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin,I. (2017). Attention is all you need. Arxiv 1706.03762.

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

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

[68] Yehudai, G., and Shamir, O. (2022). On the power and limitations of random features for understanding neural networks. Preprint, *arXiv: 1904.00687*

[69] Zhang, T. (2004). Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, **32**, pp. 56 - 134.

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

A. Supplement: Proof of Lemma 5

In the following we prove the weight constraints of Lemma 5 by modifying the auxiliary results of the proof of Kohler and Langer (2021).

A.1. Further notation and definitions

The following auxiliary notation is required for the statement of these resuts:

We introduce our framework for a fully-connected neural network $g : \mathbb{R}^d \to \mathbb{R}$ with ReLU activation function $\sigma(x) = \max\{x, 0\}$: Let

$$g(x) = \sum_{i=1}^{k_L} v_{1,i}^{(L)} g_i^{(L)}(x) + v_{1,0}^{(L)}, \tag{28}$$

where output weights $v_{1,0}^{(L)}, \ldots, v_{1,k_L}^{(L)} \in \mathbb{R}$ denote the output weights of the network. The outputs of the neurons $g_i^{(L)}$ are recursively defined by

$$g_i^{(r)}(x) = \sigma\left(\sum_{j=1}^{k_{r-1}} v_{i,j}^{(r-1)} g_j^{(r-1)}(x) + v_{i,0}^{(r-1)}\right),$$

with inner weights $v_{i,0}^{(r-1)}, \ldots, v_{i,k_{r-1}}^{(r-1)} \in \mathbb{R}$ for $i \in \{1, \ldots, k_r\}$, $r \in \{1, \ldots, L\}$, $k_0 = d$ and

$$g_j^{(0)}(x) = x^{(j)}.$$

A fully-connected neural network of the form (28) is dependent on the number of layers L and a width vector $\mathbf{k} = (k_1, \ldots, k_L)$, hence we will denote the corresponding function class by $\mathcal{F}_{L,\mathbf{k}}^{FNN}$. In case $k_1 = \ldots = k_L = r$, we write $\mathcal{F}_{L,r}^{FNN}$ to indicate that all layers consist of a constant number of r neurons. Further, we denote by $\mathbf{v}_f$ the vector that collects all weights required for the computation of $f \in \mathcal{F}_{L,\mathbf{k}}^{FNN}$:

$$\mathbf{v}_f = \left(\left(v_{i,j}^{(l)}\right)_{i \in \{1,\ldots,k_{l+1}\}, j \in \{0,\ldots,k_l\}, l \in \{0,\ldots,L-1\}}, \left(v_{1,i}^{L}\right)_{i \in \{1,\ldots,k_L\}}\right).$$

The proof is based on an approximation by a piecewise Taylor polynomial, which is defined using a partition into equivolume cubes. If C is a cube, then C_{left} is used to denote the "bottom left" of C. We can thus write each half-open cube C with side length s as a polytope defined by

$$-x^{(j)} + C_{left}^{(j)} \leq 0 \text{ and } x^{(j)} - C_{left}^{(j)} - s < 0 \quad (j \in \{1, \ldots, d\}).$$

Furthermore, we describe by $C_\delta^0 \subset C$ the cube, which contains all $x \in C$ that lie with a distance of at least δ to the boundaries of C, i.e. C_δ^0 is the polytope defined by

$$-x^{(j)} + C_{left}^{(j)} \leq -\delta \text{ and } x^{(j)} - C_{left}^{(j)} - s < -\delta \quad (j \in \{1, \ldots, d\}).$$

If $\mathcal{P}$ is a partition of cubes of $[-a,a)^d$ and $x \in [-a,a)^d$, then we denote the cube $C \in \mathcal{P}$, which satisfies $x \in C$, by $C_{\mathcal{P}}(x)$.

Let $\mathcal{P}_N$ be the linear span of all monomials of the form

$$\prod_{k=1}^{d} \left(x^{(k)}\right)^{r_k}$$

for some $r_1, \ldots, r_d \in \mathbb{N}_0$, $r_1 + \cdots + r_d \leq N$. Then, $\mathcal{P}_N$ is a linear vector space of functions of dimension

$$dim\ \mathcal{P}_N = \left|\left\{(r_0, \ldots, r_d) \in \mathbb{N}_0^{d+1} : r_0 + \cdots + r_d = N\right\}\right| = \binom{d+N}{d}.$$

A.2. Auxiliary results

Lemma 12 *Let $p = q + s$ for some $q \in \mathbb{N}_0$ and $s \in (0,1]$, and let $C > 0$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a (p,C)-smooth function, let $x_0 \in \mathbb{R}^d$ and let T_{f,q,x_0} be the Taylor polynomial of total degree q around x_0 defined by*

$$T_{f,q,x_0}(x) \quad = \quad \sum_{j \in \mathbb{N}_0^d : \|\mathbf{j}\|_1 \leq q} (\partial^{\mathbf{j}} f)(x_0) \cdot \frac{(x - x_0)^{\mathbf{j}}}{\mathbf{j}!}$$

Then for any $x \in \mathbb{R}^d$

$$|f(x) - T_{f,q,x_0}(x)| \leq c_{35} \cdot C \cdot \|x - x_0\|^p$$

holds for a constant $c_{35} = c_{35}(q,d)$ depending only on q and d.

Proof. See Lemma 1 in Kohler (2014). □

In the proof of Lemma 5 we use Lemma 12 and approximate our function by a piecewise Taylor polynomial. To define this piecewise Taylor polynomial, we partition $[-a,a)^d$ into M^d and M^{2d} half-open equivolume cubes of the form

$$[\boldsymbol{\alpha}, \boldsymbol{\beta}) = [\boldsymbol{\alpha}^{(1)}, \boldsymbol{\beta}^{(1)}) \times \cdots \times [\boldsymbol{\alpha}^{(d)}, \boldsymbol{\beta}^{(d)}), \quad \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^d,$$

respectively. Let

$$\mathcal{P}_1 = \{C_{k,1}\}_{k \in \{1,\ldots,M^d\}} \text{ and } \mathcal{P}_2 = \{C_{j,2}\}_{j \in \{1,\ldots,M^{2d}\}} \tag{29}$$

be the corresponding partitions. We denote for each $i \in \{1, \ldots, M^d\}$ those cubes of $\mathcal{P}_2$ that are contained in $C_{i,1}$ by $\tilde{C}_{1,i}, \ldots, \tilde{C}_{M^d,i}$ and order the cubes in such a way that the bottom left of $\tilde{C}_{1,i}$ and $C_{i,1}$ coincide, i.e. such that we have $(\tilde{C}_{1,i})_{left} = (C_{i,1})_{left}$ and that

$$(\tilde{C}_{k,i})_{left} = (\tilde{C}_{k-1,i})_{left} + \tilde{\mathbf{v}}_k \tag{30}$$

holds for all $k \in \{2, \dots, M^d\}, i \in \{1, \dots, M^d\}$ and some vector $\tilde{\mathbf{v}}_k$ with entries in $\{0, 2a/M^2\}$ where exactly one entry is different to zero. Here the vector $\tilde{\mathbf{v}}_k$ describes the position of $(C_{k,i})_{left}$ relative to $(C_{k-1,i})_{left}$ and we order the cubes in such a way that the position is independent of i. Then Taylor expansion in Lemma 12 can be used to define a piecewise Taylor polynomial on $\mathcal{P}_2$ by

$$T_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x) = \sum_{k \in \{1,\dots,M^d\}, i \in \{1,\dots,M^d\}} T_{f,q,(\tilde{C}_{k,i})_{left}}(x) \cdot \mathbb{1}_{\tilde{C}_{k,i}}(x)$$

and this piecewise Taylor polynomial satisfies

$$\sup_{x \in [-a,a)^d} \left| f(x) - T_{f,q,((C_{\mathcal{P}_2}(x))_{left}}(x) \right| \leq c_{35} \cdot C \cdot (2 \cdot a \cdot d)^p \cdot \frac{1}{M^{2p}}.$$

To compute $T_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)$ the very deep neural network of Lemma 5 b) proceeds in two steps: In a first step it computes $(C_{\mathcal{P}_1}(x))_{left}$ and the values of

$$(\partial^{\mathbf{l}} f)((C_{i,1})_{left})$$

for each $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$ and suitably defined numbers

$$b_{k,i}^{(\mathbf{l})} \in \mathbb{Z}, \quad |b_{k,i}^{(\mathbf{l})}| \leq e^d + 1 \quad (k \in \{1, \dots, M^d\}),$$

which depend on $C_{i,1}$ for $i \in \{1, \dots, M^d\}$. Assume that $x \in C_{i,1}$ for some $i \in \{1, \dots, M^d\}$. In the second step the neural network successively computes approximations

$$(\partial^{\mathbf{l}} \hat{f})((\tilde{C}_{k,i})_{left}), \quad k \in \{1, \dots, M^d\}$$

of

$$(\partial^{\mathbf{l}} f)((\tilde{C}_{k,i})_{left})$$

for each $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$. To do this we start with

$$(\partial^{\mathbf{l}} \hat{f})((\tilde{C}_{1,i})_{left}) = (\partial^{\mathbf{l}} f)((C_{\mathcal{P}_1}(x))_{left}).$$

By construction of the first step and since $(\tilde{C}_{1,i})_{left} = (C_{\mathcal{P}_1}(x))_{left}$ these estimates have error zero. As soon as we have computed the above estimates for some $k \in \{1, \dots, M^d - 1\}$ we use the Taylor polynomials with these coefficients around $(\tilde{C}_{k,i})_{left}$ in order to compute

$$\sum_{\substack{\mathbf{j} \in \mathbb{N}_0^d: \\ \|\mathbf{j}\|_1 \leq q - \|\mathbf{l}\|_1}} \frac{(\partial^{\mathbf{l}+\mathbf{j}} \hat{f})((\tilde{C}_{k,i})_{left})}{\mathbf{j}!} \cdot \left((\tilde{C}_{k+1,i})_{left} - (\tilde{C}_{k,i})_{left} \right)^{\mathbf{j}}$$

for $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$ and we define

$$(\partial^{\mathbf{l}} \hat{f})((\tilde{C}_{k+1,i})_{left}) = \sum_{\substack{\mathbf{j} \in \mathbb{N}_0^d: \\ \|\mathbf{j}\|_1 \leq q - \|\mathbf{l}\|_1}} \frac{(\partial^{\mathbf{l}+\mathbf{j}} \hat{f})((\tilde{C}_{k,i})_{left})}{\mathbf{j}!} \cdot \left((\tilde{C}_{k+1,i})_{left} - (\tilde{C}_{k,i})_{left} \right)^{\mathbf{j}}$$

$$+ b_{k,i}^{(\mathbf{l})} \cdot c_{36} \cdot \left(\frac{2a}{M^2}\right)^{p-\|\mathbf{l}\|_1}$$

where

$$c_{36} = C \cdot d^p \cdot \max\{c_{35}(q,d), c_{35}(q-1,d), \ldots, c_{35}(0,d)\}$$

(and c_{35} is the constant of Lemma 12). Assume that

$$\left|(\partial^{\mathbf{l}} \hat{f})((\tilde{C}_{k,i})_{left}) - (\partial^{\mathbf{l}} f)((\tilde{C}_{k,i})_{left})\right| \leq c_{36} \cdot \left(\frac{2a}{M^2}\right)^{p-\|\mathbf{l}\|_1}$$

holds for all $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$ (which holds by construction for $k = 1$). Then

$$\begin{aligned}
&\Bigg| \sum_{\substack{\mathbf{j}\in\mathbb{N}_0^d:\\ \|\mathbf{j}\|_1 \leq q-\|\mathbf{l}\|_1}} \frac{(\partial^{\mathbf{l}+\mathbf{j}} \hat{f})\left((\tilde{C}_{k,i})_{left}\right)}{\mathbf{j}!} \cdot \left((\tilde{C}_{k+1,i})_{left} - (\tilde{C}_{k,i})_{left}\right)^{\mathbf{j}} \\
&\quad - (\partial^{\mathbf{l}} f)\left((C_{k+1,i})_{left}\right) \Bigg| \\
\leq &\Bigg| \sum_{\substack{\mathbf{j}\in\mathbb{N}_0^d:\\ \|\mathbf{j}\|_1 \leq q-\|\mathbf{l}\|_1}} \frac{(\partial^{\mathbf{l}+\mathbf{j}} \hat{f})((\tilde{C}_{k,i})_{left})}{\mathbf{j}!} \cdot \left((\tilde{C}_{k+1,i})_{left} - (\tilde{C}_{k,i})_{left}\right)^{\mathbf{j}} \\
&\quad - \sum_{\substack{\mathbf{j}\in\mathbb{N}_0^d:\\ \|\mathbf{j}\|_1 \leq q-\|\mathbf{l}\|_1}} \frac{(\partial^{\mathbf{l}+\mathbf{j}} f)((\tilde{C}_{k,i})_{left})}{\mathbf{j}!} \cdot \left((\tilde{C}_{k+1,i})_{left} - (\tilde{C}_{k,i})_{left}\right)^{\mathbf{j}} \Bigg| \\
&\quad + \Bigg| \sum_{\substack{\mathbf{j}\in\mathbb{N}_0^d:\\ \|\mathbf{j}\|_1 \leq q-\|\mathbf{l}\|_1}} \frac{(\partial^{\mathbf{l}+\mathbf{j}} f)(\tilde{C}_{k,i})_{left})}{\mathbf{j}!} \cdot \left((\tilde{C}_{k+1,i})_{left} - (\tilde{C}_{k,i})_{left}\right)^{\mathbf{j}} \\
&\quad - (\partial^{\mathbf{l}} f)((\tilde{C}_{k+1,i})_{left}) \Bigg| \\
\leq &\sum_{\substack{\mathbf{j}\in\mathbb{N}_0^d:\\ \|\mathbf{j}\|_1 \leq q-\|\mathbf{l}\|_1}} \frac{1}{\mathbf{j}!} \cdot c_{36} \cdot \left(\frac{2a}{M^2}\right)^{p-\|\mathbf{l}+\mathbf{j}\|_1} \cdot \left(\frac{2a}{M^2}\right)^{\|\mathbf{j}\|_1} + c_{36} \cdot \left(\frac{2a}{M^2}\right)^{p-\|\mathbf{l}\|_1} \\
\leq &(c_{36} \cdot e^d + c_{36}) \cdot \left(\frac{2a}{M^2}\right)^{p-\|\mathbf{l}\|_1}.
\end{aligned}$$

This implies that we can choose $b_{k,i}^{(\mathbf{l})} \in \mathbb{Z}$ such that

$$|b_{k,i}^{(\mathbf{l})}| \leq e^d + 1$$

and

$$\left|(\partial^{\mathbf{l}}\hat{f})((\tilde{C}_{k+1,i})_{left}) - (\partial^{\mathbf{l}}f)((\tilde{C}_{k+1,i})_{left})\right| \leq c_{36} \cdot \left(\frac{2a}{M^2}\right)^{p-\|\mathbf{l}\|_1}.$$

Observe that in this way we have defined the coefficients $b_{k,i}^{(\mathbf{l})}$ for each cube $C_{i,1}$. We will encode these coefficients for each $i \in \{1, \ldots, M^d\}$ and each $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$ in the single number

$$b_i^{(\mathbf{l})} = \sum_{k=1}^{M^d-1} \left(b_{k,i}^{(\mathbf{l})} + \lceil e^d \rceil + 2\right) \cdot (4 + 2\lceil e^d \rceil)^{-k} \in [0,1].$$

In the last step the neural network then computes

$$\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x) := \sum_{\substack{\mathbf{l}\in\mathbb{N}_0^d:\\ \|\mathbf{l}\|_1\leq q}} \frac{(\partial^{\mathbf{l}}\hat{f})((C_{\mathcal{P}_2}(x))_{left})}{\mathbf{l}!} \cdot (x - (C_{\mathcal{P}_2}(x))_{left})^{\mathbf{l}}, \tag{31}$$

where by construction we have $C_{\mathcal{P}_2}(x) = \tilde{C}_{k,i}$ for some $k \in \{1, \ldots, M^d\}$. Since

$$\begin{aligned}
&\left|\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x) - T_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)\right| \\
&\leq \sum_{\substack{\mathbf{l}\in\mathbb{N}_0^d:\\ \|\mathbf{l}\|_1\leq q}} \frac{\left|(\partial^{\mathbf{l}}\hat{f} - \partial^{\mathbf{l}}f)((C_{\mathcal{P}_2}(x))_{left})\right|}{\mathbf{l}!} \cdot |x - (C_{\mathcal{P}_2}(x))_{left}|^{\mathbf{l}} \\
&\leq e^d \cdot c_{36} \cdot \left(\frac{2a}{M^2}\right)^p
\end{aligned} \tag{32}$$

the network approximating $\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)$ is also a good approximation for $T_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)$.

To approximate $f(x)$ by neural networks the proof of Kohler and Langer follows *four* key steps:

1. Compute $\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)$ by recursively defined functions.

2. Approximate the recursive functions by neural networks. The resulting network is a good approximation for $f(x)$ in case that

$$x \in \bigcup_{k\in\{1,\ldots,M^{2d}\}} (C_{k,2})_{1/M^{2p+2}}^0.$$

3. Approximate the function $w_{\mathcal{P}_2}(x) \cdot f(x)$ by deep neural networks, where

$$w_{\mathcal{P}_2}(x) = \prod_{j=1}^{d} \left(1 - \frac{M^2}{a} \cdot \left|(C_{\mathcal{P}_2}(x))_{left}^{(j)} + \frac{a}{M^2} - x^{(j)}\right|\right)_+ \tag{33}$$

is a linear tensor product B-spline which takes its maximum value at the center of $C_{\mathcal{P}_2}(x)$, which is nonzero in the inner part of $C_{\mathcal{P}_2}(x)$ and which vanishes outside of $C_{\mathcal{P}_2}(x)$.

4. Apply those networks to 2^d slightly shifted partitions of $\mathcal{P}_2$ to approximate $f(x)$ in supremum norm.

We focus on step 2 and 3 and modify the construction of the auxiliary neural networks by deriving constraints for the required weights.

A.2.1. Key step 1: A recursive definition of $\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)$

To derive a recursive definition of $\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)$, we set

$$\boldsymbol{\phi}_{1,0} = \left(\phi_{1,0}^{(1)}, \dots, \phi_{1,0}^{(d)}\right) = x$$
$$\boldsymbol{\phi}_{2,0} = \left(\phi_{2,0}^{(1)}, \dots, \phi_{2,0}^{(d)}\right) = \mathbf{0}$$

and

$$\phi_{3,0}^{(\mathbf{l})} = 0 \text{ and } \phi_{4,0}^{(\mathbf{l})} = 0$$

for each $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$. For $j \in \{1, \dots, M^d\}$ set

$$\boldsymbol{\phi}_{1,j} = \boldsymbol{\phi}_{1,j-1},$$

$$\boldsymbol{\phi}_{2,j} = (C_{j,1})_{left} \cdot \mathbb{1}_{C_{j,1}}(\boldsymbol{\phi}_{1,j-1}) + \boldsymbol{\phi}_{2,j-1},$$

$$\phi_{3,j}^{(\mathbf{l})} = (\partial^{\mathbf{l}} f)((C_{j,1})_{left}) \cdot \mathbb{1}_{C_{j,1}}(\boldsymbol{\phi}_{1,j-1}) + \phi_{3,j-1}^{(\mathbf{l})}$$

and

$$\phi_{4,j}^{(\mathbf{l})} = b_j^{(\mathbf{l})} \cdot \mathbb{1}_{C_{j,1}}(\boldsymbol{\phi}_{1,j-1}) + \phi_{4,j-1}^{(\mathbf{l})}.$$

Furthermore set

$$\boldsymbol{\phi}_{1,M^d+j} = \boldsymbol{\phi}_{1,M^d+j-1}, \quad j \in \{1, \dots, M^d\},$$

$$\boldsymbol{\phi}_{2,M^d+j} = \boldsymbol{\phi}_{2,M^d+j-1} + \tilde{\mathbf{v}}_{j+1},$$

$$\phi_{3,M^d+j}^{(\mathbf{l})} = \sum_{\substack{\mathbf{s} \in \mathbb{N}_0^d \\ \|\mathbf{s}\|_1 \leq q - \|\mathbf{l}\|_1}} \frac{\phi_{3,M^d+j-1}^{(\mathbf{l}+\mathbf{s})}}{\mathbf{s}!} \cdot (\tilde{\mathbf{v}}_{j+1})^{\mathbf{s}}$$

$$+\left(\lfloor(4+2\cdot\lceil e^d\rceil)\cdot\phi^{(\mathbf{l})}_{4,M^d+j-1}\rfloor-\lceil e^d\rceil-2\right)\cdot c_{36}\cdot\left(\frac{2a}{M^2}\right)^{p-\|\mathbf{l}\|_1},$$

$$\phi^{(\mathbf{l})}_{4,M^d+j}=(4+2\cdot\lceil e^d\rceil)\cdot\phi^{(\mathbf{l})}_{4,M^d+j-1}-\lfloor(4+2\cdot\lceil e^d\rceil)\cdot\phi^{(\mathbf{l})}_{4,M^d+j-1}\rfloor$$

for $j\in\{1,\dots,M^d-1\}$ and each $\mathbf{l}\in\mathbb{N}_0^d$ with $\|\mathbf{l}\|_1\leq q$ and

$$\boldsymbol{\phi}_{5,M^d+j}=\mathbb{1}_{\mathcal{A}^{(j)}}(\boldsymbol{\phi}_{1,M^d+j-1})\cdot\boldsymbol{\phi}_{2,M^d+j-1}+\boldsymbol{\phi}_{5,M^d+j-1}$$

and

$$\phi^{(\mathbf{l})}_{6,M^d+j}=\mathbb{1}_{\mathcal{A}^{(j)}}(\phi_{1,M^d+j-1})\cdot\phi^{(\mathbf{l})}_{3,M^d+j-1}+\phi^{(\mathbf{l})}_{6,M^d+j-1}$$

for $j\in\{1,\dots,M^d\}$, where

$$\boldsymbol{\phi}_{5,M^d}=\left(\phi^{(1)}_{5,M^d},\dots,\phi^{(d)}_{5,M^d}\right)=\mathbf{0},\ \phi^{(\mathbf{l})}_{6,M^d}=0$$

and

$$\begin{aligned}\mathcal{A}^{(j)}=\Big\{&x\in\mathbb{R}^d:-x^{(k)}+\phi^{(k)}_{2,M^d+j-1}\leq 0\\&\text{und } x^{(k)}-\phi^{(k)}_{2,M^d+j-1}-\frac{2a}{M^2}<0 \text{ for all } k\in\{1,\dots,d\}\Big\}.\end{aligned}$$

Finally define

$$\boldsymbol{\phi}_{1,2M^d+1}=\sum_{\substack{\mathbf{l}\in\mathbb{N}_0^d:\\\|\mathbf{l}\|_1\leq q}}\frac{\phi^{(\mathbf{l})}_{6,2M^d}}{\mathbf{l}!}\cdot\left(\boldsymbol{\phi}_{1,2M^d}-\boldsymbol{\phi}_{5,2M^d}\right)^{\mathbf{l}}.$$

The next lemma shows that this recursion computes $\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x)$.

Lemma 13 *Let $p=q+s$ for some $q\in\mathbb{N}_0$ and $s\in(0,1]$, let $C>0$ and $x\in[-a,a)^d$. Let $f:\mathbb{R}^d\to\mathbb{R}$ be a (p,C)-smooth function and let $\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}$ be defined as in (31). Define $\boldsymbol{\phi}_{1,2M^d+1}$ recursively as above. Then we have*

$$\phi_{1,2M^d+1}=\hat{T}_{f,q,(C_{\mathcal{P}_2}(x))_{left}}(x).$$

Proof. See Lemma 11 in Kohler and Langer (2021). □

A.2.2. Key step 2: Approximating $\phi_{1,2M^d+1}$ by neural networks

In this step we show that a neural network approximates $\phi_{1,2M^d+1}$ in case that

$$x\in\bigcup_{i\in\{1,\dots,M^{2d}\}}(C_{i,2})^0_{1/M^{2p+2}}.$$

We define a composition neural network, which approximately computes the recursive functions in the definition of $\phi_{1,2M^d+1}$.

Lemma 14 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Let $\mathcal{P}_2$ be defined as in (29). Let $p = q + s$ for some $q \in \mathbb{N}_0$ and $s \in (0, 1]$, and let $C > 0$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a (p, C)-smooth function. Let $1 \leq a < \infty$. Then there exists for $M \in \mathbb{N}$ sufficiently large (independent of the size of a, but*

$$\begin{aligned} M^{2p} \quad \geq \quad & 2^{4(q+1)+1} \max\{c_{37} \cdot (6 + 2\lceil e^d \rceil)^{4(q+1)}, c_{36} \cdot e^d\} \\ & \cdot \left(\max\left\{ a, \|f\|_{C^q([-a,a]^d)} \right\} \right)^{4(q+1)} \end{aligned} \tag{34}$$

must hold), a neural network $\hat{f}_{deep,\mathcal{P}_2} \in \mathcal{F}(L, r)$ with

(i) $$\begin{aligned} L = 4M^d + \left\lceil \log_4 \left(M^{2p+4 \cdot d \cdot (q+1)} \cdot e^{4 \cdot (q+1) \cdot (M^d - 1)} \right) \right\rceil \\ \cdot \lceil \log_2(\max\{q + 1, 2\}) \rceil \end{aligned}$$

(ii) $$\begin{aligned} r = \max \Big\{ & 10d + 4d^2 + 2 \cdot \tbinom{d+q}{d} \cdot \left(2 \cdot (4 + 2\lceil e^d \rceil) + 5 + 2d \right), \\ & 18 \cdot (q + 1) \cdot \tbinom{d+q}{d} \Big\} \end{aligned}$$

such that

$$|\hat{f}_{deep,\mathcal{P}_2}(x) - f(x)| \leq c_{38} \cdot \left(\max\left\{ 2a, \|f\|_{C^q([-a,a]^d)} \right\} \right)^{4(q+1)} \cdot \frac{1}{M^{2p}}$$

holds for all $x \in \bigcup_{i \in \{1,\dots,M^{2d}\}} (C_{i,2})^0_{1/M^{2p+2}}$. The network value is bounded by

$$\begin{aligned} |\hat{f}_{deep,\mathcal{P}_2}(x)| \leq 1 + \Bigg(& \|f\|_{C^q([-a,a]^d)} \cdot e^{(M^d - 1)} \\ & + (4 + 2 \cdot \lceil e^d \rceil) \cdot (M^d - 1) \cdot e^{(M^d - 2)} \Bigg) \cdot e^{2ad} \end{aligned}$$

for all $x \in [-a, a)^d$.
$\hat{f}_{deep,\mathcal{P}_2}$ satisfies the weight constraint:

$$\left\| \mathbf{v}_{\hat{f}_{deep,\mathcal{P}_2}} \right\|_\infty \leq e^{c_{39} \cdot (M^d + d) \cdot 2(q+1)},$$

where $c_{39} = c_{39}(f)$.

As in Kohler and Langer (2021), auxiliary networks are required to prove these results. We introduce the auxiliary networks with weight constraints and modify parts of the proofs accordingly.

In the construction of our network we will compose smaller subnetworks to successively build the final network. Here instead of using an additional layer, we "merge" the weights of both networks f and g to define $f \circ g$. The following lemma clarifies this idea and derives appropriate weight constraints:

Lemma 15 *Let $f_0 : \mathbb{R}^k \to \mathbb{R}$ be a neural network of the class $\mathcal{F}(L, r)$ with weight vector $\mathbf{v}_0$ and let $f_1, \ldots, f_k : \mathbb{R}^d \to \mathbb{R}$ be neural networks of class $\mathcal{F}(\bar{L}, \bar{r})$ with weight vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$. Denote by $\bar{\mathbf{v}}$ the vector that contains $(\mathbf{v}_j)_{j\in\{1,\ldots,k\}}$.*
Then the network $f = f_0(f_1, \ldots, f_k)$ has $L + \bar{L}$ layers and at most $\max\{k \cdot \bar{r}, r\}$ neurons.

a) In general $\mathbf{v}$ satisfies the constraints

$$\|\mathbf{v}\|_\infty \leq \max\left\{\|\mathbf{v}_0\|_\infty, \|\bar{\mathbf{v}}\|_\infty, \left\|(\mathbf{v}_0)^{(0)}\right\|_\infty \cdot \left(k \left\|(\bar{\mathbf{v}})^{(\bar{L})}\right\|_\infty + 1\right)\right\}.$$

b) If $(\mathbf{v}_j)_{1,0}^{(\bar{L})} = 0$ for all $j \in \{1, \ldots, k\}$, $\mathbf{v}$ satisfies :

$$\|\mathbf{v}\|_\infty \leq \max\left\{\|\mathbf{v}_0\|_\infty, \|\bar{\mathbf{v}}\|_\infty, \left\|(\mathbf{v}_0)_{i,j>0}^{(0)}\right\|_\infty \cdot \left\|(\bar{\mathbf{v}})_{1,j>0}^{(\bar{L})}\right\|_\infty\right\}.$$

c) If additionally to $(\mathbf{v}_j)_{1,0}^{(\bar{L})} = 0$ for all $j \in \{1, \ldots, k\}$, we have $\left\|(\mathbf{v}_0)_{i,j>0}^{(0)}\right\|_\infty \leq 1$ or $\left\|(\bar{\mathbf{v}})_{i,j>0}^{(\bar{L})}\right\|_\infty \leq 1$, then $\mathbf{v}$ satisfies

$$\|\mathbf{v}\|_\infty \leq \max\left\{\|\mathbf{v}_0\|_\infty, \|\bar{\mathbf{v}}\|_\infty\right\}.$$

Proof. The network $f = f_0(f_1, \ldots, f_k)$ is recursively defined as follows

$$f(x) = \sum_{i=1}^{r} (\mathbf{v}_0)_{1,i}^{(L)} \, f_i^{(\bar{L}+L)}(x) + (\mathbf{v}_0)_{1,0}^{(L)},$$

where for $l \in \{2, \ldots, L\}$ the outputs of the neurons $f_i^{(\bar{L}+l)}$ are recursively defined by

$$f_i^{(\bar{L}+l)}(x) = \sigma\left(\sum_{j=1}^{r} (\mathbf{v}_0)_{i,j}^{(l-1)} \, f_j^{(\bar{L}+l-1)}(x) + (\mathbf{v}_0)_{i,0}^{(l-1)}\right)$$

for $i \in \{1, \ldots, r\}$. In layer $\bar{L}$, the effect of "merging" the networks together becomes apparent and we can see that layer $\bar{L}$ the network $f = f_0(f_1, \ldots, f_k)$ consists of $k \cdot \bar{r}$ neurons:

$$\begin{aligned} f_i^{(\bar{L}+1)}(x) &= \sigma\left(\sum_{j=1}^{k} (\mathbf{v}_0)_{i,j}^{(0)} f_j(x) + (\mathbf{v})_{i,0}^{(0)}\right) \\ &= \sigma\left(\sum_{j=1}^{k} (\mathbf{v}_0)_{i,j}^{(0)} \cdot \left(\sum_{l=1}^{\bar{r}} (\mathbf{v}_j)_{1,l}^{(\bar{L})} \cdot f_{j,l}^{(\bar{L})}(x) + (\mathbf{v}_j)_{1,0}^{(\bar{L})}\right) + (\mathbf{v}_0)_{i,0}^{(0)}\right) \qquad (35) \\ &= \sigma\left(\sum_{j=1}^{k} \sum_{l=1}^{\bar{r}} (\mathbf{v}_0)_{i,j}^{(0)} \cdot (\mathbf{v}_j)_{1,l}^{(\bar{L})} \cdot f_{j,l}^{(\bar{L})}(x) + \sum_{j=1}^{k} (\mathbf{v}_0)_{i,j}^{(0)} \cdot (\mathbf{v}_j)_{1,0}^{(\bar{L})} + (\mathbf{v}_0)_{i,0}^{(0)}\right) \end{aligned}$$

for $i \in \{1, \dots, r\}$, where the $f_{j,i}^{(s)}(x)$ are defined by

$$f_{(j-1)\cdot i+i}^{(s)}(x) = f_{j,i}^{(s)}(x) = \sigma\left(\sum_{l=1}^{\bar{r}} (\mathbf{v}_j)_{i,l}^{(s-1)} f_{j,l}^{(s-1)}(x) + (\mathbf{v}_j)_{i,0}^{(s-1)}\right)$$

for $j \in \{1, \dots, k\}, s \in \{1, \dots, \bar{L}\}$ and $i \in \{1, \dots, \bar{r}\}$. Finally we have

$$f_{(j-1)\cdot l+l}^{(s)}(x) = f_{j,l}^{(0)}(x) = x^{(l)}.$$

for $j \in \{1, \dots, k\}, l \in \{1, \dots, d\}$.

In layers $l \in \{1, \dots \bar{L}-1\}$, the weights of f satisfy the same constraints as $f_1, \dots, f_k$, in layers $l \in \{\bar{L}+1, \dots, L+\bar{L}\}$ the weight constraints of f correspond to the constraints of f_0. However, in layer $\bar{L}$ we have to consider the product of the output weights of f_i for $i \in \{1, \dots, k\}$ and the weights of the input layer of f_0, as shown in (35). The weights there satisfy

$$|(\mathbf{v}_0)_{i,j}^{(0)} \cdot (\mathbf{v}_j)_{1,l}^{(\bar{L})}| \leq \max_{j\in\{1,\dots,k\}, l\in\{1,\dots,r\}} \left|(\mathbf{v}_0)_{i,j}^{(0)}\right| \cdot \left|(\mathbf{v}_j)_{1,l}^{(\bar{L})}\right| \leq \left\|(\mathbf{v}_0)^{(0)}\right\|_\infty \cdot \left\|(\bar{\mathbf{v}})^{(\bar{L})}\right\|_\infty$$

and

$$\left|\sum_{j=1}^{k} (\mathbf{v}_0)_{i,j}^{(0)} \cdot (\mathbf{v}_j)_{1,0}^{(\bar{L})} + (\mathbf{v}_0)_{i,0}^{(0)}\right| \leq \left\|(\mathbf{v}_0)^{(0)}\right\|_\infty \cdot \left(\sum_{j=1}^{k} \left\|(\mathbf{v}_j)^{(\bar{L})}\right\|_\infty + 1\right),$$

which implies part a) of the assertion.
If, additionally, $(\mathbf{v}_j)_{1,0}^{(\bar{L})} = 0$ for all $j \in \{1, \dots, k\}$, the latter bound can be refined to

$$|(\mathbf{v})|_{i,0}^{(\bar{L})} = \left|(\mathbf{v}_0)_{i,0}^{(0)}\right| \leq \left\|(\mathbf{v}_0)^{(0)}\right\|_\infty.$$

Since $\left\|(\mathbf{v}_0)^{(0)}\right\|_\infty \leq \|(\mathbf{v}_0)\|_\infty$, this case no longer influences the upper bound and we can reduce the third argument of the maximum to $\left\|(\mathbf{v}_0)_{i,j>0}^{(0)}\right\|_\infty \cdot \left\|(\bar{\mathbf{v}})_{i,j>0}^{(\bar{L})}\right\|_\infty$, which implies the upper bound of part b) of the assertion.
Part c) follows directly from part b). □

The following lemma presents a neural network, that approximates the square function, which is essential to build neural networks for more complex tasks.

Lemma 16 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Then for any $R \in \mathbb{N}$ and any $a \geq 1$ a neural network*

$$\hat{f}_{sq} \in \mathcal{F}(R, 9)$$

exists with weight constraints

$$\left\|\mathbf{v}_{\hat{f}_{sq}}\right\|_\infty = \left\|(\mathbf{v}_{\hat{f}_{sq}})^{(R)}\right\|_\infty \leq 4 \cdot a^2 \qquad \text{and} \qquad \left\|(\mathbf{v}_{\hat{f}_{sq}})^{(0)}\right\|_\infty \leq 1,$$

such that

$$\left|\hat{f}_{sq}(x) - x^2\right| \leq a^2 \cdot 4^{-R}$$

holds for $x \in [-a, a]$.

Proof. This proof follows as in Kohler and Langer (2021). We only modify the parts required to show the constraints on the weights. In Kohler and Langer (2021) it was shown that linear combinations of the "tooth" function $g : [0,1] \to [0,1]$

$$g(x) = \begin{cases} 2x & , x \leq \frac{1}{2} \\ 2 \cdot (1-x) & , x > \frac{1}{2} \end{cases}$$

and the iterated composition function

$$g_s(x) = \underbrace{g \circ g \circ \cdots \circ g}_{s}(x).$$

can be used to approximate $f(x) = x^2$ for $x \in [0,1]$. Let S_R denote the piecewise linear interpolation of f with $2^R + 1$ uniformly distributed breakpoints.

It can be shown that $S_R(x)$ is given by

$$S_R(x) = x - \sum_{s=1}^{R} \frac{g_s(x)}{2^{2s}}$$

and that it satisfies

$$|S_R(x) - x^2| \leq 2^{-2R-2}$$

for $x \in [0,1]$.

In a *third step of their proof* Kohler and Langer show, that there exists a feedforward neural network that computes $S_R(x)$ for $x \in [0,1]$. In order to derive the weight constraints we include the construction of this network.

The function $g(x)$ can be implemented by the network:

$$\hat{f}_g(x) = 2 \cdot \sigma(x) - 4 \cdot \sigma(x - \frac{1}{2}) + 2 \cdot \sigma(x-1)$$

and the function $g_s(x)$ can be implemented by a network

$$\hat{f}_{g_s} \in \mathcal{F}(s, 3)$$

with

$$\hat{f}_{g_s}(x) = \underbrace{\hat{f}_g(\hat{f}_g(\ldots(\hat{f}_g}_{s}(x))).$$

Thus we have

$$\left\| \mathbf{v}_{\hat{f}_g} \right\|_\infty = \left\| \mathbf{v}_{\hat{f}_{g_s}} \right\|_\infty = 4 \quad \text{for all } s \in \mathbb{N}.$$

Let

$$\hat{f}_{id}(z) = \sigma(z) - \sigma(-z),$$

and define $\hat{f}_{id}^t$ recursively by

$$\hat{f}_{id}^0(z) = z \qquad (z \in \mathbb{R})$$

$$\hat{f}_{id}^{t+1}(z) = \hat{f}_{id}(\hat{f}_{id}^{t}(z)) \qquad\qquad (z \in \mathbb{R}, t \in \mathbb{N}_0),$$

which implies

$$\hat{f}_{id}^{t}(z) = z.$$

It is easy to see that these networks satisfy

$$\left\|\mathbf{v}_{\hat{f}_{id}}\right\|_\infty = \left\|\mathbf{v}_{\hat{f}_{id}^{t}}\right\|_\infty = 1 \quad \text{for all } t \in \mathbb{N}.$$

By combining the networks above we can implement the function $S_R(x)$ by a network

$$\hat{f}_{sq_{[0,1]}} \in \mathcal{F}(R, 7)$$

recursively defined as follows: We set $\hat{f}_{1,0}(x) = \hat{f}_{2,0}(x) = x$ and $\hat{f}_{3,0}(x) = 0$. Then we set

$$\hat{f}_{1,i+1}(x) = \hat{f}_{id}(\hat{f}_{1,i}(x)),$$

$$\hat{f}_{2,i+1}(x) = \frac{\hat{f}_g(\hat{f}_{2,i}(x))}{4}$$

and

$$\hat{f}_{3,i+1}(x) = f_{id}(\hat{f}_{3,i}(x)) - \frac{\hat{f}_g(\hat{f}_{2,i}(x))}{4}$$

for $i \in \{0, 1, \ldots, R-2\}$ and

$$\hat{f}_{sq_{[0,1]}}(x) = \hat{f}_{id}(\hat{f}_{1,R-1}(x)) - \frac{\hat{f}_g(\hat{f}_{2,R-1}(x))}{4} + \hat{f}_{id}(\hat{f}_{3,R-1}(x)).$$

Using the positive homogeneity of the ReLU function, this implies

$$\begin{aligned}\hat{f}_{sq_{[0,1]}}(x) =& \hat{f}_{id}^{R}(x) - \frac{1}{2^{2R}}\hat{f}_{g_R}(x) - \hat{f}_{id}\left(\frac{1}{2^{2(R-1)}}\hat{f}_{g_{R-1}}(x)\right.\\ &\left.-\hat{f}_{id}\left(\frac{1}{2^{2(R-2)}}\hat{f}_{g_{R-2}}(x) - \cdots - \hat{f}_{id}\left(\frac{1}{2^2}\hat{f}_{g_1}(x)\right)\right)\right)\\ =& S_R(x),\end{aligned}$$

hence $\hat{f}_{sq_{[0,1]}}(x)$ satisfies

$$|\hat{f}_{sq_{[0,1]}}(x) - x^2| \leq 2^{-2R-2} \tag{36}$$

for $x \in [0,1]$. By construction of $\hat{f}_{sq_{[0,1]}}$ it is easy to see that its weights satisfy the constraint

$$\left\|\mathbf{v}_{\hat{f}_{sq_{[0,1]}}}\right\|_\infty = \frac{1}{4} \cdot \left\|\mathbf{v}_{\hat{f}_g}\right\|_\infty = 1.$$

In a *last step* $\hat{f}_{sq_{[0,1]}}$ is extended to approximate the function $f(x) = x^2$ on the domain $[-a, a]$. Therefore $f_{tran} : [-a, a] \to [0, 1]$ is defined by

$$f_{tran}(z) = \frac{z}{2a} + \frac{1}{2}$$

to transfers the value of $x \in [-a, a]$ in the interval, where (36) holds. Set

$$\hat{f}_{sq}(x) = 4a^2 \hat{f}_{sq_{[0,1]}}(f_{tran}(x)) - (2a \cdot \hat{f}_{id}^R(x) + a^2)$$

The extension to the domain $[-a, a]$ only increases the weights of the last layer of the network, which results in the constraints

$$\left\|(\mathbf{v}_{\hat{f}_{sq}})\right\|_\infty = \left\|(\mathbf{v}_{\hat{f}_{sq}})^{(R)}\right\|_\infty \leq 4 \cdot a^2 \qquad \text{and} \qquad \left\|(\mathbf{v}_{\hat{f}_{sq_{[0,1]}}})^{(0)}\right\|_\infty = \left\|(\mathbf{v}_{\hat{f}_{sq}})^{(0)}\right\|_\infty \leq 1.$$

Since

$$x^2 = 4a^2 \cdot \left(\frac{x}{2a} + \frac{1}{2}\right)^2 - 2ax - a^2$$

we have

$$\begin{aligned}
&|\hat{f}_{sq}(x) - x^2| \\
=&|4a^2 \hat{f}_{sq_{[0,1]}}(f_{tran}(x)) - (2a \cdot \hat{f}_{id}^R(x) + a^2) - (4a^2 \cdot (f_{tran}(x))^2 - 2ax - a^2)| \\
\leq&4a^2 \cdot |\hat{f}_{sq_{[0,1]}}(f_{tran}(x)) - (f_{tran}(x))^2| + 2a|\hat{f}_{id}^R(x) - x| \\
\leq&4a^2 \cdot 2^{-2R-2} = a^2 \cdot 4^{-R}.
\end{aligned}$$

□

Lemma 17 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Then for any $R \in \mathbb{N}$ and any $a \geq 1$ a neural network*

$$\hat{f}_{mult} \in \mathcal{F}(R, 18)$$

exists, whose weights satisfy

$$\left\|\mathbf{v}_{\hat{f}_{mult}}\right\|_\infty \leq 4 \cdot a^2, \qquad \left(\mathbf{v}_{\hat{f}_{mult}}\right)_{1,0}^{(R)} = 0 \qquad \textit{and}$$

$$\left\|\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(0)}\right\|_\infty \leq 1,$$

such that

$$|\hat{f}_{mult}(x, y) - x \cdot y| \leq 2 \cdot a^2 \cdot 4^{-R}$$

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

Proof. Let

$$\hat{f}_{sq} \in \mathcal{F}(R, 9)$$

be the neural network from Lemma 16, i.e.

$$\hat{f}_{sq}(x) = 16a^2 \hat{f}_{sq_{[0,1]}}\left(\frac{x}{4a} + \frac{1}{2}\right) - (4a \cdot \hat{f}_{id}^R(x) + 4a^2).$$

which satisfies

$$|\hat{f}_{sq}(x) - x^2| \leq 4 \cdot a^2 \cdot 4^{-R}$$

for $x \in [-2a, 2a]$ and with weight constraints $\left\|\mathbf{v}_{\hat{f}_{sq_{[0,1]}}}\right\|_\infty = \left\|\mathbf{v}_{\hat{f}^R_{id}}\right\|_\infty = 1$, and set

$$\begin{aligned}\hat{f}_{mult}(x,y) =& \frac{1}{4} \cdot \left(\hat{f}_{sq}(x+y) - \hat{f}_{sq}(x-y)\right) \\ =& 4a^2 \hat{f}_{sq_{[0,1]}}\left(\frac{x+y}{4a} + \frac{1}{2}\right) - a \cdot \hat{f}^R_{id}(x+y) \\ & - \left(4a^2 \hat{f}_{sq_{[0,1]}}\left(\frac{x-y}{4a} + \frac{1}{2}\right) - a \cdot \hat{f}^R_{id}(x-y)\right).\end{aligned}$$

Note that since $\hat{f}_{sq}(x+y)$ and $\hat{f}_{sq}(x-y)$ have the same offset in the last layer, they cancel out and we have $\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(R)}_{1,0} = 0$. The constraints $\left\|\mathbf{v}_{\hat{f}_{mult}}\right\|_\infty = 4 \cdot a^2$ and $\left\|\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(0)}\right\|_\infty \leq 1$ follow directly from Lemma 16. Since

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

we have

$$\begin{aligned}|\hat{f}_{mult}(x,y) - x \cdot y| &\leq \frac{1}{4} \cdot \left|\hat{f}_{sq}(x+y) - (x+y)^2\right| + \frac{1}{4} \cdot \left|(x-y)^2 - \hat{f}_{sq}(x-y)\right| \\ &\leq \frac{1}{4} \cdot 2 \cdot 4 \cdot a^2 \cdot 4^{-R} \\ &\leq 2 \cdot a^2 \cdot 4^{-R}\end{aligned}$$

for $x, y \in [-a, a]$. □

Lemma 18 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Then for $R \in \mathbb{N}, R \geq \log_4\left(2 \cdot 4^{2 \cdot d} \cdot a^{2 \cdot d}\right)$ and any $a \geq 1$ a neural network*

$$\hat{f}_{mult,d} \in \mathcal{F}(R \cdot \lceil \log_2(d) \rceil, 18d),$$

which satisfies

$$\left\|\mathbf{v}_{\hat{f}_{mult,d}}\right\|_\infty \leq 4 \cdot 4^{2d} \cdot a^{2d}, \quad \left(\mathbf{v}_{\hat{f}_{mult,d}}\right)^{(R \cdot \lceil \log_2(d) \rceil)}_{1,0} = 0 \quad \textit{and}$$

$$\left\|\left(\mathbf{v}_{\hat{f}_{mult,d}}\right)^{(0)}\right\|_\infty \leq 1,$$

exists such that

$$\left|\hat{f}_{mult,d}(x) - \prod_{i=1}^{d} x^{(i)}\right| \leq 4^{4d+1} \cdot a^{4d} \cdot d \cdot 4^{-R}$$

holds for all $x \in [-a, a]^d$.

Proof. We set $q = \lceil \log_2(d) \rceil$. The feedforward neural network $\hat{f}_{mult,d}$ with $L = R \cdot q$ hidden layers and $r = 18d$ neurons in each layer is constructed as follows: Set

$$(z_1, \dots, z_{2^q}) = \left(x^{(1)}, x^{(2)}, \dots, x^{(d)}, \underbrace{1, \dots, 1}_{2^q - d} \right). \tag{37}$$

In the construction of our network we will use the network $\hat{f}_{mult}$ of Lemma 17, which satisfies

$$|\hat{f}_{mult}(x, y) - x \cdot y| \leq 2 \cdot (4^d a^d)^2 \cdot 4^{-R} \tag{38}$$

and

$$\left\| \mathbf{v}_{\hat{f}_{mult}} \right\|_\infty \leq 4 \cdot 4^{2d} a^{2d}, \quad \left(\mathbf{v}_{\hat{f}_{mult}} \right)_{1,0}^{(R)} = 0 \quad \text{and} \quad \left\| \left(\mathbf{v}_{\hat{f}_{mult}} \right)^{(0)} \right\|_\infty \leq 1$$

for $x, y \in [-4^d a^d, 4^d a^d]$. In the first R layers we compute

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

which can be done by R layers of $18 \cdot 2^{q-1} \leq 18 \cdot d$ neurons. E.g., in case in case $z_l = x^{(d)}$ and $z_{l+1} = 1$ we have

$$\hat{f}_{mult}(z_l, z_{l+1}) = \hat{f}_{mult}(x^{(d)}, 1).$$

As a result of the first R layers we get a vector of outputs which has length 2^{q-1}. Next we pair these outputs and apply $\hat{f}_{mult}$ again. This procedure is continued until there is only one output left. Therefore we need $L = Rq$ hidden layers and at most $18d$ neurons in each layer.

By (38) and $R \geq \log_4 \left(2 \cdot 4^{2 \cdot d} \cdot a^{2 \cdot d} \right)$ we get for any $l \in \{1, \dots, d\}$ and any $z_1, z_2 \in [-(4^l - 1) \cdot a^l, (4^l - 1) \cdot a^l]$

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

From this we get successively that all outputs of layer $l \in \{1, \dots, q-1\}$ are contained in the interval $[-(4^{2^l} - 1) \cdot a^{2^l}, (4^{2^l} - 1) \cdot a^{2^l}]$, hence in particular they are contained in the interval $[-4^d a^d, 4^d a^d]$ where inequality (38) does hold.

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

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

and

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

The constraints $\left\| \left(\mathbf{v}_{\hat{f}_{mult,d}} \right)^{(0)} \right\|_\infty \leq 1$ and $\left(\mathbf{v}_{\hat{f}_{mult}} \right)_{1,0}^{(R)} = 0$ follow directly from Lemma 17, since $\hat{f}_{mult,d}$ is a repeated composition of $\hat{f}_{mult}$. Note that our construction of $\hat{f}_{mult}$

satisfies the special case of Lemma 15, i.e. $\left(\mathbf{v}_{\hat{f}_{mult}}\right)_{1,0}^{(R)} = 0$ and further $\left\|\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(0)}\right\|_\infty \leq 1$.

Applying Lemma 15 b) we get for the repeated composition of $\hat{f}_{mult}$:

$$\left\|\mathbf{v}_{\hat{f}_{mult,d}}\right\|_\infty \leq \left\|\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(0)}\right\|_\infty \cdot \left\|\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(R)}\right\|_\infty \leq 4 \cdot 4^{2d} \cdot a^{2d}.$$

The rest of the proof follows analogously to the proof of Lemma 8 in Kohler and Langer (2021).

□

Lemma 19 *Let $m_1, \ldots, m_{\binom{d+N}{d}}$ denote all monomials in $\mathcal{P}_N$ for some $N \in \mathbb{N}$. Let $r_1, \ldots, r_{\binom{d+N}{d}} \in \mathbb{R}$, define*

$$p\left(x, y_1, \ldots, y_{\binom{d+N}{d}}\right) = \sum_{i=1}^{\binom{d+N}{d}} r_i \cdot y_i \cdot m_i(x), \quad x \in [-a,a]^d, y_i \in [-a,a]$$

and set $\bar{r}(p) = \max_{i \in \{1,\ldots,\binom{d+N}{d}\}} |r_i|$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Then for any $a \geq 1$ and

$$R \geq \log_4(2 \cdot 4^{2\cdot(N+1)} \cdot a^{2\cdot(N+1)}) \tag{39}$$

a neural network $\hat{f}_p \in \mathcal{F}(L, r)$ with $L = R \cdot \lceil \log_2(N+1) \rceil$ and $r = 18 \cdot (N+1) \cdot \binom{d+N}{d}$ exists, whose weights satisfy

$$\left\|\left(\mathbf{v}_{\hat{f}_p}\right)^{(0)}\right\|_\infty \leq 1, \quad \left(\mathbf{v}_{\hat{f}_p}\right)_{1,0}^{(R)} = 0 \quad \text{and}$$

$$\left\|\mathbf{v}_{\hat{f}_p}\right\|_\infty \leq 4 \cdot \bar{r}(p) \cdot 4^{2(N+1)} \cdot a^{2(N+1)},$$

such that

$$\left|\hat{f}_p\left(x, y_1, \ldots, y_{\binom{d+N}{d}}\right) - p\left(x, y_1, \ldots, y_{\binom{d+N}{d}}\right)\right| \leq c_{40} \cdot \bar{r}(p) \cdot a^{4(N+1)} \cdot 4^{-R}$$

for all $x \in [-a,a]^d$, $y_1, \ldots, y_{\binom{d+N}{d}} \in [-a,a]$, where c_{40} depends on d and N.

Proof. A neural network $\hat{f}_m$ is constructed in order to approximate

$$y \cdot m(x) = y \cdot \prod_{k=1}^{d} \left(x^{(k)}\right)^{r_k}, \quad x \in [-a,a]^d, y \in [-a,a],$$

where $m \in \mathcal{P}_N$ and $r_1, \ldots, r_d \in \mathbb{N}_0$ with $r_1 + \cdots + r_d \leq N$. Note that Lemma 18 can easily be extended to monomials. We set d by $N+1$ and thus get a network

$$\hat{f}_m \in \mathcal{F}(R \cdot \lceil \log_2(N+1) \rceil, 18 \cdot (N+1))$$

whose weights satisfy

$$\left\|\mathbf{v}_{\hat{f}_m}\right\|_\infty \leq 4 \cdot 4^{2(N+1)} \cdot a^{2(N+1)}.$$

We then set $\hat{f}_p = \sum_{i=1}^{\binom{d+N}{d}} r_i \cdot \hat{f}_{m_i}(x, y_i)$, which increases the weight constraint by a factor $\bar{r}(p)$. The weight constraints $\left\|\left(\mathbf{v}_{\hat{f}_p}\right)^{(0)}\right\|_\infty \leq 1$ and $\left(\mathbf{v}_{\hat{f}_p}\right)_{1,0}^{(R)} = 0$ follow directly from Lemma 18. The rest of the proof follows as in the proof of Lemma 5 of Kohler and Langer (2021). □

Lemma 20 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Let $R \in \mathbb{N}$. Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ with*

$$b^{(i)} - a^{(i)} \geq \frac{2}{R} \text{ for all } i \in \{1, \ldots, d\}$$

and let

$$K_{1/R} = \big\{x \in \mathbb{R}^d : x^{(i)} \notin [a^{(i)}, a^{(i)} + 1/R) \cup (b^{(i)} - 1/R, b^{(i)}) \text{ for all } i \in \{1, \ldots, d\}\big\}.$$

a) Then the network

$$\hat{f}_{ind,[\mathbf{a},\mathbf{b})}(x) = \sigma\Bigg(1 - R \cdot \sum_{i=1}^{d} \Bigg(\sigma\left(a^{(i)} + \frac{1}{R} - x^{(i)}\right) + \sigma\left(x^{(i)} - b^{(i)} + \frac{1}{R}\right)\Bigg)\Bigg)$$

of the class $\mathcal{F}(2, 2d)$ satisfies the weight constraint

$$\left\|\mathbf{v}_{\hat{f}_{ind,[\mathbf{a},\mathbf{b})}}\right\|_\infty \leq \max\left\{\|\mathbf{a}\|_\infty + \frac{1}{R}, \|\mathbf{b}\|_\infty + \frac{1}{R}, R\right\},$$

as well as

$$\left\|\left(\mathbf{v}_{\hat{f}_{ind,[\mathbf{a},\mathbf{b})}}\right)_{i,j>0}^{(0)}\right\|_\infty = 1, \quad \left(\mathbf{v}_{\hat{f}_{ind,[\mathbf{a},\mathbf{b})}}\right)_{1,0}^{(2)} = 0 \quad \text{and} \quad \left\|\left(\mathbf{v}_{\hat{f}_{ind,[\mathbf{a},\mathbf{b})}}\right)_{1,i>0}^{(2)}\right\|_\infty = 1.$$

For $x \in K_{1/R}$ we have

$$\hat{f}_{ind,[\mathbf{a},\mathbf{b})}(x) = \mathbb{1}_{[\mathbf{a},\mathbf{b})}(x)$$

and

$$\left|\hat{f}_{ind,[\mathbf{a},\mathbf{b})}(x) - \mathbb{1}_{[\mathbf{a},\mathbf{b})}(\mathbf{x})\right| \leq 1$$

for $x \in \mathbb{R}^d$.
b) Let $|s| \leq R$. *Then the network*

$$\hat{f}_{test}(x, \mathbf{a}, \mathbf{b}, s) = \sigma\Bigg(\hat{f}_{id}(s) - R^2 \cdot \sum_{i=1}^{d} \Bigg(\sigma\left(a^{(i)} + \frac{1}{R} - x^{(i)} \right) + \sigma\left(x^{(i)} - b^{(i)} + \frac{1}{R} \right) \Bigg) \Bigg) - \sigma\Bigg(- \hat{f}_{id}(s) - R^2 \cdot \sum_{i=1}^{d} \Bigg(\sigma\left(a^{(i)} + \frac{1}{R} - x^{(i)} \right) + \sigma\left(x^{(i)} - b^{(i)} + \frac{1}{R} \right) \Bigg) \Bigg)$$

of the class $\mathcal{F}(2, 2 \cdot (2d+2))$ *satisfies the weight constraint*

$$\left\| \mathbf{v}_{\hat{f}_{test}} \right\|_\infty \leq R^2,$$

as well as

$$\left\| \left(\mathbf{v}_{\hat{f}_{test}} \right)_{i,j>0}^{(0)} \right\|_\infty = 1, \quad \left(\mathbf{v}_{\hat{f}_{test}} \right)_{1,0}^{(2)} = 0 \quad \textit{and} \quad \left\| \left(\mathbf{v}_{\hat{f}_{test}} \right)_{1,i>0}^{(2)} \right\|_\infty = 1.$$

For $x \in K_{1/R}$ $\hat{f}_{test}(x, \mathbf{a}, \mathbf{b}, s)$ *satisfies*

$$\hat{f}_{test}(x, \mathbf{a}, \mathbf{b}, s) = s \cdot \mathbb{1}_{[\mathbf{a}, \mathbf{b})}(x)$$

and

$$\left| \hat{f}_{test}(x, \mathbf{a}, \mathbf{b}, s) - s \cdot \mathbb{1}_{[\mathbf{a}, \mathbf{b})}(x) \right| \leq |s|$$

for $x \in \mathbb{R}^d$.

Proof. The weight constraints can easily be seen in the definition of $\hat{f}_{ind,[\mathbf{a},\mathbf{b})}$ and $\hat{f}_{test}$, the proof of the approximation bounds can be found in the proof of Lemma 6 in Kohler and Langer (2021). □

Lemma 21 *Let* $\sigma : \mathbb{R} \to \mathbb{R}$ *be the ReLU activation function* $\sigma(x) = \max\{x, 0\}$. *Let* $R > 0$, $B \in \mathbb{N}$ *and*

$$\hat{f}_{ind,[j,\infty)}(z) = R \cdot \sigma(z - j) - R \cdot \sigma\left(z - j - \frac{1}{R} \right) \in \mathcal{F}(1, 2)$$

for $j \in \{1, \ldots, B\}$. *Then the neural network*

$$\hat{f}_{trunc}(z) = \sum_{j=1}^{B} \hat{f}_{ind,[j,\infty)}(z) \in \mathcal{F}(1, 2B)$$

satisfies

$$\left\|\mathbf{v}_{\hat{f}_{trunc}}\right\|_\infty \leq \max\left\{R, B+\frac{1}{R}\right\},$$

more specifically the network has no offset in its last layer, i.e. $\left(\mathbf{v}_{\hat{f}_{trunc}}\right)_{1,0}^{(1)} = 0$, *and satisfies* $\left\|\left(\mathbf{v}_{\hat{f}_{trunc}}\right)_{i,j>0}^{(0)}\right\|_\infty = 1$ *and* $\left\|\left(\mathbf{v}_{\hat{f}_{trunc}}\right)_{i,j>0}^{(1)}\right\|_\infty = R$.
Further, $\hat{f}_{trunc}$ *satisfies*

$$\hat{f}_{trunc}(z) = \lfloor z \rfloor$$

for $z \in [0, B+1)$ *and* $\min\{|z-j| : j \in \mathbb{N}\} \geq 1/R$.

Proof. The weight constraints can easily be derived from the definition of $\hat{f}_{trunc}$, the proof of the approximation bounds can be found in the proof of Lemma 13 in Kohler and Langer (2021). □

In the proof of Lemma 14 every function of $\boldsymbol{\phi}_{1,2M^d+1}$ is computed by a neural network. In particular, the indicator functions in $\boldsymbol{\phi}_{2,j}$, $\phi_{3,j}^{(\mathbf{l})}$ and $\phi_{4,j}^{(\mathbf{l})}$ ($j \in \{1,\dots,M^d\}, \mathbf{l} \in \mathbb{N}_0^d, \|\mathbf{l}\|_1 \leq q$) are computed by Lemma 20 a), while we apply the identity network to shift the computed values from the previous step. The functions $\phi_{3,M^d+j}^{(\mathbf{l})}$ and $\phi_{4,M^d+j}^{(\mathbf{l})}$ are then computed according to their definition above Lemma 13, while we again use the identity network to shift values in the next hidden layers. For the functions $\boldsymbol{\phi}_{5,M^d+j}$ and $\phi_{6,M^d+j}^{(\mathbf{l})}$ we use the network of Lemma 20 b) to successively compute $(C_{\mathcal{P}_2}(x))_{left}$ and the derivatives on the cube $C_{\mathcal{P}_2}(x)$. The final Taylor polynomial in $\phi_{1,2M^d+1}$ is then approximated with the help of Lemma 19.

Proof of Lemma 14. In the *first step of the proof* we describe how $\phi_{1,2M^d+1}$ of Lemma 13 can be approximated by neural networks. In the construction we will use the network $\hat{f}_{ind,[\mathbf{a},\mathbf{b})} \in \mathcal{F}(2,2d)$ and the network $\hat{f}_{test} \in \mathcal{F}(2, 2\cdot(2d+2))$ of Lemma 20. Here we set $R = B_M = M^{2p+2}$ in Lemma 20, such that the weights of the network satisfy the constraints

$$\left\|\mathbf{v}_{\hat{f}_{ind}}\right\|_\infty \leq M^{2p+2} \quad \text{and} \quad \left\|\mathbf{v}_{\hat{f}_{test}}\right\|_\infty \leq M^{4p+4}.$$

For some vector $\mathbf{v} \in \mathbb{R}^d$ we set

$$\mathbf{v} \cdot \hat{f}_{ind,[\mathbf{a},\mathbf{b})}(x) = \left(v^{(1)} \cdot \hat{f}_{ind,[\mathbf{a},\mathbf{b})}(x), \dots, v^{(d)} \cdot \hat{f}_{ind,[\mathbf{a},\mathbf{b})}(x)\right).$$

Furthermore we use the networks

$$\hat{f}_{trunc,i} \in \mathcal{F}(1, 2\cdot(4+2\lceil e^d \rceil)), \quad (i \in \{1,\dots,M^d-1\})$$

of Lemma 21. Here we choose

$$R = R_{M,i} = (4+2\lceil e^d \rceil)^{M^d-i-1} \quad \text{and} \quad B = 4+2\lceil e^d \rceil$$

in Lemma 21, which implies

$$\left\|\mathbf{v}_{\hat{f}_{trunc}}\right\|_\infty \leq \max\left\{R, B + \frac{1}{R}\right\} \leq 4 + 2\lceil e^d \rceil + (4 + 2\lceil e^d \rceil)^{M^d}. \tag{40}$$

To compute the final Taylor polynomial we use the network

$$\hat{f}_p \in \mathcal{F}\left(B_{M,p} \cdot \lceil \log_2(\max\{q+1, 2\}) \rceil, 18 \cdot (q+1) \cdot \binom{d+q}{d}\right)$$

from Lemma 19 satisfying

$$\begin{aligned}\left\|\mathbf{v}_{\hat{f}_p}\right\|_\infty \leq 4 \cdot \bar{r}(p) \cdot 4^{2(q+1)} \cdot \Bigg(&2 \cdot \max\left\{\|f\|_{C^q([-a,a]^d)}, a\right\} \cdot e^{(M^d-1)} \\ &+ (4 + 2\lceil e^d \rceil) \cdot (M^d - 1) \cdot e^{(M^d-2)}\Bigg)^{2(q+1)},\end{aligned}$$

and

$$\begin{aligned}&\left|\hat{f}_p\left(\mathbf{z}, y_1, \ldots, y_{\binom{d+q}{q}}\right) - p\left(\mathbf{z}, y_1, \ldots, y_{\binom{d+q}{q}}\right)\right| \\ &\leq c_{41} \cdot (6 + 2\lceil e^d \rceil)^{4(q+1)} \cdot \bar{r}(p) \cdot \left(\max\left\{\|f\|_{C^q([-a,a]^d)}, a\right\}\right)^{4(q+1)} \\ &\quad \cdot M^{d \cdot 4 \cdot (q+1)} \cdot e^{4(q+1)\cdot(M^d-1)} \cdot 4^{-B_{M,p}}\end{aligned} \tag{41}$$

for all $z^{(1)}, \ldots, z^{(d)}, y_1, \ldots, y_{\binom{d+q}{d}}$ contained in

$$\begin{aligned}\Big[&-2 \cdot \max\left\{\|f\|_{C^q([-a,a]^d)}, a\right\} \cdot e^{(M^d-1)} + (4 + 2\lceil e^d \rceil) \cdot (M^d - 1) \cdot e^{(M^d-2)}, \\ &2 \cdot \max\left\{\|f\|_{C^q([-a,a]^d)}, a\right\} \cdot e^{(M^d-1)} + (4 + 2\lceil e^d \rceil) \cdot (M^d - 1) \cdot e^{(M^d-2)}\Big],\end{aligned}$$

where

$$R = B_{M,p} = \left\lceil \log_4\left(M^{2p+4\cdot d\cdot(q+1)} \cdot e^{4\cdot(q+1)\cdot(M^d-1)}\right)\right\rceil.$$

A polynomial of degree zero is treated as a polynomial of degree 1, where we choose $r_i = 0$ for all coefficients greater than zero. Thus we substitute $\log_2(q+1)$ by $\log_2(\max\{q+1, 2\})$ in the definition of L in Lemma 19. To compute $\boldsymbol{\phi}_{1,j}, \boldsymbol{\phi}_{2,j}, \phi_{3,j}^{(\mathbf{l})}$ and $\phi_{4,j}^{(\mathbf{l})}$ for $j \in \{0, \ldots, M^d\}$ and each $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$ we use the networks

$$\begin{aligned}\hat{\boldsymbol{\phi}}_{1,0} &= \left(\hat{\phi}_{1,0}^{(1)}, \ldots, \hat{\phi}_{1,0}^{(d)}\right) = x \\ \hat{\boldsymbol{\phi}}_{2,0} &= \left(\hat{\phi}_{2,0}^{(1)}, \ldots, \hat{\phi}_{2,0}^{(d)}\right) = \mathbf{0}, \\ \hat{\phi}_{3,0}^{(\mathbf{l})} &= 0 \text{ and } \hat{\phi}_{4,0}^{(\mathbf{l})} = 0.\end{aligned}$$

for $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$. For $j \in \{1, \dots, M^d\}$ we set

$$\begin{aligned}
\hat{\boldsymbol{\phi}}_{1,j} &= \hat{f}_{id}^2 \left(\hat{\boldsymbol{\phi}}_{1,j-1} \right), \\
\hat{\boldsymbol{\phi}}_{2,j} &= (C_{j,1})_{left} \cdot \hat{f}_{ind,C_{j,1}}(\hat{\boldsymbol{\phi}}_{1,j-1}) + \hat{f}_{id}^2(\hat{\boldsymbol{\phi}}_{2,j-1}), \\
\hat{\phi}_{3,j}^{(\mathbf{l})} &= (\partial^{\mathbf{l}} f)((C_{j,1})_{left}) \cdot \hat{f}_{ind,C_{j,1}}(\hat{\boldsymbol{\phi}}_{1,j-1}) + \hat{f}_{id}^2(\hat{\phi}_{3,j-1}^{(\mathbf{l})}), \\
\hat{\phi}_{4,j}^{(\mathbf{l})} &= b_j^{(\mathbf{l})} \cdot \hat{f}_{ind,C_{j,1}}(\hat{\boldsymbol{\phi}}_{1,j-1}) + \hat{f}_{id}^2(\hat{\phi}_{4,j-1}^{(\mathbf{l})})
\end{aligned}$$

for $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}\|_1 \leq q$.

It is easy to see that this parallelized network needs $2M^d$ hidden layers and $2d + d \cdot (2d+2) + 2 \cdot \binom{d+q}{d} \cdot (2d+2)$ neurons per layer, where we have used the fact that we have $\binom{d+q}{d}$ different vectors $\mathbf{l} \in \mathbb{N}_0^d$ satisfying $\|\mathbf{l}\|_1 \leq q$.
To compute $\boldsymbol{\phi}_{1,M^d+j}, \boldsymbol{\phi}_{5,M^d+j}$ and $\phi_{6,M^d+j}^{(\mathbf{l})}$ for $j \in \{1, \dots, M^d\}$ and $\boldsymbol{\phi}_{2,M^d+j}, \phi_{3,M^d+j}^{(\mathbf{l})}$ and $\phi_{4,M^d+j}^{(\mathbf{l})}$ for $j \in \{1, \dots, M^d - 1\}$ we use the networks

$$\begin{aligned}
\hat{\boldsymbol{\phi}}_{1,M^d+j} &= \hat{f}_{id}^2 \left(\hat{\boldsymbol{\phi}}_{1,M^d+j-1} \right), \quad j \in \{1, \dots, M^d\} \\
\hat{\boldsymbol{\phi}}_{2,M^d+j} &= \hat{f}_{id}^2 \left(\hat{\boldsymbol{\phi}}_{2,M^d+j-1} + \tilde{\mathbf{v}}_{j+1} \right) \\
\hat{\phi}_{3,M^d+j}^{(\mathbf{l})} &= \hat{f}_{id} \Bigg(\hat{f}_{id} \Bigg(\sum_{\substack{\mathbf{s} \in \mathbb{N}_0^d \\ \|\mathbf{s}\|_1 \leq q - \|\mathbf{l}\|_1}} \frac{\hat{\phi}_{3,M^d+j-1}^{(\mathbf{l}+\mathbf{s})}}{\mathbf{s}!} \cdot (\tilde{\mathbf{v}}_{j+1})^{\mathbf{s}} \Bigg) \\
&\qquad + \Bigg(\hat{f}_{trunc,j} \left((4 + 2 \cdot \lceil e^d \rceil) \cdot \hat{\phi}_{4,M^d+j-1}^{(\mathbf{l})} \right) \\
&\qquad\quad - \lceil e^d \rceil - 2 \Big) \cdot c_{36} \cdot \left(\frac{2a}{M^2} \right)^{p - \|\mathbf{l}\|_1} \Bigg), \\
\hat{\phi}_{4,M^d+j}^{(\mathbf{l})} &= \hat{f}_{id} \Big(\hat{f}_{id} \left((4 + 2 \cdot \lceil e^d \rceil) \cdot \hat{\phi}_{4,M^d+j-1}^{(\mathbf{l})} \right) \\
&\qquad - \hat{f}_{trunc,j} \left((4 + 2 \cdot \lceil e^d \rceil) \cdot \hat{\phi}_{4,M^d+j-1}^{(\mathbf{l})} \right) \Big)
\end{aligned}$$

for $j \in \{1, \dots, M^d - 1\}$.
Further we set

$$\begin{aligned}
\hat{\phi}_{5,M^d+j}^{(k)} &= \hat{f}_{test} \Big(\hat{\boldsymbol{\phi}}_{1,M^d+j-1}, \hat{\boldsymbol{\phi}}_{2,M^d+j-1}, \\
&\qquad \hat{\boldsymbol{\phi}}_{2,M^d+j-1} + \frac{2a}{M^2} \cdot \mathbf{1}, \hat{\phi}_{2,M^d+j-1}^{(k)} \Big) \\
&\quad + \hat{f}_{id}^2 \left(\hat{\phi}_{5,M^d+j-1}^{(k)} \right)
\end{aligned} \tag{42}$$

and

$$\hat{\phi}_{6,M^d+j}^{(\mathbf{l})} = \hat{f}_{test} \Big(\hat{\boldsymbol{\phi}}_{1,M^d+j-1}, \hat{\boldsymbol{\phi}}_{2,M^d+j-1},$$

$$\hat{\boldsymbol{\phi}}_{2,M^d+j-1} + \frac{2a}{M^2} \cdot \mathbf{1}, \hat{\phi}^{(\mathbf{l})}_{3,M^d+j-1}\Big) + \hat{f}^2_{id}\left(\hat{\phi}^{(\mathbf{l})}_{6,M^d+j-1}\right), \tag{43}$$

where $\hat{\boldsymbol{\phi}}_{5,M^d} = \left(\hat{\phi}^{(1)}_{5,M^d}, \ldots, \hat{\phi}^{(d)}_{5,M^d}\right) = \mathbf{0}$ and $\hat{\phi}^{(\mathbf{l})}_{6,M^d} = 0$ for each $\mathbf{l} \in \mathbb{N}_0^d$ with $\|\mathbf{l}_1\| \leq q$.

Again it is easy to see, that this parallelized and composed network needs $4M^d$ hidden layers and has width r with

$$r = 10d + 4d^2 + 2 \cdot \binom{d+q}{d} \cdot \left(2 \cdot (4 + 2\lceil e^d \rceil) + 5 + 2d\right).$$

Choose $\mathbf{l}_1, \ldots, \mathbf{l}_{\binom{d+q}{d}}$ such that

$$\left\{\mathbf{l}_1, \ldots, \mathbf{l}_{\binom{d+q}{d}}\right\} = \left\{\mathbf{s} \in \mathbb{N}_0^d : \|\mathbf{s}\|_1 \leq q\right\}$$

holds. The value of $\phi_{1,2M^d+1}$ can then be computed by

$$\hat{\phi}_{1,2M^d+1} = \hat{f}_p\left(\mathbf{z}, y_1, \ldots, y_{\binom{d+q}{d}}\right), \tag{44}$$

where

$$\mathbf{z} = \hat{\boldsymbol{\phi}}_{1,2M^d} - \hat{\boldsymbol{\phi}}_{5,2M^d}$$

and

$$y_v = \hat{\phi}^{(\mathbf{l}_v)}_{6,2M^d}$$

for $v \in \left\{1, \ldots, \binom{d+q}{d}\right\}$. The coefficients $r_1, \ldots, r_{\binom{d+q}{d}}$ in Lemma 19 are chosen as

$$r_i = \frac{1}{\mathbf{l}_i!}, \quad i \in \left\{1, \ldots, \binom{d+q}{d}\right\},$$

i.e. $\bar{r}(p) \leq 1$. The final network $\hat{\phi}_{1,2M^d+1}$ is then contained in the class

$$\mathcal{F}(4M^d + B_{M,p} \cdot \lceil \log_2(\max\{q+1, 2\}) \rceil, r)$$

with

$$r = \max\left\{10d + 4d^2 + 2 \cdot \binom{d+q}{d} \cdot \left(2 \cdot (4 + 2\lceil e^d \rceil) + 5 + 2d\right), 18 \cdot (q+1) \cdot \binom{d+q}{d}\right\}$$

and we set

$$\hat{f}_{deep,\mathcal{P}_2}(x) = \hat{\phi}_{1,2M^d+1}.$$

Note that the repeated composition of $\hat{f}_{id}$ and $\hat{f}_{ind,C_{j,1}}$ does not affect any weight constraints, since both networks have no offset in their respective output layers and the weights used in the input and output (i.e. last) layers are bounded by 1 respectively, such that Lemma 15 c) is applicable. This also holds for the repeated composition of $\hat{f}_{id}$ and $\hat{f}_{test}$. Since $\left\|\mathbf{v}_{\hat{f}_{id}^t}\right\|_\infty = 1$, we have $\left\|\mathbf{v}_{\hat{\boldsymbol{\phi}}_{1,j}}\right\|_\infty = 1$ for $j \in \{1, \dots, 2M^d\}$.
From Lemma 20, $((C_{j,1})_{left}) < a$, the conditions on M and since by definition $\tilde{\mathbf{v}}_{j+1}$ has entries in $\{0, \frac{2a}{M^2}\}$, we conclude that $\hat{\boldsymbol{\phi}}_{2,j}$ satisfies the weight constraint $\left\|\mathbf{v}_{\hat{\boldsymbol{\phi}}_{2,j}}\right\|_\infty \leq \max\left\{a + \frac{1}{B_M}, B_M\right\} \leq M^{2p+2}$ for $j \in \{1, \dots, 2M^d\}$.
Analogously, we can derive the weight constraints for $\hat{\phi}_{3,j}^{(\mathbf{l})}$ and $\hat{\phi}_{4,j}^{(\mathbf{l})}$:

$$\left\|\mathbf{v}_{\hat{\phi}_{3,j}^{(\mathbf{l})}}\right\|_\infty \leq \max\left\{\|f\|_{C^q([-a,a]^d)}, a + \frac{1}{B_M}, B_M\right\} \leq M^{2p+2} + c_{42}(f),$$

respectively for $\hat{\phi}_{4,j}^{(\mathbf{l})}$, we have

$$\left\|\mathbf{v}_{\hat{\boldsymbol{\phi}}_{4,j}^{(\mathbf{l})}}\right\|_\infty \leq \max\left\{a + \frac{1}{B_M}, B_M\right\} \leq M^{2p+2} \quad \text{and}$$

$\left\|\left(\mathbf{v}_{\hat{\phi}_{4,j}^{(\mathbf{l})}}\right)_{1,i>0}^{(0)}\right\|_\infty = 1$, $\left(\mathbf{v}_{\hat{\phi}_{4,j}^{(\mathbf{l})}}\right)_{1,0}^{(2j)} = 0$ and $\left\|\left(\mathbf{v}_{\hat{\phi}_{4,j}^{(\mathbf{l})}}\right)_{1,i>0}^{(2j)}\right\|_\infty \leq 1$, where we have used Lemma 20 and that by definition

$$b_i^{(\mathbf{l})} \in [0, 1].$$

Note that by construction $\left(\mathbf{v}_{\hat{\boldsymbol{\phi}}_{2,2M^d}}\right)_{1,0}^{(4M^d)} = 0$ and $\left\|\left(\mathbf{v}_{\hat{\boldsymbol{\phi}}_{2,2M^d}}\right)^{(4M^d)}\right\|_\infty \leq 1$. Since we have $\left\|\left(\mathbf{v}_{\hat{\phi}_{4,j}^{(\mathbf{l})}}\right)_{1,i>0}^{(2M^d)}\right\|_\infty \leq 1$, $\left\|\left(\mathbf{v}_{\hat{f}_{trunc}}\right)_{1,i>0}^{(0)}\right\|_\infty = 1$ and $\left(\mathbf{v}_{\hat{\phi}_{4,j}^{(\mathbf{l})}}\right)_{1,0}^{(2j)} = 0$ by Lemma 15 c) the composition of $\hat{f}_{trunc}$ and $\hat{\phi}_{4,j}^{(\mathbf{l})}$ does not affect the weight constraints. Thus $\hat{\phi}_{3,M^d+j}^{(\mathbf{l})}$ and $\hat{\phi}_{4,M^d+j}^{(\mathbf{l})}$ satisfy the constraint given in (40), i.e.

$$\left\|\mathbf{v}_{\hat{\phi}_{3,M^d+j}^{(\mathbf{l})}}\right\|_\infty = \left\|\mathbf{v}_{\hat{\phi}_{4,M^d+j}^{(\mathbf{l})}}\right\|_\infty \leq 4 + 2\lceil e^d \rceil + (4 + 2\lceil e^d \rceil)^{M^d} + (c_{36} + 1) \cdot M^{2p+2}.$$

By Lemma 20 a) and the previously stated weight constraints for $\hat{\boldsymbol{\phi}}_{1,M^d+j}$, $\hat{\boldsymbol{\phi}}_{2,M^d+j}$ and $\hat{\boldsymbol{\phi}}_{3,M^d+j}$, we have

$$\left\|\mathbf{v}_{\hat{\boldsymbol{\phi}}_{5,M^d+j}}\right\|_\infty \leq M^{4p+4}, \quad \left\|\left(\mathbf{v}_{\hat{\phi}_{5,M^d+j}}\right)^{(2M^d+2j)}\right\|_\infty \leq 1, \quad \left(\mathbf{v}_{\hat{\phi}_{5,M^d+j}}\right)_{1,0}^{(2M^d+2j)} = 0 \quad \text{and}$$

$$\left\|\mathbf{v}_{\hat{\phi}_{6,M^d+j}}\right\|_\infty \leq \max\left\{M^{4p+4}, \left\|\mathbf{v}_{\hat{\phi}_{3,M^d+j}^{(\mathbf{l})}}\right\|_\infty\right\} \leq 4 + 2\lceil e^d \rceil + (4 + 2\lceil e^d \rceil)^{M^d} + (c_{36} + 1) \cdot M^{4p+4},$$

$$\left\|\left(\mathbf{v}_{\hat{\phi}_{6,M^d+j}}\right)^{(2M^d+2j)}\right\|_\infty \leq 1, \quad \left(\mathbf{v}_{\hat{\phi}_{6,M^d+j}}\right)_{1,0}^{(2M^d+2j)} = 0.$$

To derive the weight constraints for the final network $\hat{f}_{deep,\mathcal{P}_2}(x)$, note that using the previously stated constraints for the weights of $\hat{\boldsymbol{\phi}}_{1,2M^d}$, $\hat{\boldsymbol{\phi}}_{5,2M^d}$ and $\hat{\boldsymbol{\phi}}_{6,2M^d}$ as well as Lemma 19, we know that the composition of $\hat{f}_p$ and $\hat{\boldsymbol{\phi}}_{1,2M^d} - \hat{\boldsymbol{\phi}}_{5,2M^d}$ fulfills the conditions of Lemma 15 c). This results in

$$\begin{aligned}
\left\|\mathbf{v}_{\hat{f}_{deep,\mathcal{P}_2}}\right\|_\infty &\leq \max\{4 \cdot 4^{2(q+1)} \cdot \Big(2 \cdot \max\left\{\|f\|_{C^q([-a,a]^d)}, a\right\} \cdot e^{(M^d-1)} + (4 + 2\lceil e^d \rceil) \\
&\qquad \cdot (M^d - 1) \cdot e^{(M^d-2)}\Big)^{2(q+1)}, (c_{36}+1) \cdot M^{4p+4}\} \\
&\leq \left(c_{43}(f) \cdot e^{M^d-1} \cdot \left(6 + 2\lceil e^d \rceil\right) \cdot M^d\right)^{2(q+1)} + (c_{36}+1) \cdot M^{4p+4} \\
&\leq e^{c_{44}(f)\cdot(p+1)\cdot M^d}
\end{aligned}$$

The rest of the proof follows as in Kohler and Langer (2021). □

A.2.3. Key step 3: Approximating $w_{\mathcal{P}_2}(x) \cdot f(x)$ by deep neural networks

In order to approximate $f(x)$ in supremum norm, a neural network that approximates $w_{\mathcal{P}_2}(x) \cdot f(x)$, where $w_{\mathcal{P}_2}(x)$ is defined as in (33), is required. The construction of such a network is given in the following three results.

Lemma 22 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Let $1 \leq a < \infty$ and $M \in \mathbb{N}_0$ sufficiently large (independent of the size of a, but*

$$\begin{aligned}
M^{2p} \quad \geq \quad & 2^{4(q+1)} \cdot \max\{c_{45}(6 + 2\lceil e^d \rceil)^{4(q+1)}, c_{36} \cdot e^d\} \\
& \cdot \left(\max\left\{a, \|f\|_{C^q([-a,a]^d)}\right\}\right)^{4(q+1)}
\end{aligned}$$

must hold). Let $p = q + s$ for some $q \in \mathbb{N}_0$, $s \in (0, 1]$ and let $C > 0$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a (p, C)-smooth function and let $w_{\mathcal{P}_2}$ be defined as in (33). Then there exists a network

$$\hat{f} \in \mathcal{F}(L, r)$$

with

$$\begin{aligned}
L =& 5M^d + \left\lceil \log_4\left(M^{2p+4\cdot d\cdot(q+1)} \cdot e^{4\cdot(q+1)\cdot(M^d-1)}\right)\right\rceil \cdot \lceil \log_2(\max\{q, d\} + 1)\rceil \\
& + \lceil \log_4(M^{2p})\rceil
\end{aligned}$$

and

$$\begin{aligned}
r =& \max\left\{10d + 4d^2 + 2 \cdot \binom{d+q}{d} \cdot \left(2 \cdot (4 + 2\lceil e^d \rceil) + 5 + 2d\right), \right. \\
& \left. 18 \cdot (q+1) \cdot \binom{d+q}{d}\right\} + 6d^2 + 20d + 2.
\end{aligned}$$

which satisfies

$$\left\|\mathbf{v}_{\hat{f}}\right\|_\infty \leq e^{c_{46}(f)\cdot(p+1)\cdot M^d}$$

such that

$$\left|\hat{f}(x) - w_{\mathcal{P}_2}(x) \cdot f(x)\right| \leq c_{47} \cdot \left(\max\left\{2a, \|f\|_{C^q([-a,a]^d)}\right\}\right)^{4(q+1)} \cdot \frac{1}{M^{2p}}$$

for $x \in [-a,a)^d$.

Further auxiliary lemmata are required to show this result. First, it is shown that each weight $w_{\mathcal{P}_2}(x)$ can also be approximated by a very deep neural network.

Lemma 23 *Let* $\sigma : \mathbb{R} \to \mathbb{R}$ *be the ReLU activation function* $\sigma(x) = \max\{x, 0\}$. *Let* $1 \leq a < \infty$ *and* $M \geq 4^{4d+1} \cdot d$. *Let* $\mathcal{P}_2$ *be the partition defined in (29) and let* $w_{\mathcal{P}_2}(x)$ *be defined by* (33). *Then there exists a neural network*

$$\hat{f}_{w_{\mathcal{P}_2},deep} \in \mathcal{F}(L, r),$$

with

$$L = 4M^d + 1 + \lceil \log_4(M^{2p}) \rceil \cdot \lceil \log_2(d) \rceil \quad \textit{and} \quad r = \max\left\{18d, 4d^2 + 10d\right\}$$

which satisfies

$$\left\|\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},deep}}\right\|_\infty \leq M^{4p+4}, \quad \left(\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},deep}}\right)_{1,0}^{(L)} = 0 \quad \textit{and} \quad \left\|\left(\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},deep}}\right)_{1,j>0}^{(L)}\right\|_\infty \leq 4^{3d+1}$$

such that

$$\left|\hat{f}_{w_{\mathcal{P}_2},deep}(x) - w_{\mathcal{P}_2}(x)\right| \leq 4^{4d+1} \cdot d \cdot \frac{1}{M^{2p}}$$

for $x \in \bigcup_{i \in \{1,\ldots,M^{2d}\}} (C_{i,2})^0_{1/M^{2p+2}}$ *and*

$$|\hat{f}_{w_{\mathcal{P}_2},deep}(x)| \leq 1$$

for $x \in [-a,a)^d$.

Proof. The first $4M^d$ hidden layers of $\hat{f}_{w_{\mathcal{P}_2}}$ compute the value of

$$(C_{\mathcal{P}_2}(x))_{left}$$

using $\hat{\boldsymbol{\phi}}_{5,2M^d}$ of Lemma 14 (with $d \cdot (2 \cdot (2d+2)+2) + 2d$ neurons per layer) and shift the value of x in the next hidden layer using the network $\hat{f}_{id}^{4M^d}$. As stated in the proof of Lemma 14, $\hat{\boldsymbol{\phi}}_{5,2M^d}$ has the following weight constraints $\left\|\mathbf{v}_{\hat{\boldsymbol{\phi}}_{5,M^d+j}}\right\|_\infty \leq M^{4p+4}$, $\left\|\left(\mathbf{v}_{\hat{\boldsymbol{\phi}}_{5,M^d+j}}\right)^{(2M^d+2j)}\right\|_\infty \leq 1$ and $\left(\mathbf{v}_{\hat{\boldsymbol{\phi}}_{5,M^d+j}}\right)_{1,0}^{(2M^d+2j)} = 0$. The next hidden layer then computes the functions

$$\left(1 - \frac{M^2}{a} \cdot \left|(C_{\mathcal{P}_2}(x))_{left}^{(j)} + \frac{a}{M^2} - x^{(j)}\right|\right)_+$$

$$
\begin{aligned}
&= \left(\frac{M^2}{a} \cdot \left(x^{(j)} - (C_{\mathcal{P}_2}(x))^{(j)}_{left} \right) \right)_+ \\
&\quad -2 \cdot \left(\frac{M^2}{a} \cdot \left(x^{(j)} - (C_{\mathcal{P}_2}(x))^{(j)}_{left} - \frac{a}{M^2} \right) \right)_+ \\
&\quad + \left(\frac{M^2}{a} \cdot \left(x^{(j)} - (C_{\mathcal{P}_2}(x))^{(j)}_{left} - \frac{2 \cdot a}{M^2} \right) \right)_+, \quad j \in \{1, \ldots, d\},
\end{aligned}
$$

using the networks

$$
\begin{aligned}
\hat{f}_{w_{\mathcal{P}_2,j}}(x) = \sigma &\left(\frac{M^2}{a} \cdot \left(\hat{f}_{id}^{4M^d}(x^{(j)}) - \hat{\phi}^{(j)}_{5,2M^d} \right) \right) \\
&- 2 \cdot \sigma \left(\frac{M^2}{a} \cdot \left(\hat{f}_{id}^{4M^d}(x^{(j)}) - \hat{\phi}^{(j)}_{5,2M^d} - \frac{a}{M^2} \right) \right) \\
&+ \sigma \left(\frac{M^2}{a} \cdot \left(\hat{f}_{id}^{4M^d}(x^{(j)}) - \hat{\phi}^{(j)}_{5,2M^d} - \frac{2 \cdot a}{M^2} \right) \right)
\end{aligned}
$$

with $3d$ neurons in the last layer. Note that $|\hat{f}_{w_{\mathcal{P}_2},j}(x)| \leq 1$ for $j \in \{1, \ldots, d\}$. The product of $w_{\mathcal{P}_2,j}(x)$ ($j \in \{1, \ldots, d\}$) can then be computed by the network $\hat{f}_{mult,d}(x)$ of Lemma 18 for values $x \in [-1,1]^d$, where we choose $x^{(j)} = \hat{f}_{w_{\mathcal{P}_2,j}}(x)$ and $R = \lceil \log_4(M^{2p}) \rceil$. Finally we set

$$
\hat{f}_{w_{\mathcal{P}_2},deep}(x) = \hat{f}_{mult,d} \left(\hat{f}_{w_{\mathcal{P}_2,1}}(x), \ldots, \hat{f}_{w_{\mathcal{P}_2,d}}(x) \right).
$$

Since by Lemma 18, in this case $\hat{f}_{mult,d}$ satisfies

$$
\left\| \mathbf{v}_{\hat{f}_{mult,d}} \right\|_\infty \leq 4 \cdot 4^{2d}, \qquad \left(\mathbf{v}_{\hat{f}_{mult,d}} \right)^{(R \cdot \lceil \log_2(d) \rceil)}_{1,0} = 0 \qquad \text{and}
$$

$$
\left\| \left(\mathbf{v}_{\hat{f}_{mult,d}} \right)^{(0)} \right\|_\infty \leq 1
$$

and by construction

$$
\left(\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},j}} \right)^{(4M^d+1)}_{1,0} = 0, \quad \left\| \left(\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},j}} \right)^{(4M^d+1)}_{1,i>0} \right\|_\infty \leq 2,
$$

$$
\left\| \mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},j}} \right\|_\infty = M^{4p+4}
$$

an application Lemma 15 b) gives us

$$
\left\| \mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},deep}} \right\|_\infty \leq \max \left\{ \left\| \mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},j}} \right\|_\infty, 4^{2d+1}, 2 \right\} = M^{4p+4}
$$

The rest of the proof follows as in the proof of Lemma 16 in Kohler and Langer (2021). □

Lemma 24 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU activation function $\sigma(x) = \max\{x, 0\}$. Let $1 \leq a < \infty$. Let $C_{i,2}$ ($i \in \{1, \ldots, M^{2d}\}$) be the cubes of partition $\mathcal{P}_2$ as described in (29) and let $M \in \mathbb{N}$. Then there exists a neural network*

$$\hat{f}_{check,deep,\mathcal{P}_2}(x) \in \mathcal{F}\left(5M^d, 2d^2 + 6d + 2\right)$$

satisfying

$$\left\|\mathbf{v}_{\hat{f}_{check,deep,\mathcal{P}_2}}\right\|_\infty \leq M^{4p+4}, \quad \left\|\mathbf{v}^{(5M^d)}_{\hat{f}_{check,deep,\mathcal{P}_2}}\right\|_\infty \leq 1$$

$$\hat{f}_{check,deep,\mathcal{P}_2}(x) = \mathbb{1}_{\bigcup_{i\in\{1,\ldots,M^{2d}\}} C_{i,2}\setminus(C_{i,2})^0_{1/M^{2p+2}}}(x)$$

for $x \notin \bigcup_{i\in\{1,\ldots,M^{2d}\}}(C_{i,2})^0_{1/M^{2p+2}}\setminus(C_{i,2})^0_{2/M^{2p+2}}$ and

$$\hat{f}_{check,deep,\mathcal{P}_2}(x) \in [0, 1]$$

for $x \in [-a, a)^d$.

Proof. The value of $(C_{\mathcal{P}_1}(x))_{left}$ is computed by the network $\hat{\boldsymbol{\phi}}_{2,M^d}$ of Lemma 14 with $2M^d$ hidden layers and $d\cdot(2d+2)$ neurons per layer and x is shifted in consecutive layers by successively applying $\hat{f}_{id} \in \mathcal{F}(1, 2)$. Furthermore we compute

$$f_1(x) = 1 - \sum_{i\in\{1,\ldots,M^d\}} \mathbb{1}_{(C_{i,1})^0_{1/M^{2p+2}}}(x)$$

by a network

$$\hat{f}_{1,j}(x) = \hat{f}^2_{id}(\hat{f}_{1,j-1}) - \hat{f}_{ind,(C_{j,1})^0_{1/M^{2p+2}}}(\hat{f}^{2(j-1)}_{id}(x)), \quad j \in \{1, \ldots, M^d\},$$

where $\hat{f}_{1,0} = 1$. Here we use again the network $\hat{f}_{ind,[\mathbf{a},\mathbf{b})]}$ from Lemma 20 a with $R = M^{2p+2}$. Next we define

$$\bar{\boldsymbol{\phi}}_{2,M^d+j} = \hat{f}^3_{id}(\hat{\boldsymbol{\phi}}_{2,M^d+j-1} + \tilde{\mathbf{v}}_{j+1}) \in \mathcal{F}(2M^d + 3j, 2d)$$

for $j \in \{1, \ldots, M^d\}$. It is easy to see that $\bar{\boldsymbol{\phi}}_{2,M^d+j}$ satisfies the same weight constraints as $\bar{\boldsymbol{\phi}}_{2,M^d+j}$, i.e. $\left\|\mathbf{v}_{\hat{\boldsymbol{\phi}}_{2,M^d+j}}\right\|_\infty \leq M^{2p+2}$,which we derived in the proof of Lemma 14 and by construction we have $\left(\mathbf{v}_{\bar{\boldsymbol{\phi}}_{2,2M^d}}\right)^{(4M^d)}_{1,0} = 0$ and $\left\|\left(\mathbf{v}_{\bar{\boldsymbol{\phi}}_{2,2M^d}}\right)^{(4M^d)}\right\|_\infty \leq 1$. The value of

$$\mathbb{1}_{\bigcup_{i\in\{1,\ldots,M^{2d}\}} C_{i,2}\setminus(C_{i,2})^0_{1/M^{2p+2}}}(x)$$

is then successively computed by

$$\hat{f}_{1,M^d+j}(x)$$

$$= 1 - \sigma \left(1 - \hat{f}_{test} \left(\hat{f}_{id}^{2M^d+3(j-1)}(x), \bar{\boldsymbol{\phi}}_{2,M^d+j-1} + \tilde{\mathbf{v}}_j + \frac{1}{M^{2p+2}} \cdot \mathbf{1}, \right. \right.$$
$$\left. \left. \bar{\boldsymbol{\phi}}_{2,M^d+j-1} + \tilde{\mathbf{v}}_j + \frac{2a}{M^2} \cdot \mathbf{1} - \frac{1}{M^{2p+2}} \cdot \mathbf{1}, 1 \right) - \hat{f}_{id}^2 \left(\hat{f}_{1,M^d+j-1} \right) \right)$$

for $j \in \{1, \ldots, M^d\}$, where we use networks $\hat{f}_{test}$ from Lemma 20 b with $R = M^{2p+2}$ and which thus satisfies $\left\| \mathbf{v}_{\hat{f}_{test}} \right\|_\infty \leq M^{4p+4}$).

Finally we set

$$\hat{f}_{check,deep,\mathcal{P}_2}(x) = \hat{f}_{1,2M^d}(x).$$

The weight constraints follow again from Lemma 15, arguing in the same way as in the proof of Lemma 14.

□

In the proof of Lemma 22 we use Lemma 23 to approximate $w_{\mathcal{P}_2}(x)$ and Lemma 14 to compute $f(x)$. As in Lemma 7 of Kohler and Langer (2021) we apply a network, that *checks* whether x is close to the boundaries of the cubes of the partition.

Proof of Lemma 22. This result follows by a straightforward modification of the proof of Lemma 7 of Kohler and Langer (2021). Here we use the network $\hat{f}_{deep,\mathcal{P}_2}$ of Lemma 14 and $\hat{f}_{check,deep,\mathcal{P}_2}$ of Lemma 22 to define

$$\begin{aligned} \hat{f}_{\mathcal{P}_2,true}(x) &= \sigma \left(\hat{f}_{deep,\mathcal{P}_2}(x) - B_{true} \cdot \hat{f}_{check,deep,\mathcal{P}_2}(x) \right) \\ &\quad - \sigma \left(-\hat{f}_{deep,\mathcal{P}_2}(x) - B_{true} \cdot \hat{f}_{check,deep,\mathcal{P}_2}(x) \right), \end{aligned}$$

with

$$B_{true} = 1 + \left| \left(\|f\|_{C^q([-a,a]^d)} \cdot e^{(M^d-1)} \right. \right.$$
$$\left. \left. + (4 + 2 \cdot \lceil e^d \rceil) \cdot (M^d - 1) \cdot e^{(M^d-2)} \right) \cdot e^{2ad} \right|$$

which satisfies

$$\begin{aligned} \left\| \mathbf{v}_{\hat{f}_{\mathcal{P}_2,true}} \right\|_\infty &\leq \max \left\{ B_{true} \cdot \left\| \mathbf{v}^{(5M^d)}_{\hat{f}_{check,deep,\mathcal{P}_2}} \right\|_\infty, \left\| \mathbf{v}_{\hat{f}_{deep,\mathcal{P}_2}} \right\|_\infty, \left\| \mathbf{v}_{\hat{f}_{check,deep,\mathcal{P}_2}} \right\|_\infty \right\} \\ &\leq e^{c_{48} \cdot (p+1) \cdot M^d} \end{aligned}$$

and $\left(\mathbf{v}_{\hat{f}_{\mathcal{P}_2,true}} \right)^{(L)}_{1,0} = 0, \quad \left\| \left(\mathbf{v}_{\hat{f}_{\mathcal{P}_2,true}} \right)^{(L)}_{1,j>0} \right\|_\infty \leq 1.$

Remark that by successively applying $\hat{f}_{id}$ to the output of the networks $\hat{f}_{deep,\mathcal{P}_2}$ and $\hat{f}_{check,deep,\mathcal{P}_2}$ we can achieve that both networks have depth

$$L = 5M^d + \left\lceil \log_4 \left(M^{2p+4 \cdot d \cdot (q+1)} \cdot e^{4 \cdot (q+1) \cdot (M^d-1)} \right) \right\rceil \cdot \lceil \log_2(\max\{q+1, 2\}) \rceil.$$

Furthermore, it is easy to see that this networks needs at most

$$\max\left\{10d+4d^2+2\cdot\binom{d+q}{d}\cdot\left(2\cdot(4+2\lceil e^d\rceil)+5+2d\right),\right.$$
$$\left.18\cdot(q+1)\cdot\binom{d+q}{d}\right\}+2d^2+6d+2$$

neurons per layer. In the definition of the final network we use the network $\hat{f}_{w_{\mathcal{P}_2},deep}$ of Lemma 23,and the network $\hat{f}_{mult}$

$$\hat{f}_{\text{mult}}\ \in\mathcal{F}\left(\left\lceil\log_4\left(M^{2p}\right)\right\rceil,18\right)$$

of Lemma 17 for

$$a=1+\left(\|f\|_{C^q([-a,a]^d)}\cdot e^{(M^d-1)}+(4+2\cdot\lceil e^d\rceil)\cdot(M^d-1)\cdot e^{(M^d-2)}\right)\cdot e^{2ad},$$

which satisfies the constraint

$$\left\|\mathbf{v}_{\hat{f}_{mult}}\right\|_\infty\leq 4\cdot 4^{2d}\cdot a^2\leq e^{c_{49}(f)}.$$

Again we synchronize the depth of $\hat{f}_{w_{\mathcal{P}_2},deep}$ and $\hat{f}_{\mathcal{P}_2,true}$ to insure that both networks have

$$\bar{L}=5M^d+\left\lceil\log_4\left(M^{2p+4\cdot d\cdot(q+1)}\cdot e^{4\cdot(q+1)\cdot(M^d-1)}\right)\right\rceil\cdot\lceil\log_2(\max\{q,d\}+1)\rceil$$

many layers. The final network is given by

$$\hat{f}(x)=\hat{f}_{mult}\left(\hat{f}_{\mathcal{P}_2,true}(x),\hat{f}_{w_{\mathcal{P}_2},deep}(x)\right).$$

By Lemma 23 and by definition of $\hat{f}_{\mathcal{P}_2,true}(x)$, it is easy to see that both networks have no offset in their respective output layers, thus Lemma 15 b) is applicable and we get

$$\begin{aligned}\left\|\mathbf{v}_{\hat{f}}\right\|_\infty\leq&\max\left\{\left\|\mathbf{v}_{\hat{f}_{mult}}\right\|_\infty,\max\left\{\left\|\mathbf{v}_{\hat{f}_{\mathcal{P}_2,true}}\right\|_\infty,\left\|\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},deep}}\right\|_\infty\right\},\right.\\&\left.\max\left\{\left\|\left(\mathbf{v}_{\hat{f}_{\mathcal{P}_2,true}}\right)^{(\bar{L})}\right\|_\infty\cdot\left\|\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(0)}\right\|_\infty,\left\|\left(\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},deep}}\right)^{(\bar{L})}\right\|_\infty\cdot\left\|\left(\mathbf{v}_{\hat{f}_{mult}}\right)^{(0)}\right\|_\infty\right\}\right\}\\\leq&\max\left\{e^{c_{49}(f)\cdot M^d},\max\left\{\left\|\mathbf{v}_{\hat{f}_{\mathcal{P}_2,true}}\right\|_\infty,\left\|\mathbf{v}_{\hat{f}_{w_{\mathcal{P}_2},deep}}\right\|_\infty\right\}\right\}\\\leq&e^{c_{50}\cdot(p+1)\cdot M^d}\end{aligned}$$

This network is contained in the network class $\mathcal{F}(L,r)$ with

$$L=5M^d+\left\lceil\log_4\left(M^{2p+4\cdot d\cdot(q+1)}\cdot e^{4\cdot(q+1)\cdot(M^d-1)}\right)\right\rceil\cdot\lceil\log_2(\max\{q,d\}+1)\rceil$$

$$+ \lceil \log_4(M^{2p}) \rceil$$

and

$$r = \max \left\{ 10d + 4d^2 + 2 \cdot \binom{d+q}{d} \cdot \left(2 \cdot (4 + 2\lceil e^d \rceil) + 5 + 2d \right), \right.$$
$$\left. 18 \cdot (q+1) \cdot \binom{d+q}{d} \right\} + 2d^2 + 6d + 2 + \max\{18d, 4d^2 + 10d\}.$$

With the same argument as in the proof of Lemma 10 of Kohler and Langer (2021) we can show the assertion. □

In a last step of the proof, one has to apply $\hat{f}$ to slightly shifted partitions. This follows as in section A.1.12 of Kohler and Langer (2021) and has no effect on the weight bounds that were just derived.