Risk bounds for deep learning

Thijs Bos

Risk bounds for deep learning
Author: Thijs Bos
176 pages

PhD thesis, Universiteit Leiden, The Netherlands (2024)
With summary in English and Dutch

ISBN: 978-94-6469-978-4

Printed by ProefschriftMaken || www.proefschriftmaken.nl

The research described in this thesis was financially supported by the NWO/STAR grant 613.009.034b and the NWO Vidi grant VI.Vidi.192.021.

Risk bounds for deep learning

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op woensdag 19 juni 2024
klokke 10:00 uur

door

Johan Matthijs Bos
geboren te Bergschenhoek
in 1991

Promotores:
Prof.dr. P.D. Grünwald (Universiteit Leiden en Centrum Wiskunde & Informatica)
Prof.dr. A.J. Schmidt-Hieber (Universiteit Twente)

Promotiecommissie:
Prof.dr.ir. G.L.A. Derks
Prof.dr. J.J. Goeman
Prof.dr. A. Rohde (Universität Freiburg)
Prof.dr. M. Kohler (Technische Universität Darmstadt)
Dr. R.M. Castro (Technische Universiteit Eindhoven)

Contents

Chapter 1

Introduction

Deep learning is, broadly speaking, the training of artificial neural networks on data for prediction tasks. It became popular in the 2010s after achieving hugely improved performance on benchmark datasets used in machine-learning competitions [60, 80, 51, 126]. This was the start of a revolution. Within a few years, deep neural networks achieved (super)human level performance in visual object recognition and speech recognition tasks [126, 30, 58, 3]. By now, deep learning is also successfully used in various other applications.

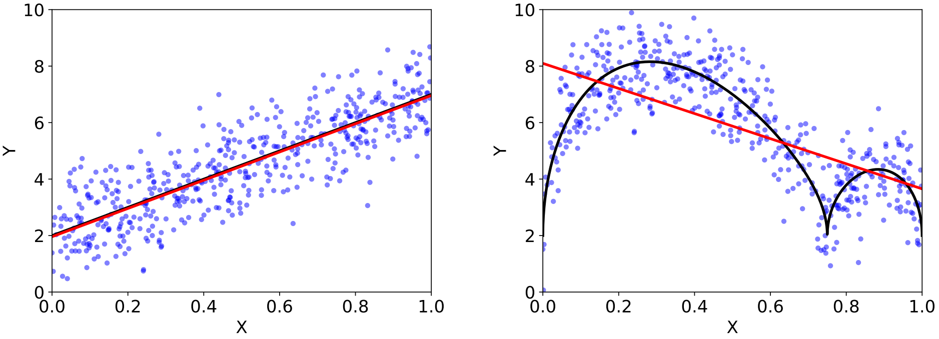
In this thesis, we approach deep learning from the statistical viewpoint: what can be said about the expected error if we view deep learning as a statistical estimation method. This perspective connects machine learning and statistics.

The introduction is structured as follows. The statistical background is introduced in Section 1.1. Section 1.2 provides an overview of deep learning. Sections 1.3, 1.4 and 1.5 briefly introduce the subjects of each of the later chapters.

1.1 Statistical background

Deep learning gained momentum by providing state-of-the-art procedures for supervised learning tasks, where one observes data-pairs $(\mathbf{X}_i, \mathbf{Y}_i)$, with $\mathbf{X}_i$ an input vector and $\mathbf{Y}_i$ the corresponding output. This setting is called supervised because the output $\mathbf{Y}_i$ is already provided (as if by a teacher), contrary to unsupervised learning where the algorithm is fed with unlabeled data. Regression and classification are the two main sub-classes of supervised learning. In classification, the goal is to predict the class a new data point belongs to. Examples include spam detection and image recognition. For instance, in the latter case, the data-pairs could consist of images of different types of animals (the input) and their corresponding labels 'dog', 'cat', etc (the output). For a new image, the neural network must predict which category this image belongs to.

In regression, the output is a real-valued response. For example, predicting income based on years of education. Various (nonparametric) methods have been developed for regression problems. Many of them are also theoretically well-understood. First, we will give a concise introduction of nonparametric statistics, mostly restricting ourselves to regression. Classification will be treated in Section 1.3. After introducing nonparametric methods, we will describe the decision theoretic concepts of loss, risk, and convergence rates. This is followed by sections on empirical risk minimization and the challenges arising in high-dimensional input-spaces.



(a) Linear

(b) Non-linear

Figure 1.1: Plots of the samples (blue), linear least squares estimator (red) and the true regression function (black) of a linear and a non-linear model.

1.1.1 Nonparametric statistics

In statistics, recurring questions are: what does one already know about the distribution of the data or what can one reasonably assume about it? This is of particular importance for complex statistical models. Below, we illustrate this using the nonparametric regression model, as this is a widely accepted framework to study supervised learning.

Example 1.1.1 (Regression)**.** In the (multivariate) regression model we observe n random pairs $(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots, (\mathbf{X}_n, Y_n)$ satisfying

$$Y_i = f_0(\mathbf{X}_i) + \epsilon_i.$$

Here the ϵ_i are noise variables and f_0 is an unknown deterministic function. Models of this form are also called signal plus noise models. The regression problem is to infer f_0 from the data. It is common to assume that the noise variables ϵ_i are independent and this assumption will be used throughout this chapter.

The available prior information plays a crucial role. For instance, suspecting that the regression function f_0 is linear, all that remains is to estimate the parameters of the linear function. In Figure 1.1a, data from a linear regression problem are plotted in blue and the parametric linear least squares estimator is displayed in red. Linear regression is an example of a parametric model. Parametric models are statistical models that depend only on a finite number of parameters. See for example [152, 119] for an introduction and overview of parametric methods. However, often far less is known about the underlying model. For instance, it is clear that the (blue) data-sample in Figure 1.1b was not generated from a linear regression function. The estimate of the (red) linear least squares estimator does not reflect the data. It is, however, not immediately clear which parametric family of functions would be a better candidate. In cases like this, a method is required that assumes less about the structure of the data. In nonparametric statistics, problems are studied that do not satisfy a parametric model. Instead, one considers problems that are too big to be parametrized by a finite number of parameters. For instance, in the regression problem, Example 1.1.1, we can assume that the function f_0 is in the class of all continuous functions. This class is so large that it is impossible to represent it by a finite basis expansion.

There exist many different nonparametric regression estimation methods. Here we provide two examples: the Nadaraya-Watson estimator, Example 1.1.2, and the Fourier-series estimator, Example 1.1.3.

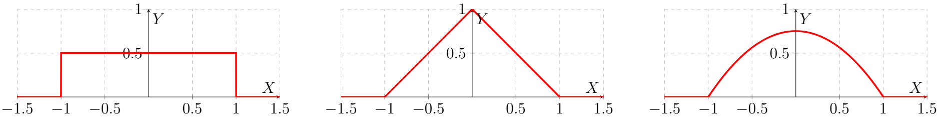
Example 1.1.2 (Nadaraya-Watson estimator)**.** A kernel in d-dimensions is a function $K:\mathbb{R}^d\to\mathbb{R}$ such that $\int_{\mathbb{R}^d}K(\mathbf{u})\,d\mathbf{u}=1$. Some standard kernels in one dimension are displayed in Figure 1.2.

The Nadaraya-Watson estimator f_{NW} with kernel K and bandwidth $h>0$ is given by

$$\widehat{f}_{NW}(\mathbf{x}):=\sum_{i=1}^{n}Y_i\frac{K\Big(\frac{1}{h}(\mathbf{X}_i-\mathbf{x})\Big)}{\sum_{j=1}^{n}K\Big(\frac{1}{h}(\mathbf{X}_j-\mathbf{x})\Big)},$$

when $\sum_{j=1}^{n}K\Big(\frac{\mathbf{X}_j-\mathbf{x}}{h}\Big)\neq 0$ and $f_{NW}(\mathbf{x})=0$ otherwise.

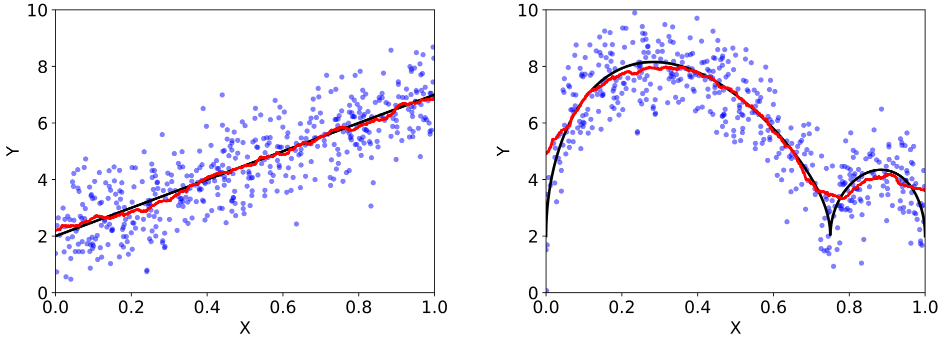
The Nadaraya-Watson estimator was first proposed in 1964 by Nadaraya [100] and Watson [154]. The estimate in a point $\mathbf{x}$ is determined by a weighted average of the output variables Y_i. The weight assigned to each sample is determined by the chosen kernel K and the bandwidth h. The role of the bandwidth h in tuning the



(a) Rectangular kernel (b) Triangular kernel (c) Epanechnikov kernel

Figure 1.2: Three examples of standard kernels in one dimension.

estimator will be discussed in Section 1.1.3. In Figure 1.3 the (red) Nadaraya-Watson estimator with the rectangular kernel is shown on the linear and non-linear regression dataset from before. Compared to the linear least squares estimator in Figure 1.1, this estimator provides a reasonable estimate for both problems.



(a) Linear (b) Non-linear

Figure 1.3: Plots of the samples (blue), Nadaraya-Watson estimator (red) and the true regression function (black) of a linear and a non-linear model.

Example 1.1.3 (Fourier-series estimator)**.** Consider the problem of estimating a function in the function space $L^2[0,1]$: the space of square integrable functions on the interval $[0,1]$. The Fourier basis in $L^2[0,1]$ is given by

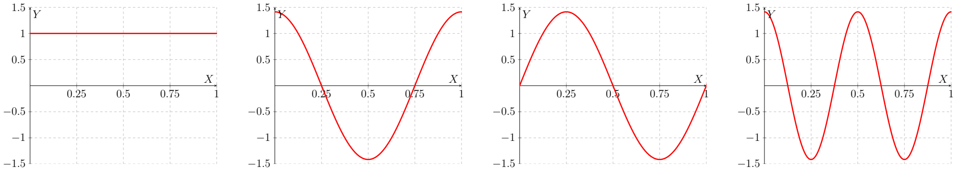
$$\begin{aligned}\psi_1(x) &:= 1,\\ \psi_{2k}(x) &:= \sqrt{2}\cos(2k\pi x),\\ \psi_{2k+1}(x) &:= \sqrt{2}\sin(2k\pi x),\end{aligned}$$

for $k = 1, 2, \ldots$. In other words, for every function $f \in L^2[0,1]$, there exist coefficients $(c_j)_{j=1}^{\infty}$ such that f can be written as $f(x) = \sum_{j=1}^{\infty} c_j \psi_j$. The first four Fourier basis functions are plotted in Figure 1.4.

The Fourier-series estimator of level N and with threshold τ is given by

$$\widehat{f}_F(x) := \sum_{j=1}^{N} \widehat{c}_j \psi_j(x) \mathbb{1}\big(|\widehat{c}_j| \geq \tau\big),$$

where $\widehat{c}_j := n^{-1} \sum_{i=1}^{n} Y_i \psi_j(X_i)$ is the empirical Fourier coefficient. Here $\mathbb{1}(\cdot)$ denotes the indicator function, returning one when $|\widehat{c}_j| \geq \tau$ and zero otherwise.



(a) ψ_1 (b) ψ_2 (c) ψ_3 (d) ψ_4

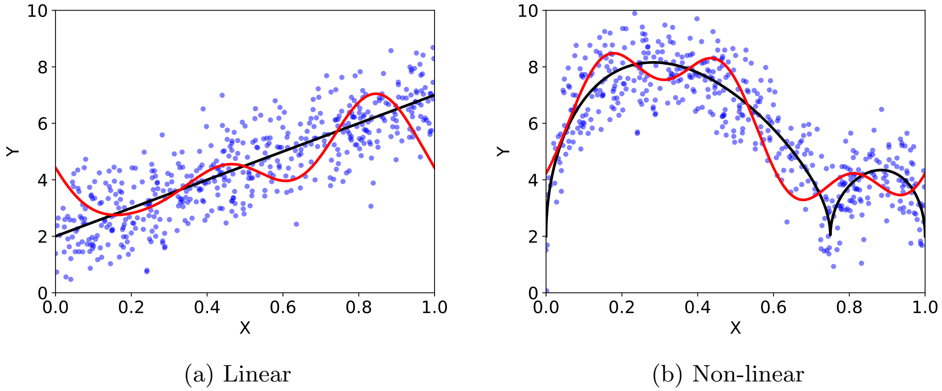
Figure 1.4: The first four Fourier basis functions

The Fourier-series estimator is an example of a projection estimator (also known as orthogonal series estimator), [153, 146, 44], as it projects on the space spanned by the first N basis elements and then estimates the coefficients of these N elements. The choice of the (orthogonal) basis determines how one can interpret an orthogonal series estimator. In the case of the Fourier-series estimator, the basis functions represent frequencies. In Figure 1.5 the (red) Fourier-series estimator is shown on the linear and non-linear regression dataset from before. The waveform of the sine and cosine functions used in the Fourier-basis can be seen back in the estimated function.

For a further introduction and overview of nonparametric estimation methods see for example [153, 146, 54].

1.1.2 Loss, risk and minimax rates

An important aspect of statistical estimation theory is to provide a quantification of how 'good' an estimator is. Consider for instance the regression problem, Example 1.1.1; Given an estimator $\widehat{f}$, how far away is it from the true regression function f_0? The first step is to choose a loss function for measuring the estimation error.



(a) Linear (b) Non-linear

Figure 1.5: Plots of the samples (blue), Fourier-series estimator (red) and the true regression function (black) of a linear and a non-linear model.

Definition 1.1.4 (Loss-function)**.** Let $\mathcal{G}$ represent the class containing the quantity to be estimated. Denote by $\mathcal{F}$ the class in which the estimator takes its values. A loss function ℓ assigns a non-negative number to any combination of $g \in \mathcal{G}$ and $f \in \mathcal{F}$ and returns zero if $f = g$.

For nonparametric regression, an example is the squared pointwise loss: $\ell(\widehat{f}, f_0) = \big(\widehat{f}(\mathbf{x}) - f_0(\mathbf{x})\big)^2$ with $\mathbf{x}$ a given point. This loss is derived from the log-likelihood of the regression model with Gaussian noise. In classification, the goal is to predict a label y. A common choice for this task is the zero-one loss $\ell(\widehat{f}, y) = \mathbb{1}(\widehat{f} \neq y)$. This loss returns zero if the predicted label is correct and one otherwise.

To evaluate the performance of an estimator we consider the expected average loss over all possible realizations of the data.

Definition 1.1.5 (Risk)**.** The risk of an estimator $\widehat{f}$ with respect to the loss function ℓ is given by $R(\widehat{f}, g) := \mathbb{E}_g\Big[\ell\big(\widehat{f}, g\big)\Big]$. Here the expectation $\mathbb{E}_g$ is over the data distribution with parameter g.

In general the data distribution may also depend on other parameters besides g.

To compare estimators, we look at their worst-case risk over all possible parameters g. In other words, we are interested in $\sup_{g \in \mathcal{G}} R(\widehat{f}, g)$, where $\mathcal{G}$ denotes the class containing the quantity to be estimated. The (worst-case) risk cannot be computed exactly in most cases. Instead, one relies on upper bounds. For a sensible estimator, this upper bound should converge to zero as the sample size n increases. The rate at which the upper bound decreases is called the convergence rate.

Ideally, the worst-case risk should decrease to zero as fast as possible, but how fast can this possibly happen? The answer to this question depends on the imposed assumptions. For instance, for the parametric linear regression problem a faster rate of convergence can be reached, than for a nonparametric regression problem that involves all differentiable functions. It can be shown that without assumptions, the worst-case risk in the nonparametric regression model is lower bounded by some constant no matter how much data is available, see for example Theorem 3.1 of [54] or Section 7 of [38].

For many classes it is possible to prove lower bounds on the rate of convergence of the worst-case risk of any estimator. In other words, it can be shown that $\inf_{\widetilde{f}} \sup_{g \in \mathcal{G}} R(\widetilde{f}, g)$ cannot tend to zero faster than a certain rate. For an introduction to lower bounds, see for example [146]. Lower bounds for standard nonparametric regression settings were proven in [138].

If the rate of the lower bound matches the convergence rate of some estimator for this estimation problem, then the rate is called minimax (rate) optimal.

The main focus in this thesis are convergence rates for worst-case upper bounds in high-dimensional input settings related to deep learning. The empirical counterpart of the risk plays a crucial role in establishing these bounds.

1.1.3 Empirical risk minimization

The chosen risk determines the quality of the estimator and a 'good' estimator will have small risk. However, direct minimization of the risk is impossible; Computing the expected value in the definition of the risk requires knowledge of the underlying unknown data-distribution. To overcome this, one can replace the expectation by an average.

Definition 1.1.6 (Empirical risk)**.** The empirical risk of an estimator $\widehat{f}$ with respect to the loss function ℓ is given by $R_n(\widehat{f}) := \frac{1}{n}\sum_{i=1}^n \ell\big(\widehat{f}(\mathbf{X}_i), Y_i\big)$. In other words, the empirical risk is the average loss over all data-pairs $(\mathbf{X}_i, Y_i)$ in the dataset. An empirical risk minimizer with respect to the class $\mathcal{F}$ is any estimator $\widehat{f}$ satisfying $\widehat{f} \in \arg\min_{f \in \mathcal{F}} R_n(f)$.

The hope is that minimizing the empirical risk results in an estimator with low risk. In other words, it leads to a procedure that generalizes well to unseen samples. Plenty of theory has been developed showing that this is indeed true if the estimator class is not too large or too small, see for example [148, 150]. If the estimator class is too large, then too much of the noise gets incorporated into the estimator. In the extreme case, the estimator interpolates the data points. In this instance the empirical risk is zero, but depending on the loss function, the risk may be arbitrarily large. The estimator class in empirical risk minimization should also be not too small. Otherwise, the best function in the class could still be far away from the true function. The error that arises from this difference is known as the approximation error, while the error that comes from the vulnerability of the estimator to noise is called the stochastic error. A major challenge of empirical risk minimization is to choose function classes that balance approximation and stochastic error.

Both the Nadaraya-Watson estimator, Example 1.1.2, and the Fourier-series estimator, Example 1.1.3, can be derived as variants of empirical risk minimizers for different estimator classes. The linear least squares estimator for linear regression, used in Figure 1.1, is a parametric example of an empirical risk minimizer. It minimizes the empirical risk for the squared loss $\ell(\widehat{f}(\mathbf{x}), y) = (\widehat{f}(\mathbf{x}) - y)^2$ over the class of all linear functions.

The Nadaraya-Watson estimator $\widehat{f}_{NW}$, with nonnegative kernel K is a minimizer of a localized version of the empirical risk,

$$\widehat{f}_{NW}(\mathbf{x}) = \operatorname*{arg\,min}_{\theta \in \mathbb{R}} \sum_{i=1}^{n} (Y_i - \theta)^2 K\left(\frac{1}{h}\left(\mathbf{X}_i - \mathbf{x}\right)\right). \tag{1.1.1}$$

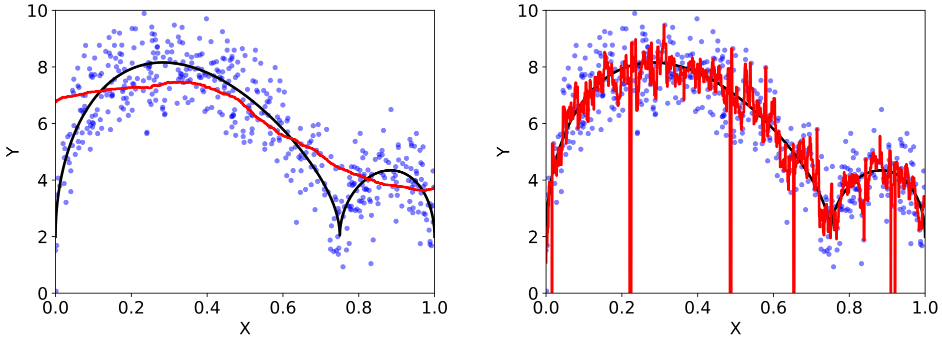
The family of local polynomial estimators can be obtained by replacing θ in (1.1.1) by a polynomial, see for example [146, 153, 93, 54]. The Nadaraya-Watson estimator is therefore a specific local polynomial estimator. For kernel estimators, the bandwidth parameter h controls the trade-off between the approximation and stochastic error. Larger h means that the neighborhood that is included in the kernel increases. This leads to a smoother estimate with a smaller stochastic error, but with a larger approximation error. On the other hand, a smaller h means that a smaller neighborhood is considered by the kernel. This results in a less smooth estimate with smaller approximation error, but a larger stochastic error. In Figure 1.6 the Nadaraya-Watson estimator for large and small bandwidth is plotted, using the same kernel and dataset as in Figure 1.3b.

Now consider the Fourier-series estimator $\widehat{f}_F(x)$ of level N with threshold τ as defined in Example 1.1.3. The linear combinations of the first N Fourier-basis functions form the class

$$\mathcal{F}_N := \left\{ \sum_{j=1}^{N} c_j \psi_j, c_j \in \mathbb{R} \text{ for } j = 1, 2, \ldots, N \right\}.$$

Using this class, the Fourier-series estimator can be rewritten as

$$\widehat{f}_F(x) = \operatorname*{arg\,min}_{f \in \mathcal{F}_N} \frac{1}{n} \sum_{i=1}^{n} \big(f(X_i) - Y_i\big)^2 + \frac{\lambda_\tau}{n} \operatorname{pen}(f), \tag{1.1.2}$$



(a) Large bandwidth

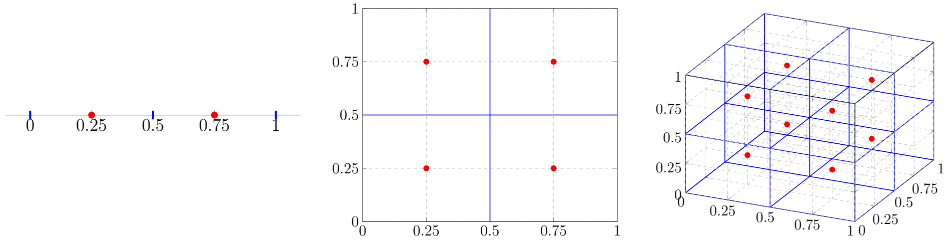
(b) Small bandwidth

Figure 1.6: Plots of the samples (blue), Nadaraya-Watson estimator (red) with a large bandwidth h on the left and a small bandwidth h on the right.

where the penalty $\mathrm{pen}(f)$ counts the number of non-zero coefficients c_j of f and the parameter λ_τ depends on the value of the threshold τ. This estimator thus minimizes a penalized/regularized version of the empirical risk. The number of terms considered in the series estimator N and the threshold τ control the approximation and stochastic error. Including more terms, thus increasing N and decreasing τ, leads to a smaller approximation error and a larger stochastic error. Including less terms decreases the stochastic error and increases the approximation error.

1.1.4 Curse of dimensionality

Nonparametric methods work well for low-dimensional input problems. However, in higher dimensions their performance degrades. The (input) dimension d appears in the minimax rate of many nonparametric problems in the power of n. For example, the minimax rate for the squared loss for estimating a β times differentiable function is of order $n^{-2\beta/(2\beta+d)}$, [137, 138, 54, 44]. As d grows, this rate gets slower. The reason for this is that in higher dimensions one needs (many) more data-points before every local neighborhood contains at least one observation. To see this, consider the problem of placing points in the space $[0,1]^d$ such that everywhere in $[0,1]^d$ one is at most $1/4$ in the maximum-norm distance away from one of these points. This is depicted in Figure 1.7. One needs $2 = 2^1$ points in one dimension, $4 = 2^2$ points in two dimensions and $8 = 2^3$ points in three dimensions. In general, one needs 2^d points in dimension d, an exponential growth in the dimension.



(a) Dimension 1 (b) Dimension 2 (c) Dimension 3

Figure 1.7: Required number of points to cover the cube $[0,1]^d$ with maximum norm balls of radius $1/4$.

In contrast, the dimension d appears in the minimax rate for parametric problems only as a multiplicative constant. For example, for linear regression the minimax rate for the squared loss is d/n, [63, 98, 144]. The strong structural assumptions in parametric models reduce the intrinsic dimension of the problem. This is related to the main idea behind approaches to tackle the curse of dimensionality in nonparametric regression: Introduce additional assumptions on the structure such that the intrinsic dimension of the function class becomes smaller. An example are generalized additive models, [56], assuming that the multivariate function can be written as a linear combination of univariate functions. Methods that make use of this structure are able to achieve the dimensionless rate $n^{-2\beta/(2\beta+1)}$ for β times differentiable functions instead of $n^{-2\beta/(2\beta+d)}$, see for example [139] or Section 22 of [54]. The possibility that deep neural networks are able to circumvent the curse of dimensionality is one of the motivations for the statistical analysis of deep learning.

1.2 Deep learning

Around 1960, the first trainable artificial neural networks were developed: The perceptron of Rosenblatt [121, 122] has zero-one output, and the adaptive linear element (ADALINE) [156] is a linear model trained with (a version of) gradient descent. Both the perceptron and ADALINE existed as dedicated physical machines, in contrast to modern neural networks which are software implementations.

In essence, these first neural networks consisted of one single trainable neuron. More neurons and layers were proposed [121, 122, 156], but in those setups only the parameters of the last neuron were changed during training. The other parameters were kept fixed [121, 150]. This means that, from a statistical point of view, all these early neural networks are parametric methods.

In Figure 1.8 a neural network with a single neuron is shown. This neural network multiplies each input coefficient x_i with a weight W_i. It then takes the sum over all these weighted inputs and adds the shift v to this sum. Finally, the activation function σ is applied in the single neuron. The trainable parameters in this neural network are the weights W_i and the shift v. This simple neural network already can be used for various tasks by choosing a suitable activation function σ. The original perceptron used the heaviside function $\sigma(x) = \mathbb{1}(x \geq 0)$, Figure 1.9a. With this activation function the neural network can be used for binary classification, e.g., for answering yes or no questions. When, as in ADALINE, σ is the identity function, Figure 1.9c, this neural network does linear regression. Taking σ as the logistic function $\sigma(x) = 1/(1+e^{-x})$, Figure 1.9b, results in a neural network that does logistic regression, estimating the probability of an event. Sigmoid activation functions, such as the logistic function, can be considered as smooth alternatives for the heaviside function. These activation functions became the standard in the late eighties.

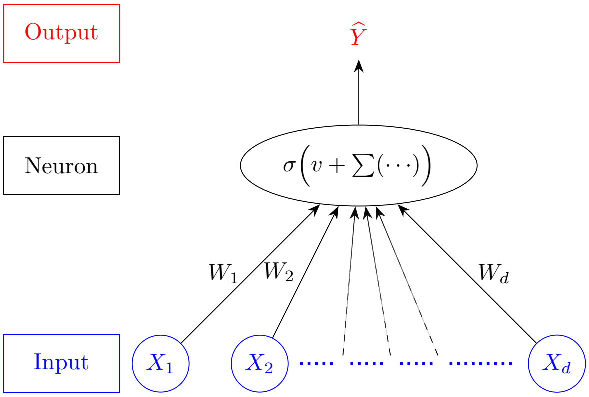
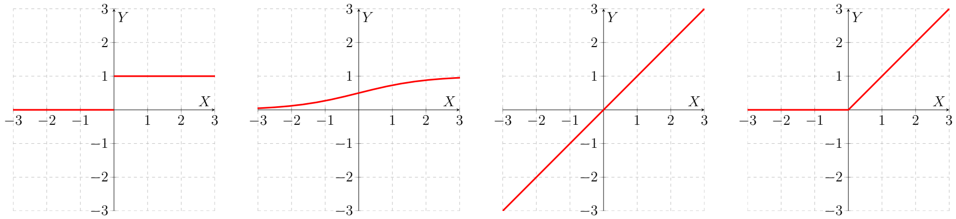


Figure 1.8: A neural network with one single neuron.

Modern deep neural networks consist of multiple neurons ordered in layers. A neural network consists of an input layer, several hidden layers, and an output layer. The number of hidden layers is also called the depth of the neural network, and the 'deep' in deep learning refers to neural networks with multiple layers. In Figure 1.10 an example of a deep neural network is given. This example has d inputs, three hidden layers with four neurons each, and a single neuron as its output. Such a neural network is called a fully connected feedforward network: fully connected since



(a) Heaviside

(b) Logistic

(c) Linear

(d) Rectified Linear Unit (ReLU)

Figure 1.9: Examples of activation functions

every node in a layer is connected to all nodes in the previous and the next layer, feedforward because all connections go forward to the next layer. In fully connected feedforward networks, all hidden units generally have the same activation function σ. Nowadays, the most common choice is the rectified linear unit (ReLU) activation function, Figure 1.9d. This is the activation function used in the hidden layers of the neural networks considered in Chapters 2 and 3. Depending on the learning task, the output may use different activation functions. The linear activation function is used for regression/function estimation problems. Chapter 3 considers neural networks with this function in the output layer. For estimating a binary probability the logistic function is the standard choice for the activation function of the output neuron. Chapter 2 deals with estimating vectors of probabilities. For this task a multivariate version of the logistic function, the softmax function, is used as output activation function.

Around 1990 it was shown that shallow neural networks, neural networks with one hidden layer, have the so-called universal approximation property: If the number of neurons in the hidden layer is allowed to become arbitrarily large, then shallow neural networks are able to approximate any continuous function arbitrarily well, [34, 61, 46]. Multi-neuron networks are thus able to approximate large function classes if the neural network size, and thus the number of parameters, grows as a function of the data. This means that multi-neuron networks can be considered as nonparametric methods.

In the nineties it was proven that shallow neural networks could approximate specific classes of multivariate functions with a rate whose power of n does not contain the input dimension d, [65, 7, 8, 27]. However, these classes come with Fourier transform conditions that depend on d: if the dimension increases, then these conditions become more restrictive. More recently, it has been shown that deep neural networks with ReLU activation function can approximate various classes of functions better than shallow neural networks, [160, 141, 110]. This is in contrast to older works which considered smooth and bounded activation functions, such as the logistic activation function, Figure 1.9b. Following these approximation results, convergence rates for the risk in the regression problem were derived. These results showed that if the regression function has a compositional structure, then deep neural networks can exploit this structure to circumvent the curse of dimensionality, [71, 62, 72, 113, 11, 127, 77].

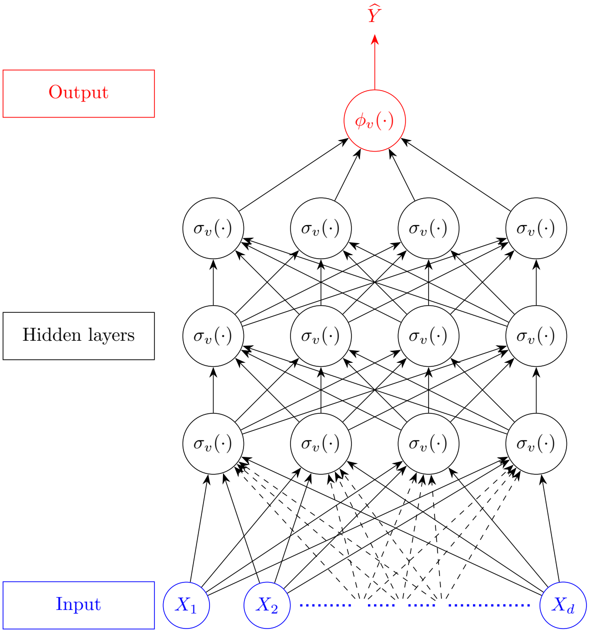


Figure 1.10: A neural network with three hidden layers with four neurons each, an input layer with d inputs, and an output layer with a single output neuron.

1.2.1 Training of neural networks

For supervised learning problems training a neural network means minimizing a training loss. In other words, deep learning is an example of a (regularized) empirical risk minimization approach as discussed in Section 1.1.3. Unlike the examples in that section, it is impossible to derive an explicit solution of the minimization problem. Instead, the empirical risk is minimized stepwise during training by an optimization method. The most common method is gradient descent (or a variation thereof): In each training step the parameters are updated according to the update rule:

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \alpha_k \nabla_{\boldsymbol{\theta}_{k-1}} \left(\frac{1}{n} \sum_{i=1}^{n} \ell\big(f_{\boldsymbol{\theta}_{k-1}}(\mathbf{X}_i), Y_i\big) \right), \tag{1.2.1}$$

for a sequence of positive numbers α_k called the learning rate and $f_{\boldsymbol{\theta}_{k-1}}$ the neural network with parameters $\boldsymbol{\theta}_{k-1}$. In the neural networks as described before, the parameters $\boldsymbol{\theta}$ are all the weight matrices W and all the shift vectors v. For convex problems, gradient descent converges to the global minimum for suitable sequences of learning rates. Neural networks viewed as functions do not lead to a convex function class. Training neural networks is therefore a non-convex problem. Instead of one global minimum, there may exist multiple minima, which may be local or global [37, 125, 29]. As a consequence, a minimum found during training might be a local minimum and in this case there exist neural network-parameters that achieve a lower training loss.

Training neural networks with multiple layers of neurons became feasible with the introduction of backpropagation in 1986 in [124] and [86]. Backpropagation consists of a forward and a backward pass through the neural network. In the forward pass, the neural network is given a training sample as input and the output of the neural network is computed for this sample. In the backward pass, the output is used to calculate the training loss, after which the gradient for all parameters is calculated using the chain-rule for differentiation.

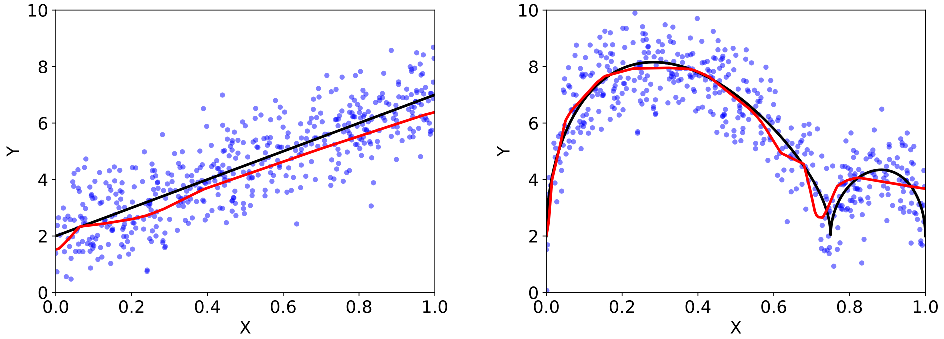
In Figure 1.11 estimates by fully connected feedforward networks for the regression dataset from Section 1.1 are shown. These neural networks have 10 hidden layers with 50 neurons each and use the ReLU activation function in the hidden layers and the linear activation in the output layer.

For a further history of deep learning see [51] or for a more statistics oriented overview of machine learning see [150].

1.3 Introduction for Chapter 2: Classification

The current popularity of deep learning is in part caused by the performance of deep neural networks for image recognition tasks: identifying what object is depicted in an image. In 2012, a deep neural network [80] became state-of-the-art by outperforming other methods in the ImageNet Large Scale Visual Recognition Challenge.

In statistical terms, image recognition is an example of a supervised classification problem. In the classification model with K classes, we observe n random pairs $(\mathbf{X}_1, \mathbf{Y}_1), (\mathbf{X}_2, \mathbf{Y}_2), \ldots, (\mathbf{X}_n, \mathbf{Y}_n)$, with $\mathbf{Y}_i$ the encoding of the observed label or class



(a) Linear

(b) Non-linear

Figure 1.11: Plots of the samples (blue), Deep Neural Network estimator (red) and the true regression function (black) of a linear and a non-linear model.

of the input $\mathbf{X}_i$. For more than two classes, it is standard to represent the labels with one-hot encoding: Each $\mathbf{Y}_i$ is a K-dimensional vector consisting of exactly one 1 and all other coefficients are set to zero. The relationship between $\mathbf{X}_i$ and $\mathbf{Y}_i$ is determined by the (unknown) conditional class probabilities

$$\mathbb{P}(Y_{i,k} = 1|\mathbf{X}_i = \mathbf{x}),$$

where $Y_{i,k}$ is the k-th coefficient of $\mathbf{Y}_i$ and $\mathbf{x}$ is a specific value taken by the input $\mathbf{X}_i$. In classification, the goal is to predict the class label of a new input. In machine learning, classification methods are often compared based on the fraction of correct classifications. This is equal to one minus the risk corresponding to the 0-1 loss.

To build a classifier, one can try to estimate the decision boundaries directly. Alternatively, one can first estimate the conditional class probabilities and then plug these estimates into a decision rule. For an overview of classification methods see for example [38].

Deep neural networks output estimates of the conditional class probabilities, see for example the seminal work [80]. Furthermore, neural networks are typically compared to other methods based on the faction of correct labels that are contained in the (top five or ten) most likely predicted labels. The activation function commonly used in the output layer of these neural networks is the softmax function

$$\mathbf{\Phi}(\mathbf{x}) := \left(\frac{e^{x_1}}{\sum_{j=1}^K e^{x_j}}, \dots, \frac{e^{x_K}}{\sum_{j=1}^K e^{x_j}}\right) : \mathbb{R}^K \to \mathcal{S}^K,$$

where $\mathcal{S}^K = \big\{(p_1, \ldots, p_K : \sum_{j=1}^K p_j = 1, p_j \geq 0\big\}$ denotes the probability simplex in $\mathbb{R}^K$. The softmax function is a multivariate version of the logistic function in Figure 1.9b and guarantees that the output of the neural network is a probability vector.

Because of the non-differentiability, gradient descent cannot be applied to the 0-1 loss. Therefore, neural networks are trained using a surrogate loss [9, 140]. The cross-entropy loss is commonly used in combination with the softmax output:

$$\ell(\widehat{p}(\mathbf{X}_i), \mathbf{Y}_i) := -\sum_{k=1}^K Y_{i,k} \log(\widehat{p}_k(\mathbf{X}_i)).$$

The cross-entropy loss can be derived from the log-likelihood of the conditional class probabilities, see Section 2.2. Chapter 2 focuses on estimating the conditional class probabilities instead of the final classification. Therefore, the risk corresponding to the cross-entropy loss is considered instead of the risk corresponding to the 0-1 loss. Convergence rates for a neural network estimator are derived with respect to a truncated version of the risk corresponding to the cross-entropy loss.

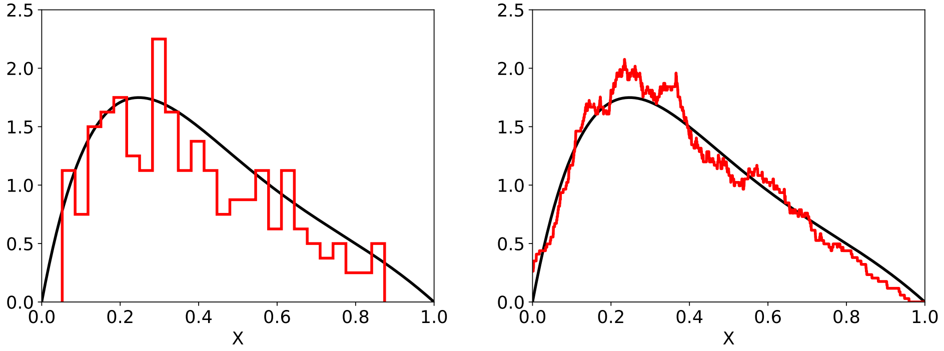
1.4 Introduction for Chapter 3: Multivariate density estimation

In multivariate density estimation, we observe a sequence of independent random vectors $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ distributed according to some multivariate density f_0. The density estimation problem is to estimate this unknown density f_0 from the data. Unlike regression and classification, there are no observed response variables Y_i, making this problem unsupervised.

Kernel density estimators are standard methods for nonparametric density estimation. A kernel in d-dimensions is a function $K : \mathbb{R}^d \to \mathbb{R}$ such that $\int_{\mathbb{R}} K(\mathbf{u})\, d\mathbf{u} = 1$. The kernel density estimator $\widehat{f}_K$ with kernel K and bandwidth h is given by:

$$\widehat{f}_K := \frac{1}{nh^d} \sum_{i=1}^n K\bigg(\frac{1}{h}\Big(\mathbf{X}_i - \mathbf{x}\Big)\bigg). \tag{1.4.1}$$

Kernel density estimators are related to histograms. For the rectangular kernel, Figure 1.2a, the corresponding kernel density estimator can be derived as an average of an infinite number of histograms with shifted bin centers. Alternatively, one can derive this estimator from an approximation of the derivative of the cumulative distribution function (cdf), an approach that was taken in early work on kernel density estimation, [123, 108]. Figure 1.12 compares histogram and kernel density estimator with rectangular kernel.



(a) Histogram

(b) Kernel density estimator

Figure 1.12: Plots, based on the same sample, of the histogram estimator (red) on the left and the kernel density estimator with rectangular kernel (red) on the right. The true density function (black) is a mixture of two beta-distributions.

In Chapter 3 we transform the unsupervised density estimation problem into a supervised regression problem. We first use a kernel density estimator to generate response variables. This allows us to fit a deep ReLU network using existing results for regression. We derive convergence rates showing that this approach can use compositional structures to partly circumvent the curse of dimensionality. We also provide an exploratory simulation study applying this method to several structural density models.

1.5 Introduction for Chapter 4: Optimization motivated by biological neural networks

From the beginning, artificial neural networks have been influenced by theories about biological neural networks (the brain). The first artificial network, the perceptron [121, 122] was based on the McCulloch-Pitts model [95] for neurons in the brain [51, 150]. The firing pattern of neurons in the brain has been cited as motivation for the ReLU activation function, Figure 1.9d, [50]. However, the main interest in ReLU activation functions originates from the empirically observed performance improvement compared to the previously used sigmoid activation functions [50, 102]. Importantly, artificial neural networks are not intended to represent learning in the brain. Popular and successful methods for artificial neural networks, such as gradient descent and backpropagation, are even implausible for biological neural networks [33, 89, 142]. One issue is that these training methods require the capacity to share information about all the parameters with the entire neural network, also known as the weight transportation problem.

Recently there has been renewed interest in the differences and similarities between artificial and biological neural networks, [89, 128, 155]. For example, alternatives to gradient descent have been proposed that are more biologically plausible. One of them is known as (weight-perturbed) forward gradient descent [13, 117]. In this method the gradient update step (1.2.1) is replaced by the update step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \alpha_k \Big(\nabla_{\boldsymbol{\theta}_{k-1}} \ell\big(f_{\boldsymbol{\theta}_{k-1}}(\mathbf{X}_k), Y_k\big)\Big)^\top \boldsymbol{\xi}_k \boldsymbol{\xi}_k,$$

where $\boldsymbol{\xi}_k$ is distributed as $\mathcal{N}(0, \mathbf{I}_d)$ and is independent of all other randomness. Thus, only random linear combinations of the gradient are required instead of the full gradient. In Chapter 4, we study forward gradient descent in the framework of the linear regression model. We prove that in this setting the mean squared error converges with a rate $d^2 \log(d)/k$, for a large enough number of samples k. This rate has an additional dimension factor $d \log(d)$ compared to the optimal rate for linear regression [144, 63, 98].

Chapter 2

Convergence rates of deep ReLU networks for multiclass classification

Abstract

For classification problems, trained deep neural networks return probabilities of class memberships. In this chapter we study convergence of the learned probabilities to the true conditional class probabilities. More specifically we consider sparse deep ReLU network reconstructions minimizing cross-entropy loss in the multiclass classification setup. Interesting phenomena occur when the class membership probabilities are close to zero. Convergence rates are derived that depend on the near-zero behaviour via a margin-type condition.

2.1 Introduction

The classification performance of a procedure is often evaluated by considering the percentage of test samples that is assigned to the correct class. The corresponding loss for this performance criterion is called the 0-1 loss. Theoretical results for this loss are often related to the the margin condition [94, 145, 5], which allows for fast convergence rates. Empirical risk minimization with respect to the non-convex 0-1 loss is computationally hard and convex surrogate losses are used instead, see for example [9, 140]. More recently, similar results have been obtained for deep neural networks in the binary classification setting. This includes results for standard deep neural networks in combination with the hinge and logistic loss as surrogate losses [67], as

This chapter is based on: Thijs Bos and Johannes Schmidt-Hieber (2022). Convergence rates of deep ReLU networks for multiclass classification. In Electronic Journal of Statistics ***16****, 2724-2773. The research has been supported by the NWO/STAR grant 613.009.034b and the NWO Vidi grant VI.Vidi.192.021.*

well as results for deep convolutional neural networks with the least squares loss [74] and logistic loss [76] as surrogate losses. More details can be found in the discussion following Theorem 2.3.3.

Trained neural networks provide more information than just a guess of the class membership. For each class and each input, they return an estimate for the probability that the true label is in this class. For an illustration, see for example Figure 4 in the seminal work [80]. In applications it is often important how certain a network is about class memberships, especially in safety-critical systems where a wrong decision can have serious consequences such as automated driving [22] and AI based disease detection [87, 52]. In fact, the conditional class probabilities provide us with a notion of confidence. If the probability of the largest class is nearly one, it is likely that this class is indeed the true one. On the other hand, if there is no clear largest class and the conditional class probabilities of several classes are close to each other, it might be advisable to let a human examine the case instead of basing the decision only on the outcome of the algorithm.

To evaluate how fast the estimated conditional class probabilities of deep ReLU networks approach the true conditional class probabilities, we consider in this chapter convergence with respect to the cross-entropy (CE) loss. If the conditional class probabilities are bounded away from zero or one, the problem is related to regression and density estimation. Therefore, it seems that one could simply modify the existing proofs on convergence rates for deep ReLU networks in the regression context under the least squares loss [127, 11]. This does, however, not work since the behaviour of the CE loss differs fundamentally from that of the least squares loss for small conditional class probabilities. The risk associated with the CE loss is the expectation with respect to the input distribution of the Kullback-Leibler divergence of the conditional class probabilities. If an estimator becomes zero for one of the conditional class probabilities while the underlying conditional class probability is positive, the risk can even become infinite, see Section 2.2. In many applications where deep learning is state-of-the-art, the covariates contain nearly all information about the label and hence the conditional class probabilities are close to zero or one. For example in image classification it is often clear which object is shown on a picture. To deal with the behaviour near zero, we introduce a truncation of the CE loss function. This allows us to obtain convergence rates without bounding either the true underlying conditional class probabilities or the estimators away from zero. Instead our rates depend on an index quantifying the behaviour of the conditional class probabilities near zero. Convergence rates and the condition on the conditional class probabilities can be found in Section 2.3.

Notation: We denote vectors and vector valued functions by bold letters. For two vector valued functions $\mathbf{f} = (f_1, \ldots, f_d)$ and $\mathbf{g} = (g_1, \ldots, g_d)$ mapping $\mathcal{D}$ to $\mathbb{R}^d$, we set $\|\mathbf{f} - \mathbf{g}\|_{\mathcal{D},\infty} := \big\| \max_{j=1,\ldots,d} |f_j(\mathbf{x}) - g_j(\mathbf{x})| \big\|_{L^\infty(\mathcal{D})}$. If it is clear to which domain $\mathcal{D}$ we refer to, we also simply write $\|\mathbf{f}-\mathbf{g}\|_\infty$. For a vector $\mathbf{v}=(v_1,\ldots,v_m)$ and a matrix $W=(W_{i,j})_{i=1,\ldots,n;j=1,\ldots,m}$ we define the maximum entry norms as $\|\mathbf{v}\|_\infty:=\max_{i=1,\ldots,m}|v_i|$ and $\|W\|_\infty:=\max_{i=1,\ldots,n}\max_{j=1,\ldots,m}|W_{i,j}|$. The counting 'norm' $\|\mathbf{v}\|_0$, $\|W\|_0$ is the number of nonzero entries in the vector $\mathbf{v}$ and matrix W, respectively. For a vector $\mathbf{v}=(v_1,\ldots,v_r)^\top$ and g a univariate function, we write $g(\mathbf{v}):=(g(v_1),\ldots,g(v_r))^\top$. We often apply this to the activation function or the logarithm $g(u)=\log(u)$. Similarly, we define for two vectors of the same length $\mathbf{v},\mathbf{v}'$, $\log(\mathbf{v}/\mathbf{v}')=\log(\mathbf{v})-\log(\mathbf{v}')$. For any natural number γ, we set $0\log^\gamma(0):=0$. For a real number $x\in\mathbb{R}$, $\lfloor x\rfloor$ is the largest integer strictly smaller than x and $\lceil x\rceil$ is the smallest integer $\geq x$. A K-dimensional standard basis vector is a vector of length K that can be written as $(0,\ldots,0,1,0,\ldots,0)^\top$. We use $\mathcal{S}^K$ to denote the $(K-1)$-simplex in $\mathbb{R}^K$, that is, $\mathcal{S}^K=\{\mathbf{v}\in\mathbb{R}^K:\sum_{k=1}^K v_k=1, v_k\geq 0, k=1,\ldots,K\}$. For two probability measures P and Q, the Kullback-Leibler divergence $\mathrm{KL}(P,Q)$ is defined as $\mathrm{KL}(P,Q):=\int\log(dP/dQ)\,dP$ if P is dominated by Q and as $\mathrm{KL}(P,Q):=\infty$ otherwise.

2.2 The multiclass classification model

In multiclass classification with $K\geq 2$ classes and design on $[0,1]^d$, we observe a dataset $\mathcal{D}_n=\big\{(\mathbf{X}_i,\mathbf{Y}_i):i=1,\ldots,n\big\}$ of n i.i.d. copies of pairs $(\mathbf{X},\mathbf{Y})$ with design/input vector $\mathbf{X}$ taking values in $[0,1]^d$ and the corresponding response vector $\mathbf{Y}$ being one of the K-dimensional standard basis vectors. The response decodes the label of the class: the output $\mathbf{Y}$ is the k-th standard basis vector if the label of the k-th class is observed. As a special case, for binary classification the output is decoded as $(1,0)^T$ if the first class is observed and as $(0,1)^T$ if the second class is observed. We write $\mathbb{P}$ for the joint distribution of the random vector $(\mathbf{X},\mathbf{Y})$ and $\mathbb{P}_{\mathbf{X}}$ for the marginal distribution of $\mathbf{X}$. The conditional probability $\mathbb{P}_{\mathbf{Y}|\mathbf{X}}$ exists since $\mathbf{Y}$ is supported on finitely many points.

An alternative model is to assume that each of the K classes is observed roughly n/K times. To derive statistical risk bounds, there is hardly any difference and the fact that the i.i.d. model generates with small probability highly unbalanced designs will not change the analysis.

The task is now to estimate/learn from the dataset $\mathcal{D}_n$ the probability that a new input vector $\mathbf{X}$ is in class k. If $\mathbf{Y}=(Y_1,\ldots,Y_K)^\top$, the true conditional class probabilities are

$$p_k^0(\mathbf{x}):=\mathbb{P}(Y_k=1|\mathbf{X}=\mathbf{x}),\quad k=1,\ldots,K.$$

For any $\mathbf{x}$ this gives a probability vector, that is, $\sum_{k=1}^K p_k^0(\mathbf{x})=1$. For notational convenience, we also define the vector of conditional class probabilities $\mathbf{p}_0(\mathbf{x}):=(p_1^0(\mathbf{x}),\cdots,p_K^0(\mathbf{x}))^\top$.

To learn the conditional class probabilities from data, the commonly employed strategy in deep learning is to minimize the log-likelihood over the free parameters of a deep neural network using (stochastic) gradient descent. The likelihood for the conditional class probability vector $\mathbf{p}(\mathbf{x}) := (p_1(\mathbf{x}), \cdots, p_K(\mathbf{x}))^\top$ is given by

$$\mathcal{L}(\mathbf{p}|\mathcal{D}_n) = \prod_{i=1}^{n} \prod_{k=1}^{K} (p_k(\mathbf{X}_i))^{Y_{ik}},$$

with Y_{ik} the k-th entry of $\mathbf{Y}_i$. The negative log-likelihood or cross-entropy loss is then

$$\mathbf{p} \mapsto \ell\big(\mathbf{p}, \mathcal{D}_n\big) := -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} Y_{ik} \log(p_k(\mathbf{X}_i)) = -\frac{1}{n} \sum_{i=1}^{n} \mathbf{Y}_i^\top \log\big(\mathbf{p}(\mathbf{X}_i)\big), \tag{2.2.1}$$

where the logarithm in the last expression is taken component-wise as explained in the notation section above and $\mathbf{Y}^T \log(\mathbf{p}(\mathbf{X}_i))$ is understood as the scalar product of the vectors $\mathbf{Y}$ and $\log(\mathbf{p}(\mathbf{X}_i))$. The response vectors $\mathbf{Y}_i$ are standard basis vectors and in particular have nonnegative entries. The cross-entropy loss is thus always nonnegative and consequently defines indeed a proper statistical loss function. The cross-entropy loss is also convex, but not strictly convex and thus also not strongly convex, see [149], Chapter III-B for a proof. For binary classification ($K = 2$), the cross-entropy loss coincides with the logistic loss. Throughout this chapter, we consider estimators/learners $\widehat{\mathbf{p}}(\mathbf{X})$ with the property that $\widehat{\mathbf{p}}(\mathbf{x})$ is a probability vector for all $\mathbf{x}$, or equivalently, $\widehat{\mathbf{p}}(\mathbf{x})$ lies in the simplex $\mathcal{S}^K$ for all $\mathbf{x}$. This is in particular true for neural networks with softmax activation function in the output layer. Recall that $\mathbf{p}_0(\mathbf{x})$ is the vector of true class probabilities. If $(\mathbf{X}, \mathbf{Y})$ has the same distribution as each of the observations and is independent of the dataset $\mathcal{D}_n$, the statistical estimation risk associated with the CE loss is

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n,(\mathbf{X},\mathbf{Y})}\left[\mathbf{Y}^\top \log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\widehat{\mathbf{p}}(\mathbf{X})}\right)\right] &= \mathbb{E}_{\mathcal{D}_n,\mathbf{X}}\left[\mathbf{p}_0(\mathbf{X})^\top \log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\widehat{\mathbf{p}}(\mathbf{X})}\right)\right] \\ &= \mathbb{E}_{\mathcal{D}_n,\mathbf{X}}\big[\mathrm{KL}\left(\mathbf{p}_0(\mathbf{X}), \widehat{\mathbf{p}}(\mathbf{X})\right)\big],\end{aligned}$$

where the first equality follows from conditioning on the design vector $\mathbf{X}$ and $\mathrm{KL}(\mathbf{p}_0(\mathbf{X}), \widehat{\mathbf{p}}(\mathbf{X}))$ is understood as the Kullback-Leibler divergence of the discrete distributions with probability mass functions $\mathbf{p}_0(\mathbf{X})|\mathbf{X}$ and $\widehat{\mathbf{p}}(\mathbf{X})|(\mathbf{X}, \mathcal{D}_n)$.

(Stochastic) gradient descent methods aim to minimize the CE loss (2.2.1) over a function class $\mathcal{F}$ induced by the method. In the context of neural networks, this class is generated by all network functions with a pre-specified network architecture. In particular, the class is parametrized through the network parameters. The maximum likelihood estimator (MLE) is by definition any global minimizer of (2.2.1). For some function classes the MLE can be given explicitly. In the extreme case that $\mathbf{x} \mapsto \mathbf{p}(\mathbf{x})$ is constraint to constant functions, the problem is equivalent to estimation of the probability vector of a multinomial distribution and the MLE is the average $\widehat{\mathbf{p}}^{\mathrm{MLE}} = \frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i$. The other extreme is the case of training error zero. If the observed design vectors are all different, training error zero is achieved whenever there exists $\mathbf{p} \in \mathcal{F}$ such that $\mathbf{Y}_i = \mathbf{p}(\mathbf{X}_i)$ for all $i = 1, \ldots, n$. This follows from $0\log(0) = 1\log(1) = 0$. To achieve training error zero, we therefore need to interpolate all data points. Notice that misclassification error zero does not necessarily require interpolation of the data points.

Already for small function classes, the MLE has infinite risk if the statistical risk is as defined above. The next lemma makes this precise.

Lemma 2.2.1. *Consider binary classification ($K = 2$) with uniform design $\mathbf{X} \sim \mathrm{Unif}([0,1]^d)$ and $\mathbf{p}_0(\mathbf{x}) := (1/2, 1/2)^\top$ for all $\mathbf{x} \in [0,1]^d$. Suppose that the function class $\mathcal{F}$ contains an element $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}))^\top$ such that $p_1(\mathbf{x}) = 0$ for all $\mathbf{x} \in [0, 1/3]^d$ and $p_1(\mathbf{x}) = 1$ for all $\mathbf{x} \in [2/3, 1]^d$. Then, there exists a MLE $\widehat{\mathbf{p}}$ with*

$$\mathbb{E}_{\mathcal{D}_n,\mathbf{X}}\left[\mathbf{p}_0(\mathbf{X})^\top \log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\widehat{\mathbf{p}}(\mathbf{X})}\right)\right] = \infty.$$

The assumption on the function class $\mathcal{F}$ in the previous statement is quite weak and is satisfied if $\mathcal{F}$ contains all piecewise constant conditional class probabilities with at most two pieces or all piecewise linear conditional class probabilities with at most three pieces. A large statistical risk occurs also in the case of zero training error or if the estimator $\widehat{\mathbf{p}}$ severely underestimates the true probabilities.

To overcome the shortcomings of the Kullback-Leibler risk, one possibility is to regularize the Kullback-Leibler divergence and to consider for some $B > 0$ the truncated Kullback-Leibler risk

$$R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) := \mathbb{E}_{\mathcal{D}_n,\mathbf{X}}\Big[\mathrm{KL}_B\left(\mathbf{p}_0(\mathbf{X}), \widehat{\mathbf{p}}(\mathbf{X})\right)\Big],$$

where

$$\mathrm{KL}_B\left(\mathbf{p}_0(\mathbf{X}), \widehat{\mathbf{p}}(\mathbf{X})\right) := \mathbf{p}_0(\mathbf{X})^\top \left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\widehat{\mathbf{p}}(\mathbf{X})}\right)\right).$$

The loss can be shown to be nonnegative whenever $B \geq 2$, see Lemma 2.3.4 below. The threshold B becomes void if the estimator $\widehat{\mathbf{p}}$ is constrained to be in $[e^{-B}, 1]^K$. If the estimator underestimates one of the true conditional class probabilities by a large factor, the logarithm becomes large and the threshold B kicks in. For $B = \infty$, we recover the Kullback-Leibler risk.

The idea of truncation is not new. [158] truncates the log-likelihood ratio to avoid problems with this ratio becoming infinite. Their risk rates, however, are in terms of the Hellinger distance and the truncation does not appear in the statement of their results. For the truncated Kullback-Leibler risk the truncation plays a much more prominent role and appears as a multiplicative factor in the risk bounds. Lemma 2.3.4 provides insight in this difference: it shows that any upper bound for any B-truncated Kullback-Leibler divergence with $B \geq 2$ provides an upper bound for the Hellinger distance.

As we are interested in the multiclass classification problem in the context of neural networks, the function class $\mathcal{F}$ is not convex. Due to this non-convexity, the training of neural networks does typically not yield a neural network achieving the global minimum. We therefore do not assume that the estimator is the MLE and use a parameter to quantify the difference between the achieved empirical risk and the global minimum: For any estimator $\widehat{\mathbf{p}}$ taking values in a function class $\mathcal{F}$, we denote the difference between $\widehat{\mathbf{p}}$ and the global minimum of the empirical risk over that entire class by

$$\Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}}) := \mathbb{E}_{\mathcal{D}_n}\Big[-\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i^\top \log(\widehat{\mathbf{p}}(\mathbf{X}_i)) - \min_{\mathbf{p}\in\mathcal{F}} -\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i^\top \log(\mathbf{p}(\mathbf{X}_i))\Big]. \quad (2.2.2)$$

2.2.1 Deep ReLU networks

In this chapter we study deep ReLU networks with softmax output layer. Recall that the rectified linear unit (ReLU) activation function is $\sigma(x) := \max\{x, 0\}$. For any vectors $\mathbf{v} = (v_1, \cdots, v_r)^\top, \mathbf{y} = (y_1, \cdots, y_r)^\top \in \mathbb{R}^r$, write $\sigma_{\mathbf{v}}\mathbf{y} := (\sigma(y_1 - v_1), \ldots, \sigma(y_r - v_r))^\top$. To ensure that the output of the network is a probability vector over the K classes, it is standard to apply the softmax function

$$\mathbf{\Phi} = \left(\frac{e^{x_1}}{\sum_{j=1}^K e^{x_j}}, \ldots, \frac{e^{x_K}}{\sum_{j=1}^K e^{x_j}}\right) : \mathbb{R}^K \to \mathcal{S}^K$$

in the last layer. We use L to denote the number of hidden layers or depth of the neural network, and $\mathbf{m} = (m_0, \cdots, m_{L+1}) \in \mathbb{N}^{L+2}$ to denote the widths, that is, the number of nodes in each layer of the network. A (ReLU) network architecture with output function $\boldsymbol{\psi} : \mathbb{R}^{m_{L+1}} \to \mathbb{R}^{m_{L+1}}$ is a pair $(L, \mathbf{m})_{\boldsymbol{\psi}}$ and a network with network architecture $(L, \mathbf{m})_{\boldsymbol{\psi}}$ is any function of the form

$$\mathbf{f} : \mathbb{R}^{m_0} \to \mathbb{R}^{m_{L+1}}, \quad \mathbf{x} \mapsto \mathbf{f}(\mathbf{x}) = \boldsymbol{\psi} W_L \sigma_{\mathbf{v}_L} W_{L-1} \sigma_{\mathbf{v}_{L-1}} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}, \quad (2.2.3)$$

where W_j is a $m_j \times m_{j+1}$ weight matrix and $\mathbf{v}_j \in \mathbb{R}^{m_j}$ is a shift vector. Throughout this chapter we use the convention that $\mathbf{v}_0 := (0, \ldots, 0)^\top \in \mathbb{R}^{m_0}$.

First we define neural network classes with the additional property that all network parameters are bounded in absolute value by one via

$$\mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m}) := \left\{ \mathbf{f} \text{ is of the form of (2.2.3)} : \max_{j \in \{0, \cdots, L\}} (\|W_j\|_\infty \vee \|\mathbf{v}_j\|_\infty) \leq 1 \right\},$$

with the maximum entry norm $\| \cdot \|_\infty$ as defined in the notation section above. As in previous work, we study estimation over s-sparse ReLU networks. Those are function classes of the form

$$\mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m}, s) := \left\{ \mathbf{f} \in \mathcal{F}(L, \mathbf{m}) : \sum_{j=0}^{L} \|W_j\|_0 + \|\mathbf{v}_j\|_0 \leq s \right\},$$

where the counting norm $\| \cdot \|_0$ denotes the number of nonzero vector/matrix entries.

All neural network classes in this chapter have either softmax output activation $\boldsymbol{\psi} = \boldsymbol{\Phi}$ or identity output activation $\boldsymbol{\psi} = \mathrm{id}$.

2.3 Main Results

Interesting phenomena occur if the conditional class probabilities are close to zero or one. We now introduce a notion measuring the size of the set on which the conditional class probabilities are small. The index α will later appear in the convergence rate.

Definition 2.3.1. (Small Value Bound) Let $\alpha \geq 0$ and $\mathcal{H}$ be a function class. We say that $\mathcal{H}$ is α-small value bounded (or α-SVB) if there exists a constant $C > 0$, such that for all $\mathbf{p} = (p_1, \ldots, p_K) \in \mathcal{H}$ it holds that

$$\mathbb{P}_{\mathbf{X}}(p_k(\mathbf{X}) \leq t) \leq Ct^\alpha, \quad \text{for all } t \in (0, 1] \text{ and all } k \in \{1, \ldots, K\}.$$

The condition always holds for $\alpha = 0$ and $C = 1$. If $\mathbb{P}_{\mathbf{X}}(p_k(\mathbf{X}) = 0) > 0$, the condition does not hold for $\alpha > 0$. If all functions in a class are lower bounded by a constant B_0, the class is α-SVB for any α with constant $C = B_0^{-\alpha}$. More generally, the index α is completely determined by the behaviour near zero: If for some function class there exists some $0 < \tau \ll 1$, so that the bound holds for α and for all $t \in (0, \tau]$, then replacing C by $C' = \max\{C, \tau^{-\alpha}\}$ guarantees that $C'\tau^\alpha \geq 1$, which in turn implies that the function class is α-SVB. Moreover, if a function class is α-SVB, then it is also α^*-SVB for all $\alpha^* \leq \alpha$. This follows immediately by noticing that $t^{\alpha^*} \geq t^\alpha$ for all $t \in (0, 1]$. Increasing the index makes the small value bound condition thus more restrictive.

To show that the definition of the small value bound makes sense, we have to check that for any $\alpha > 0$, there exist conditional class probabilities that are α-SVB for that

α, but are not α^*-SVB for any larger $\alpha^* > \alpha$. To see this, consider the case that X is uniformly distributed on $[0,1]$, and that there are three classes $K = 3$. For given $\alpha > 0$, define the function $\mathbf{p}_\alpha : [0,1] \to \mathcal{S}^3$ as $p_1(x) = \min\{x^{1/\alpha}, 1/3\}$, $p_2(x) = 1/3$ and $p_3(x) = 1 - p_1(x) - p_2(x) = 2/3 - \min\{x^{1/\alpha}, 1/3\}$. Since $p_2(x), p_3(x) \geq 1/3$, we have for $k = 2,3$ that $\mathbb{P}_X(p_k(X) \leq t) \leq (3t)^\alpha$. When $k = 1$, it holds for $t \leq 1/3$ that $\mathbb{P}_X(p_1(X) \leq t) = \mathbb{P}_X(X^{1/\alpha} \leq t) = \mathbb{P}_X(X \leq t^\alpha) = t^\alpha$. Hence $\mathbb{P}_X(p_k(X) \leq t) \leq (3t)^\alpha$ for $k = 1,2,3$, so $\mathbf{p}_\alpha$ is α-SVB with constant 3^α. Now we show that this function is not α^*-SVB for any $\alpha^* > \alpha$. Let $\alpha^* > \alpha$, then for every constant $C > 0$, there exists a $\tau_C \in (0, 1/3)$ such that $C(\tau_C)^{\alpha^*} < (\tau_C)^\alpha = \mathbb{P}_X(p_1(X) \leq \tau_C)$. Since C is arbitrary, $\mathbf{p}_\alpha$ is not α^*-SVB.

The following example provides some insights into the relation between the conditional class probabilities and the distribution of $\mathbf{X}$. Consider the binary case $K = 2$, with input domain $[0,1]^2$, $p_1(\mathbf{x}) = (3|x_1 + x_2 - 1|^8)/4$, and $p_2(\mathbf{x}) = 1 - p_1(\mathbf{x})$, see Figure 2.1. Observe that $0 \leq p_1(\mathbf{x}) \leq 3/4$ for all $\mathbf{x} \in [0,1]^2$, so $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ indeed define conditional class probabilities. Furthermore, $p_2(\mathbf{x}) \geq 1/4$, in other words, $p_2(\mathbf{x})$ is bounded away from zero. Thus, to determine the SVB index α, it remains to consider $p_1(\mathbf{x})$. If $\mathbf{X}$ is the uniform distribution on $[0,1]^2$, Proposition 2.C.7 tells us that

$$\mathbb{P}_{\mathbf{X}}\left(p_1(\mathbf{X}) \leq t\right) = 2\left(\frac{4t}{3}\right)^{\frac{1}{8}} - \left(\frac{4t}{3}\right)^{\frac{1}{4}}$$

and hence the small value bound is satisfied for α at most $1/8$. Now suppose that instead of the uniform design, the distribution of $\mathbf{X}$ is given by the density $(x_1, x_2) \mapsto 3|x_1 + x_2 - 1|$, see Figure 2.1 for a plot. Thus, the design density is zero if $p_1(\mathbf{x})$ is zero. In this case, Proposition 2.C.7 gives

$$\mathbb{P}_{\mathbf{X}}\left(p_1(\mathbf{X}) \leq t\right) = 3\left(\frac{4t}{3}\right)^{\frac{1}{4}} - 2\left(\frac{4t}{3}\right)^{\frac{3}{8}},$$

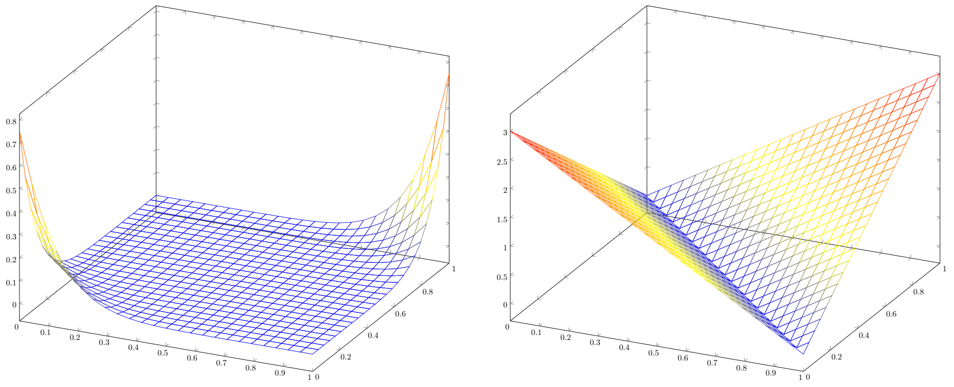
and the SVB index α is at most $1/4$.

The following theorem shows the influence of the index α in the small value bound on the approximation rates.

Theorem 2.3.2. *If the function class is α-SVB with constant C, then, for any approximating function $\mathbf{p} = (p_1, \ldots, p_k) : [0,1]^d \to \mathcal{S}^K$ satisfying $\|\mathbf{p} - \mathbf{p}_0\|_\infty \leq C_1/M$, and $\min_k \inf_{\mathbf{x}\in[0,1]^d} p_k(\mathbf{x}) \geq 1/M$, for some constant C_1, it holds that*

$$\mathbb{E}_{\mathbf{X}}\left[(\mathbf{p}_0(\mathbf{X}))^\top \log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\mathbf{p}(\mathbf{X})}\right)\right] \leq CK\frac{(C_1+1)^{2+(\alpha\wedge 1)}}{M^{1+(\alpha\wedge 1)}}\Big(1 + \frac{\mathbb{1}_{\{\alpha<1\}}}{1-\alpha} + \log(M)\Big).$$

The proof for this result bounds the Kullback-Leibler divergence by the χ^2-divergence and then distinguishes the cases where the conditional class probabilities are



(a) Conditional class probability

(b) Density

Figure 2.1: Plot of the conditional class probability $p_1(\mathbf{x}) = (3|x_1 + x_2 - 1|^8)/4$ on the left and of the density $(x_1, x_2) \mapsto 3|x_1 + x_2 - 1|$ on the right.

smaller and larger than $1/M$. Both terms can be controlled via the α-SVB condition. The convergence rate becomes faster in M up to $\alpha = 1$ and is $\log(M)/M^2$ for all $\alpha \geq 1$.

The small value bound provides a flexible framework that allows the conditional class probabilities to be close to zero and therefore generalizes the standard assumption in the nonparametric classification literature that the conditional class probabilities are bounded away from zero. Here, we argue that the regime of small conditional class probabilities is of particular relevance for classification tasks where most of the information about the class label is contained in the covariates. Indeed, if $\mathbf{X}$ contains all information about the class label Y, then $Y|\mathbf{X}$ is deterministic and the conditional class probability is either zero or one. On the contrary, in situations where the covariates/input variable $\mathbf{X}$ does not contain the full information about the class label, $Y|\mathbf{X}$ is random, and the conditional class probabilities are bounded away from zero or one. The case of small conditional class probabilities corresponds to a scenario where the covariates contain most of the information about the class label. These are classification tasks for which small misclassification errors can be achieved, but perfect classification is impossible. This is also the regime for which the SVB index α should be strictly larger than zero. For instance, for the widely used Breast Cancer Wisconsin (Diagnostic) dataset and Heart Disease dataset from the UCI machine learning repository [40] the covariates do not contain the full relevant information about the disease but small misclassification can be achieved. It is therefore conceivable that these are prototypical examples for the case $\alpha > 0$.

The small value bound has a similar flavor as Tsybakov's margin condition, which can be stated as $\mathbb{P}_{\mathbf{X}}(0 < |p_0(\mathbf{X}) - 1/2| \leq t) \leq Ct^{\gamma}$ for binary classification [5]. The margin condition provides a control on the number of data points that are close to the decision boundary $\{\mathbf{x} : p_0(\mathbf{x}) = 1/2\}$ and that are therefore hard to classify correctly. Differently speaking, the problem becomes easier if the conditional class probabilities are either close to zero or one. This is in contrast with the small value bound, which will lead to faster convergence rates when the true conditional class probabilities are mostly away from zero. This difference is due to the loss: the 0-1 loss only cares about predicting the class membership, while the CE loss measures how well the conditional class probabilities are estimated and puts additional emphasis on small conditional class probabilities by considering the ratio between prediction and truth.

To obtain estimation rates, we further assume that the underlying true conditional class probability function $\mathbf{p}_0$ belongs to the class of Hölder-smooth functions. For $\beta > 0$ and $D \subset \mathbb{R}^m$, the ball of β-Hölder functions with radius Q is defined as

$$C^{\beta}(D,Q) := \Bigg\{ f : D \to \mathbb{R} : \\ \sum_{\boldsymbol{\gamma}:\|\boldsymbol{\gamma}\|_1 < \beta} \|\partial^{\boldsymbol{\gamma}} f\|_{\infty} + \sum_{\boldsymbol{\gamma}:\|\boldsymbol{\gamma}\|_1 = \lfloor \beta \rfloor} \sup_{\mathbf{x},\mathbf{y} \in D, \mathbf{x} \neq \mathbf{y}} \frac{|\partial^{\boldsymbol{\gamma}} f(\mathbf{x}) - \partial^{\boldsymbol{\gamma}} f(\mathbf{y})|}{\|\mathbf{x} - \mathbf{y}\|_{\infty}^{\beta - \lfloor \beta \rfloor}} \leq Q \Bigg\},$$

where $\partial^{\boldsymbol{\gamma}} = \partial^{\gamma_1} \ldots \partial^{\gamma_m}$, with $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_m) \in \mathbb{N}^m$. The function class $\mathcal{G}(\beta, Q)$ of β-smooth conditional class probabilities is then defined as

$$\mathcal{G}(\beta, Q) = \Big\{ \mathbf{p} = (p_1, \cdots, p_K)^{\top} : [0,1]^d \to \mathcal{S}^K : \\ p_k \in C^{\beta}([0,1]^d, Q), k = 1, \ldots, K \Big\}.$$

If $Q < 1/K$, then, $\|\mathbf{p}\|_{\infty} \leq Q$ implies $\sum_{k=1}^{K} p_k \leq KQ < 1$, so we need Hölder radius $Q \geq 1/K$ for this class to be non-empty. Combining the smoothness and the small value bound, we write $\mathcal{G}_{\alpha}(\beta, Q) = \mathcal{G}_{\alpha}(\beta, Q, C)$ for all functions in $\mathcal{G}(\beta, Q)$ that satisfy the α-SVB condition with constant C. For large enough radius Q and constant C, the class $\mathcal{G}_{\alpha}(\beta, Q)$ is non-empty. For example, the constant function $\mathbf{p} = (1/K, \ldots, 1/K)$ is in $\mathcal{G}_{\alpha}(\beta, Q)$ for any $\beta > 0$ and $\alpha > 0$ when $Q \geq 1/K$ and $C \geq K^{\alpha}$.

For $0 \leq \alpha \leq 1$ the index from the SVB condition and β the smoothness index, we introduce the rate

$$\phi_n = K^{\frac{(1+\alpha)\beta + (3+\alpha)d}{(1+\alpha)\beta + d}} n^{-\frac{(1+\alpha)\beta}{(1+\alpha)\beta + d}}.$$

Theorem 2.3.3 (Main Risk Bound)**.** *Consider the multiclass classification model with* $\mathbf{p}_0 \in \mathcal{G}_\alpha(\beta, Q)$*,* $0 \le \alpha \le 1$*, and* $n > 1$*. Let* $\widehat{\mathbf{p}}$ *be an estimator taking values in the network class* $\mathcal{F}_{\boldsymbol{\Phi}}(L, \mathbf{m}, s)$ *satisfying*

(i) $A(d,\beta)\log_2(n) \le L \lesssim n\phi_n$*,*

(ii) $\min_{i=1,\cdots,L} m_i \gtrsim n\phi_n$*,*

(iii) $s \asymp n\phi_n \log(n)$

for a suitable constant $A(d,\beta)$*. If* n *is sufficiently large, then, there exist constants* C', C'' *only depending on* α, C, β, d*, such that whenever* $\Delta_n(\widehat{\mathbf{p}}, \mathbf{p}_0) \le C'' B\phi_n L \log^2(n)$ *then*

$$R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) \le C' B\phi_n L \log^2(n).$$

An explicit expression for the constant $A(d,\beta)$ can be derived from the proof. The risk bound depends linearly on B. Choosing, for instance, $B = O(\log(n))$ leads only to an additional logarithmic factor in the convergence rate. The risk bound grows with $K^{\frac{(1+\alpha)\beta+(3+\alpha)d}{(1+\alpha)\beta+d}}$ in the number of classes. Thus for large β, we obtain a near linear dependence on K. The worst behavior occurs for $\alpha = 1$ and d large. Then the dependence on the number of classes is essentially of the order K^4.

When the estimator $\widehat{\mathbf{p}}$ is guaranteed to have output in $[e^{-B}, 1]^K$, the truncation parameter B in the risk has no effect. The proof of the approximation properties is done by the construction of a softmax-network $\widehat{\mathbf{g}}$ with the property that $\widehat{\mathbf{g}}(\mathbf{x}) \gtrsim K^{\frac{-(2+\alpha)\beta}{(1+\alpha)\beta+d}} n^{-\frac{\beta}{(1+\alpha)\beta+d}}$, for all $\mathbf{x} \in [0,1]^d$. This means that we can pick $B \asymp \log(n)$ such that $\widehat{\mathbf{g}}(\mathbf{x}) \ge e^{-B}$ and restrict the class $\mathcal{F}_{\boldsymbol{\Phi}}(L, \mathbf{m}, s)$ to networks that are guaranteed to have output in $[e^{-B}, 1]^K$. The proof of Theorem 2.3.3 can be extended for this setting and implies a risk bound for the Kullback-Leibler risk of the form

$$\mathbb{E}_{\mathcal{D}_n, \mathbf{X}}\left[\mathrm{KL}\left(\mathbf{p}_0(\mathbf{X}), \widehat{\mathbf{p}}(\mathbf{X})\right)\right] \le C''' \phi_n L \log^3(n),$$

for some constant C'''. Thus Theorem 2.3.3 provides us with rates for the Kullback-Leibler risk when the networks outputs are guaranteed to be sufficiently large, while still providing a bound for the truncated Kullback-Leibler risk when no such guarantee can be given.

When the input dimension d is large, the obtained convergence rates become slow. A possibility to circumvent this curse of dimensionality is to assume additional structure on $\mathbf{p}_0$. For nonparametric regression, [62, 72, 11, 127, 75] show that under a composition assumption on the regression function, neural networks can exploit this structure to obtain fast convergence rates that are unaffected by the curse of dimensionality. It is conceivable that for various classification problems, such an underlying composition structure is present. For instance to classify an email as spam, the hierarchical structure is important and decision trees that are adapted to such structures work well, see Section 9.2.5 in [56]. In image classification it is often assumed that an image can be constructed from compositions of simpler features; for example a square is built from lines and can itself be used as component of more complicated shapes.

It is possible to incorporate a composition assumption on the conditional class probabilities within the considered framework. As our approximation result already depends on Theorem 5 of [127] it is relatively straightforward to sketch how the additional composition assumption can help to deal with the curse of dimensionality. Consider the class of functions that satisfy the composition assumption in [127]

$$\begin{aligned}&\mathcal{G}_{\text{comp}}(r,\mathbf{d},\mathbf{t},\boldsymbol{\beta},Q):=\\&\qquad\Big\{f=\mathbf{g}_r\circ\cdots\circ\mathbf{g}_0:\mathbf{g}_i=(g_{ij})_j:[a_i,b_i]^{d_i}\to[a_{i+1},b_{i+1}]^{d_{i+1}},\\&\qquad\qquad g_{ij}\in\mathcal{C}^{\beta_i}([a_i,b_i]^{t_i},Q)\text{ for some }|a_i|,|b_i|\le Q\Big\}.\end{aligned}$$

Here t_i is the maximal number of variables on which each of the component functions g_{ij} may depend on. For specific structural assumptions, such as generalized additive models and sparse tensor decompositions, t_i can be much smaller than the input dimension d, [127].

In our setting the composition constraint can be incorporated by assuming that each of the conditional class probabilities $p_1,\ldots,p_K$ lies in the class $\mathcal{G}_{\text{comp}}(r,\mathbf{d},\mathbf{t},\boldsymbol{\beta},Q)$. Define the effective smoothness indices as $\beta_i^*:=\beta_i\prod_{\ell=i+1}^{r}(\beta_\ell\wedge 1)$. By approximating these composition functions by neural networks as in the proof of Theorem 1 of [127] in place of Theorem 5 of the same article, one can then obtain the rate

$$\phi_n=\max_{i=0,\cdots,r}K^{\frac{(1+\alpha)\beta_i^*+(3+\alpha)t_i}{(1+\alpha)\beta_i^*t_i}}n^{-\frac{(1+\alpha)\beta_i^*}{(1+\alpha)\beta_i^*+t_i}}.$$

Let us briefly summarize the related literature. Convergence rates for neural networks in (binary) classification have recently been studied in [67, 74, 76, 134, 111] in various settings. [67] derives convergence rates for the $0-1$ loss based on different surrogate losses and assumptions. For the hinge loss as surrogate loss, the margin condition in combination with smoothness conditions on the decision boundary as well as smoothness conditions on the conditional class probabilities are studied. Moreover, the logistic loss is analyzed under a condition that requires the conditional class probabilities to be near zero or one combined with smoothness conditions on the decision boundary. Convergence rates for the $0-1$ loss for convolutional neural networks are studied in [74, 76]. Both papers assume smoothness conditions on the conditional class probabilities and impose a max-pooling structure assumption for the conditional class probability that is related to the structure of convolutional networks. In [74] the least squares loss is used as a surrogate loss, while [76] uses the logistic loss as surrogate loss. More recently, [134] studied the convergence rates for convex Lipschitz losses of convolutional neural networks in binary classification under a submanifold condition. The framework includes least squares loss, hinge loss, truncated logistic loss and truncated exponential loss. In the truncated cases, the minimizers are also truncated. Furthermore, [111] studies convergence rates for the $0-1$ loss with the hinge loss as surrogate loss, in the case that the model is deterministic and that the decision boundary is Barron regular.

2.3.1 Relationship with Hellinger distance

The multiclass classification problem can be written as statistical model $(Q_{\mathbf{p}}, \mathbf{p} \in \mathcal{F})$, where $\mathcal{F}$ is the parameter space, $\mathbf{p}$ is the unknown vector of conditional class probabilities and $Q_{\mathbf{p}}$ denotes the data distribution if the data are generated from the conditional class probabilities $\mathbf{p}$. The squared Hellinger distance $H(P,Q)^2 = \frac{1}{2}\int(\sqrt{dP}-\sqrt{dQ})^2$, with P and Q probability measures on the same probability space, induces in a natural way a loss function on such a statistical model by associating to the two parameters $\mathbf{p}$ and $\mathbf{p}'$ the loss $H(Q_{\mathbf{p}}, Q_{\mathbf{p}'})$. The Hellinger loss function has been widely studied in the context of nonparametric variations of the maximum likelihood principle, mainly for the related nonparametric density estimation problem, [158, 147, 148]. The log-likelihood is closely related to the Kullback-Leibler divergence, which in turn is related to the Hellinger distance by the inequality $H(P,Q)^2 \leq \mathrm{KL}(P,Q)$, see for example [146]. The Kullback-Leibler divergence cannot be upper bounded by the squared Hellinger distance in general, although there exists conditions under which such a bound can be established, see for example Theorem 5 of [158] and Lemma 2.3.4 below.

In density estimation, the nonparametric MLE achieves in some regimes optimal rates with respect to the Hellinger distance for convex estimator classes or if the densities (or sieve estimators) are uniformly bounded away from zero, see [157, 158] and Chapters 7 and 10 in [147]. Neural network function classes are not convex and, as argued before, there are many applications in the deep learning literature, where the conditional class probabilities are very small or even zero. Thus, these general results are not applicable in our setting.

On the contrary, the convergence rates established above for the truncated Kullback-Leibler divergence imply convergence with respect to the Hellinger loss. This relationship is made precise in the next result.

Lemma 2.3.4. *Let P and Q be two probability measures defined on the same measurable space. For any $B \geq 2$,*

$$H^2(P,Q) \leq \frac{1}{2}\,\mathrm{KL}_2(P,Q) \leq \frac{1}{2}\,\mathrm{KL}_B(P,Q) \leq 2e^{B/2}H^2(P,Q).$$

For the proof see Appendix 2.C. The upper bound on the truncated Kullback-Leibler divergence is related to the inequalities that bound the Kullback-Leibler divergence by the squared Hellinger distance under the assumption of a bounded likelihood ratio, such as (7.6) in [20] or Lemma 4 in [57].

Combining the previous lemma and Theorem 2.3.3 with $B = 2$ gives

$$\mathbb{E}_{\mathcal{D}_n}\Big[\int_{[0,1]^d}\sum_{j=1}^{K}\Big(\sqrt{p_j^0(\mathbf{x})}-\sqrt{\widehat{p}_j(\mathbf{x})}\Big)^2\,d\mathbb{P}_{\mathbf{X}}(\mathbf{x})\Big]\le 2C'\phi_n L\log^2(n), \tag{2.3.1}$$

whenever $\Delta_n(\widehat{\mathbf{p}},\mathbf{p}_0)\le C''B\phi_n L\log^2(n)$.

We can also use the relation with the Hellinger distance to show that for $\alpha = 1$, we obtain a near minimax optimal convergence rate. Indeed $n^{-\frac{2\beta}{2\beta+d}}$ is the optimal rate for the squared Hellinger distance. For references see for instance Example 7.4.1 of [147] for univariate densities bounded away from zero; the entropy bounds in Theorem 2.7.1. together with Proposition 1 of [159] for densities bounded away from zero; or the entropy bounds in Theorem 2.7.1. and Equation (3.4.5) of [148] together with Chapter 2.3. of [159] for densities p for which $\int \frac{1}{p}$ is bounded. Since the squared Hellinger distance can be upper bounded by the Kullback-Leibler divergence, the rate $n^{-\frac{2\beta}{2\beta+d}}$ is also a lower bound for the Kullback-Leibler risk. Since this rate is achieved for $\alpha = 1$, it is clear that no further gain in the convergence rate can be expected for $\alpha > 1$. For $\alpha \geq 1$, the rate of convergence is up to $\log(n)$-factors the same as in Theorem 5 of [158] and also the conditions are comparable.

It is instructive to relate the global convergence rates to pointwise convergence. Recall that for real numbers a, b, we have $(\sqrt{a}-\sqrt{b})^2 = (a-b)^2/(\sqrt{a}+\sqrt{b})^2$. If $\mathbb{P}_{\mathbf{X}}$ has a Lebesgue density that is bounded on $[0,1]^d$ from below and above and if we choose L of the order $O(\log n)$, (2.3.1) indicates that on a large subset of $[0,1]^d$, we can expect a pointwise distance

$$\Big|p_j^0(\mathbf{x})-\widehat{p}_j(\mathbf{x})\Big|\lesssim\Big|\sqrt{p_j^0(\mathbf{x})}+\sqrt{\widehat{p}_j(\mathbf{x})}\Big|K^{\frac{(1+\alpha/2)d}{(1+\alpha)\beta+d}}n^{-\frac{(1+\alpha)\beta}{2(1+\alpha)\beta+2d}}\log^{3/2}(n).$$

The pointwise convergence rate gets therefore faster if the conditional class probabilities are small. In the most extreme case, $p_j^0(\mathbf{x}) = 0$, the previous bound becomes

$$\Big|p_j^0(\mathbf{x})-\widehat{p}_j(\mathbf{x})\Big|\lesssim K^{\frac{(2+\alpha)d}{(1+\alpha)\beta+d}}n^{-\frac{(1+\alpha)\beta}{(1+\alpha)\beta+d}}\log^{3}(n).$$

Since $n^{-(1+\alpha)\beta/((1+\alpha)\beta+d)} \ll n^{-\beta/(2\beta+d)}$, this rate can be much faster than the classical nonparametric rate for pointwise estimation $n^{-\beta/(2\beta+d)}$. The gain gets accentuated as the index α increases. A large index α in the SVB bound can be chosen if the conditional class probabilities are rarely small or zero. Hence there is a trade-off and the regions on which a faster rate can be obtained are thus smaller.

2.3.2 Oracle Inequality

The risk bound of Theorem 2.3.3 relies on an oracle-type inequality. Before we can state this inequality we first need some definitions. Given a function class of conditional class probabilities $\mathcal{F}$, we denote by $\log(\mathcal{F})$ the function class containing all functions that can be obtained by applying the logarithm coefficient-wise to functions from $\mathcal{F}$, that is,

$$\log(\mathcal{F}) = \big\{\mathbf{g} = \log(\mathbf{f}) : \mathbf{f} \in \mathcal{F}\big\}.$$

Next we define a family of pseudometrics. Recall that a pseudometric is a metric without the condition that $d(f, g) = 0$ implies $f = g$. For a real number τ and $\mathbf{f}, \mathbf{g} : \mathcal{D} \to \mathbb{R}^K$, set

$$d_\tau(\mathbf{f}, \mathbf{g}) := \sup_{\mathbf{x}\in\mathcal{D}} \max_{k=1,\cdots,K} |(\tau \vee f_k(\mathbf{x})) - (\tau \vee g_k(\mathbf{x}))|.$$

Lemma 2.C.3 in the appendix verifies that this indeed defines a pseudometric. For $\tau = -\infty$, $d_\tau(\mathbf{f}, \mathbf{g})$ coincides with the L^∞-norm as defined in the notation section.

Denote by $\mathcal{N}(\delta, \mathcal{F}, d(\cdot,\cdot))$ the δ interior covering number of a function class $\mathcal{F}$ with respect to a (pseudo)metric $d(.,.)$. For interior coverings, the centers of the balls of any cover are required to be inside the function class $\mathcal{F}$. Triangle inequality shows that any (exterior) δ-cover can be used to construct an interior cover with the same number of balls, but with radius 2δ instead of δ.

Theorem 2.3.5 (Oracle Inequality). *Let $\mathcal{F}$ be a class of conditional class probabilities and $\widehat{\mathbf{p}}$ be any estimator taking values in $\mathcal{F}$. If $B \geq 2$ and $\mathcal{N}_n = \mathcal{N}(\delta, \log(\mathcal{F}), d_\tau(\cdot,\cdot)) \geq 3$ for $\tau = \log(C_n e^{-B}/n)$, then*

$$R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) \leq (1+\epsilon)\left(\inf_{\mathbf{p}\in\mathcal{F}} R(\mathbf{p}_0, \mathbf{p}) + \Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}}) + 3\delta\right) + \frac{(1+\epsilon)^2}{\epsilon} \cdot \frac{68B\log(\mathcal{N}_n) + 272B + (3/2)C_nK\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n},$$

for all $\delta, \epsilon \in (0, 1]$, $0 < C_n \leq ne^{-1}$ and $\Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}})$ as defined in (2.2.2).

The proof of this oracle inequality is a non-trivial variation of the proof for the oracle inequality in the regression model [127]. The statement seems to suggest to pick a small C_n. Then, however, also τ will be small, and d_τ becomes a stronger metric possibly leading to an increase of the covering number $\mathcal{N}_n$.

We can also replace the covering number of $\log(\mathcal{F})$ by the covering number of $\mathcal{F}$ in the oracle inequality:

Corollary 2.3.6. *Denote $\widetilde{\mathcal{N}}_n := \mathcal{N}(\delta C_n e^{-B}/n, \mathcal{F}, d_\tau(\cdot,\cdot))$, with $\tau = C_n e^{-B}/n$. Under the conditions of Theorem 2.3.5, it holds that*

$$R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) \le (1+\epsilon)\left(\inf_{\mathbf{p}\in\mathcal{F}} R(\mathbf{p}_0,\mathbf{p}) + \Delta_n(\mathbf{p}_0,\widehat{\mathbf{p}}) + 3\delta\right) + \frac{(1+\epsilon)^2}{\epsilon}\cdot\frac{68B\log(\widetilde{\mathcal{N}}_n) + 272B + (3/2)C_nK(\log(n/C_n)+B)}{n},$$

for all $\delta, \epsilon \in (0,1]$, $0 < C_n \le ne^{-1}$ and $\Delta_n(\mathbf{p}_0,\widehat{\mathbf{p}})$ as defined in (2.2.2).

Let us briefly discuss some ideas underlying the proof of the oracle inequality. For simplicity, assume that $\widehat{\mathbf{p}}$ is the MLE over a class $\mathcal{F}$ and that $\mathbf{p}_0 \in \mathcal{F}$. By the definition of the MLE $\widehat{\mathbf{p}}$, we have that $-\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i^\top \log(\widehat{\mathbf{p}}(\mathbf{X}_i)) \le -\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i^\top \log(\mathbf{p}_0(\mathbf{X}_i))$. Taking expectation on both sides, one can then show that for any $B \ge 0$,

$$\mathbb{E}_{\mathcal{D}_n}\Big[\frac{1}{n}\sum_{i=1}^n \mathbf{p}_0(\mathbf{X}_i)^\top\Big(B\wedge\log\Big(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\widehat{\mathbf{p}}(\mathbf{X}_i)}\Big)\Big)\Big] \le \mathbb{E}_{\mathcal{D}_n}\Big[\frac{1}{n}\sum_{i=1}^n \big(\mathbf{p}_0(\mathbf{X}_i)-\mathbf{Y}_i\big)^\top\Big(B\wedge\log\Big(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\widehat{\mathbf{p}}(\mathbf{X}_i)}\Big)\Big)\Big].$$

Using standard empirical process arguments, the right hand side can be roughly upper bounded by $\mathbb{E}_{\mathcal{D}_n}[\max_j \frac{1}{n}\sum_{i=1}^n(\mathbf{p}_0(\mathbf{X}_i)-\mathbf{Y}_i)^\top(B\wedge\log(\mathbf{p}_0(\mathbf{X}_i)/\mathbf{p}_j(\mathbf{X}_i)))]$, where the maximum is over all centers of an ε-covering of $\mathcal{F}$ for a sufficiently small ε. Since $\mathbb{E}_{\mathcal{D}_n}[\mathbf{Y}_i|\mathbf{X}_i] = \mathbf{p}_0(\mathbf{X}_i)$, this is the maximum over a centered process. Using empirical process theory a second time, the left hand side of the previous display can be shown to converge to the statistical risk $R_B(\mathbf{p}_0,\widehat{\mathbf{p}}) = \mathbb{E}_{\mathcal{D}_n,\mathbf{X}}[\mathrm{KL}_B(\mathbf{p}_0(\mathbf{X}),\widehat{\mathbf{p}}(\mathbf{X}))]$.

To apply Bernstein's inequality we need to bound the moments of the random variables in the empirical process. For that we have derived the following inequality that relates the m-th moment to the truncated Kullback-Leibler divergence and also shows the effect of the truncation level B.

Lemma 2.3.7. *If $B > 1$ and $m = 2, 3, \ldots$, then, for any two probability vectors $(p_1, \ldots, p_K)$ and $(q_1, \ldots, q_K)$, we have*

$$\sum_{k=1}^K p_k\left|B\wedge\log\left(\frac{p_k}{q_k}\right)\right|^m \le \max\left\{m!, \frac{B^m}{B-1}\right\}\sum_{k=1}^K p_k\left(B\wedge\log\left(\frac{p_k}{q_k}\right)\right).$$

In order to use the oracle inequality for deep ReLU networks with softmax activation in the output layer, we now state a bound on the covering number of these classes. The bound and its proof are a slight modification of Lemma 5 in [127].

Lemma 2.3.8. *If* $V := \prod_{\ell=0}^{L+1}(m_\ell + 1)$*, then for every* $\delta > 0$*,*

$$\mathcal{N}\left(\delta, \log(\mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s)), \|\cdot\|_\infty\right) \le \left(4\delta^{-1}K(L+1)V^2\right)^{s+1},$$

and

$$\log \mathcal{N}\left(\delta, \log(\mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s)), \|\cdot\|_\infty\right) \le (s+1)\log(2^{2L+6}\delta^{-1}(L+1)K^3d^2s^L).$$

The second bound follows from the first by removing inactive nodes, Proposition 2.A.1, and taking the logarithm. The full proof can be found in Appendix 2.C.

The proof of the main risk bound in Theorem 2.3.3 is based on the oracle inequality derived above. To bound the individual error terms, we apply the approximation theory developed in Theorem 2.3.2 and Lemma 2.4.3 as well as the previous bound on the metric entropy. This shows that for any $M > 1$, the truncated Kullback-Leibler risk for a network class with depth L, width $\gtrsim KM^{d/\beta}$ and sparsity $s \lesssim KM^{d/\beta}$ can be bounded by

$$R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) \lesssim K^{3+\alpha}\frac{\log(M)}{M^{1+\alpha}} + KM^{d/\beta}L\frac{\log^2(n)}{n} + \Delta_n(\widehat{\mathbf{p}}, \mathbf{p}_0).$$

Balancing the terms $K^{3+\alpha}/M^{1+\alpha}$ and $KM^{d/\beta}$ leads to $M \asymp K^{\frac{(2+\alpha)\beta}{(1+\alpha)\beta+d}}n^{\frac{\beta}{(1+\alpha)\beta+d}}$ and for small $\Delta_n(\widehat{\mathbf{p}}, \mathbf{p}_0)$, we get the rate

$$R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) \lesssim K^{\frac{(1+\alpha)\beta+(3+\alpha)d}{(1+\alpha)\beta+d}}n^{-\frac{(1+\alpha)\beta}{(1+\alpha)\beta+d}}L\log^2(n)$$

in Theorem 2.3.3.

2.4 Proofs

Proof of Lemma 2.2.1. Consider the event $\mathcal{A}_n := \{(\mathbf{X}_i, \mathbf{Y}_i) \in ([0,1/3]^d \times (1,0)^\top) \cup ([2/3,1]^d \times (0,1)^\top),\ \text{for all } i = 1, \dots, n\}$. Recall that $0\log(0) = 0$. On the event $\mathcal{A}_n$, for any $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}))^\top$ such that $p_1(\mathbf{x}) = 0$ for all $\mathbf{x} \in [0,1/3]^d$ and $p_1(\mathbf{x}) = 1$ for all $\mathbf{x} \in [2/3,1]^d$, we have that $\ell(\mathbf{p}, \mathcal{D}_n) = 0$, where $\ell(\mathbf{p}, \mathcal{D}_n)$ is the negative log-likelihood as defined in (2.2.1). Since the CE loss is nonnegative, any such $\mathbf{p}$ in the class $\mathcal{F}$ is a MLE on this event. Since $\mathbb{P}(\mathcal{A}_n) > 0$, it follows that

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n,\mathbf{X}}\left[\mathbf{p}_0(\mathbf{X})^\top \log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\widehat{\mathbf{p}}(\mathbf{X})}\right)\right] &\ge \mathbb{E}_{\mathcal{D}_n}\left[\mathbf{1}(\mathcal{A}_n)\int_{[0,1]^d}\mathbf{p}_0(\mathbf{u})^\top \log\left(\frac{\mathbf{p}_0(\mathbf{u})}{\widehat{\mathbf{p}}(\mathbf{u})}\right)d\mathbf{u}\right] \\ &= \infty \cdot \mathbb{P}(\mathcal{A}_n) \\ &= \infty.\end{aligned}$$

□

2.4.1 Approximation related results

This section is devoted to the proof of Theorem 2.3.3. First we construct a neural network that approximates $\mathbf{p}_0$ in terms of the L_∞-norm and is bounded away from zero. Afterwards we prove Theorem 2.3.2 relating the previously derived approximation theory to a bound on the approximation error in terms of the expected Kullback-Leibler divergence. We finish the proof combining this network with the new oracle inequality (Theorem 2.3.5) and an entropy bound for classes of neural networks with a softmax function in the output layer, Lemma 2.3.8. Recall that $\mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$ denotes the neural network class with L hidden layers, width vector $\mathbf{m}$, network sparsity s and identity activation function in the output layer.

Theorem 2.4.1. *For all $M \geq 2$ and $\beta > 0$ there exists a neural network $G \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$, with*
(i) $L = \lfloor 40(\beta+2)^2 \log_2(M) \rfloor$,
(ii) $\mathbf{m} = (1, \lfloor 48\lceil\beta\rceil^3 2^\beta M^{1/\beta}\rfloor, \cdots, \lfloor 48\lceil\beta\rceil^3 2^\beta M^{1/\beta}\rfloor, 1)$,
(iii) $s \leq 4284(\beta+2)^5 2^\beta M^{1/\beta} \log_2(M)$,
such that for any $x \in [0,1]$,

$$\left|e^{G(x)} - x\right| \leq \frac{4}{M} \quad \textit{and} \quad G(x) \geq \log\left(\frac{4}{M}\right).$$

The proof of this theorem can be found in Appendix 2.B. To approximate Hölder functions we use Theorem 5 from [127] with m equal to $\lceil \log_2(M))(d/\beta+1)\rceil$. We state here a variation of that theorem in our notation using weaker upper bounds to simplify the expressions for the network size. These upper bounds can be deduced directly from the depth-synchronization and network enlarging properties of neural networks stated in Section 2.A.1. Set

$$C_{Q,\beta,d} := (2Q+1)(1+d^2+\beta^2)6^d + Q3^\beta.$$

Theorem 2.4.2. *For every function $\mathbf{f} \in \mathcal{G}(\beta, Q)$ and every $M > (\beta+1)^\beta \vee (Q+1)^{\beta/d} e^\beta$, there exist neural networks $H_k \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$ with*
(i) $L = 3\lceil \log_2(M)(d/\beta+1)\rceil(1+\lceil\log_2(d \vee \beta)\rceil)$,
(ii) $\mathbf{m} = (d, 6(d+\lceil\beta\rceil)\lfloor M^{d/\beta}\rfloor, \cdots, 6(d+\lceil\beta\rceil)\lfloor M^{d/\beta}\rfloor, 1)$,
(iii) $s \leq 423(d+\beta+1)^{3+d} M^{d/\beta} \log_2(M))(d/\beta+1)$,
such that

$$\left\|H_k - f_k^0\right\|_\infty \leq \frac{C_{Q,\beta,d}}{M}, \quad \forall k \in \{1, \cdots, K\}.$$

Here the M is chosen such that $M^{d/\beta} \asymp N$, where N is as defined in Theorem 5 of [127].

Without loss of generality we can assume that the output of the H_k networks lies in $[0,1]$. Indeed if this would not be the case, then the projection-layer that we use later on in our proof will guarantee that it is in this interval. This will not increase the error since the functions f_k^0 only take values in $[0,1]$.

To obtain a neural network with softmax output, the next lemma combines the neural network constructions from the previous two theorems and replaces the output with a softmax function.

Lemma 2.4.3. *For every function* $\mathbf{f} \in \mathcal{G}(\beta, Q)$ *and every* $M > K(4 + C_{Q,\beta,d}) \vee (\beta + 1)^\beta \vee (Q+1)^{\beta/d} e^\beta$, *there exists a neural network* $\widetilde{\mathbf{q}} \in \mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s)$, *with*

(i) $L = 3\lceil \log_2(M)(d/\beta + 1)\rceil (1 + \lceil \log_2(d+\beta)\rceil) + \lfloor 40(\beta+2)^2 \log_2(M)\rfloor + 2$,

(ii) $\mathbf{m} = \Big(d, \lfloor 48K(d + \lceil\beta\rceil^3)2^\beta M^{d/\beta}\rfloor, \cdots, \lfloor 48K(d + \lceil\beta\rceil^3)2^\beta M^{d/\beta}\rfloor, K\Big)$,

(iii) $s \leq 4707K(d+\beta+1)^{4+d} 2^\beta M^{d/\beta} \log_2(M))(d/\beta + 1)$,

such that,

$$\|\widetilde{\mathbf{q}}_k - \mathbf{p}_0\|_\infty \leq \frac{2K(4 + C_{Q,\beta,d})}{M},$$

and

$$\widetilde{q}_k(\mathbf{x}) \geq \frac{1}{M}, \quad \forall\, k \in \{1, \cdots, K\}, \, \forall\, \mathbf{x} \in [0,1]^d.$$

Proof. Composing the neural networks in Theorem 2.4.1 and Theorem 2.4.2 results in a neural network $\mathbf{G} = (G(H_1), \cdots, G(H_K))$ such that for any $k = 1, \cdots, K$,

$$\left\|e^{G(H_k)} - p_k^0\right\|_\infty \leq \left\|e^{G(H_k)} - H_k\right\|_\infty + \left\|H_k - p_k^0\right\|_\infty \leq \frac{4 + C_{Q,\beta,d}}{M}.$$

Define now the vector valued function $\widetilde{\mathbf{q}}$ component-wise by

$$\widetilde{q}_k(\mathbf{x}) = \frac{e^{G(H_k(\mathbf{x}))}}{\sum_{j=1}^K e^{G(H_j(\mathbf{x}))}}, \quad k = 1, \cdots, K.$$

Applying the composition (2.A.2), depth synchronization (2.A.3) and parallelization rules (2.A.4) it follows that $\widetilde{\mathbf{q}} \in \mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s)$. To bound $\|\widetilde{q}_k - p_k^0\|_\infty$, we use that $\mathbf{p}_0 = (p_1^0, \cdots, p_K^0)$ is a probability vector, $e^{G(H_j)} \geq 0$ for $j = 1, \cdots, k$ and triangle inequality, to obtain

$$\begin{aligned}\left\|\widetilde{q}_k - p_k^0\right\|_\infty \leq & \left\| e^{G(H_k)} \left(\frac{1}{\sum_{j=1}^K e^{G(H_j)}} - 1 \right) \right\|_\infty + \left\| e^{G(H_k)} - p_k^0 \right\|_\infty \\ = & \left\| e^{G(H_k)} \left(\frac{\sum_{\ell=1}^K p_\ell^0}{\sum_{j=1}^K e^{G(H_j)}} - \frac{\sum_{\ell=1}^K e^{G(H_\ell)}}{\sum_{j=1}^K e^{G(H_j)}} \right) \right\|_\infty + \left\| e^{G(H_k)} - p_k^0 \right\|_\infty \\ \leq & \Big(\sum_{\ell=1}^K \left\| p_\ell^0 - e^{G(H_\ell)} \right\|_\infty \Big) \left\| \frac{e^{G(H_k(\cdot))}}{\sum_{j=1}^K e^{G(H_j)}} \right\|_\infty + \left\| e^{G(H_k)} - p_k^0 \right\|_\infty \\ \leq & \frac{(K+1)(4 + C_{Q,\beta,d})}{M} \leq \frac{2K(4 + C_{Q,\beta,d})}{M}.\end{aligned}$$

For the second bound of the lemma, notice that from the first bound of the lemma and the second bound of Theorem 2.4.1 it follows that

$$\widetilde{q}_k(\mathbf{x}) \geq \frac{\frac{4}{M}}{\sum_{j=1}^K e^{G(H_j(\mathbf{x}))}} \geq \frac{\frac{4}{M}}{1 + K\frac{(4+C_{Q,\beta,d})}{M}} = \frac{4}{M + K(4 + C_{Q,\beta,d})} \geq \frac{1}{M},$$

where for the second inequality we used that $p_j(\mathbf{x}) \leq 1$, so $e^{G(H_j(\mathbf{x}))} \leq p_j^0(\mathbf{x}) + (4 + C_{Q,\beta,d})/M$ and for the last inequality we used that $M \geq K(4 + C_{Q,\beta,d})$. □

The Kullback-Leibler divergence can be upper bounded by the χ^2-divergence, see for instance Lemma 2.7 in [146]. Thus,

$$\mathbb{E}_{\mathbf{X}} \left[(\mathbf{p}_0(\mathbf{X}))^\top \log \left(\frac{\mathbf{p}_0(\mathbf{X})}{\widetilde{\mathbf{q}}(\mathbf{X})} \right) \right] \leq \mathbb{E}_{\mathbf{X}} \left[\sum_{k=1}^K \frac{(p_k^0(\mathbf{X}) - \widetilde{q}_k(\mathbf{X}))^2}{\widetilde{q}_k(\mathbf{X})} \right].$$

To control the approximation error, we can combine this bound with the first bound of Lemma 2.4.3 to conclude that if $p_k^0(\mathbf{X}) > 2K(4 + C_{Q,\beta,d})/M$, then

$$\frac{(p_k^0(\mathbf{X}) - \widetilde{q}_k(\mathbf{X}))^2}{\widetilde{q}_k(\mathbf{X})} \leq \frac{4K^2(4 + C_{Q,\beta,d})^2}{M^2} \left(p_k^0(\mathbf{X}) - \frac{2K(4 + C_{Q,\beta,d})}{M} \right)^{-1}.$$

On the other hand, combining the bound with the second inequality from the same lemma yields

$$\begin{aligned}&\frac{(p_k^0(\mathbf{X}) - \widetilde{q}_k(\mathbf{X}))^2}{\widetilde{q}_k(\mathbf{X})} \\ &\qquad \leq \sum_{k=1}^K \frac{4K^2(4 + C_{Q,\beta,d})^2}{M^2} \left(\max\left\{ p_k^0(\mathbf{X}) - \frac{2K(4 + C_{Q,\beta,d})}{M}, \frac{1}{M} \right\} \right)^{-1},\end{aligned}$$

which is valid for all possible values of $\mathbf{p}_0(\mathbf{x}) \in [0,1]^k$. As M tends to infinity, $p_k^0(\mathbf{x}) - 2K(4 + C_{Q,\beta,d})/M$ tends to $p_k^0(\mathbf{x})$, while $1/M$ tends to zero. Without any further conditions on $p_k^0(\mathbf{X})$ this bound is thus of order M^{-1}. The small value bound, however, allows us to obtain an upper bound with better behaviour in M. The following proposition employs the small value bound to control the expectation of $(p_k^0(\mathbf{x}))^{-1}$ on the set that $p_k^0(\mathbf{x})$ exceeds some threshold value H.

Proposition 2.4.4. *Assume there exists an $\alpha \geq 0$ and a finite constant $C < \infty$, such that for $\mathbf{p} = (p_1, \dots, p_K) : \mathcal{D} \to \mathcal{S}^K$ we have $\mathbb{P}_{\mathbf{X}}(p_k(\mathbf{X}) \leq t) \leq Ct^\alpha$ for all $t \geq 0$ and $k \in \{1, \dots, K\}$. Let $H \in [0,1]$. Then it holds that*

$$\int_{\{p_k(\mathbf{x}) \geq H\}} \frac{1}{p_k(\mathbf{x})}\, d\mathbb{P}_{\mathbf{X}}(\mathbf{x}) \leq \begin{cases} C\frac{H^{\alpha-1}}{1-\alpha}, & \textit{if } \alpha \in [0,1), \\ C(1 - \log(H)), & \textit{if } \alpha \geq 1. \end{cases}$$

Proof. Observe that $p_k(\mathbf{X})$ is a probability. Therefore, $p_k(\mathbf{X}) \leq 1$ and consequently $C \geq 1$. For any nonnegative function h and random variable $Z \sim \mathbb{P}_Z$, we have $\int h(Z)\, d\mathbb{P}_Z = \mathbb{E}[h(Z)] = \int_0^\infty \mathbb{P}_Z(h(Z) \geq u)\, du$. Hence

$$\begin{aligned} \int_{\{p_k(\mathbf{x}) \geq H\}} \frac{1}{p_k(\mathbf{x})}\, d\mathbb{P}_{\mathbf{X}}(\mathbf{x}) &= \int_0^\infty \mathbb{P}_{\mathbf{X}}\left(\frac{1}{p_k(\mathbf{X})} \mathbb{1}_{\{p_k(\mathbf{X}) \geq H\}} \geq u\right) du \\ &\leq \int_0^{\frac{1}{H}} \mathbb{P}_{\mathbf{X}}\Big(p_k(\mathbf{X}) \leq \frac{1}{u}\Big)\, du, \end{aligned}$$

where the inequality follows from observing that $\frac{1}{p_k(\mathbf{X})} \mathbb{1}_{\{p_k(\mathbf{X}) \geq H\}} \geq u$ implies $H \leq p_k(\mathbf{X}) \leq \frac{1}{u}$ and $u \leq 1/H$.

If $\alpha = 0$, we use the trivial bound $\mathbb{P}_{\mathbf{X}}(p_k(\mathbf{x}) \leq t) \leq 1$, for all $t \in [0,1]$, and obtain

$$\int_0^{\frac{1}{H}} \mathbb{P}_{\mathbf{X}}\Big(p_k(\mathbf{X}) \leq \frac{1}{u}\Big)\, du \leq \int_0^{\frac{1}{H}} 1\, du = \frac{1}{H}.$$

If $0 < \alpha < 1$, we can invoke the assumption of this proposition to obtain

$$\int_0^{\frac{1}{H}} \mathbb{P}_{\mathbf{X}}\Big(p_k(\mathbf{X}) \leq \frac{1}{u}\Big)\, du \leq C \int_0^{\frac{1}{H}} u^{-\alpha}\, du = \frac{CH^{\alpha-1}}{1-\alpha}.$$

For $\alpha \geq 1$, we have $\mathbb{P}_{\mathbf{X}}(p_k(\mathbf{X}) \leq t) \leq Ct$ for all $0 \leq t \leq 1$. If moreover $C \leq H^{-1}$, the inequality $\mathbb{P}_{\mathbf{X}}(p_k(\mathbf{X}) \leq t) \leq \min\{1, Ct\}$ leads to

$$\begin{aligned} \int_0^{\frac{1}{H}} \mathbb{P}_{\mathbf{X}}\Big(p_k(\mathbf{X}) \leq \frac{1}{u}\Big)\, du &\leq \int_0^C 1\, du + C \int_C^{\frac{1}{H}} \frac{1}{u}\, du \\ &= C + C(-\log(H) - \log(C)). \end{aligned}$$

If $\alpha \geq 1$ and $C \geq H^{-1}$, we can upper bound the integral by $\int_0^C 1\,du = C$. The result of the proposition now follows from simplifying the expressions using that $C \geq 1$. □

We can now state and prove the main approximation bound.

Proof of Theorem 2.3.2. The condition $\|\mathbf{p} - \mathbf{p}_0\|_\infty \leq C_1/M$ implies that $p_k(\mathbf{x}) \geq p_k^0(\mathbf{x}) - C_1/M$. Combined with $p_k(\mathbf{x}) \geq 1/M$, this gives

$$p_k(\mathbf{x}) \geq \left(p_k^0(\mathbf{x}) - \frac{C_1}{M}\right) \vee \frac{1}{M} \geq \frac{p_k^0(\mathbf{x})}{C_1+1} \vee \frac{1}{M},$$

where we used that $p_k^0(\mathbf{x}) \geq (C_1+1)/M = ((C_1+1)/C_1) \cdot (C_1/M)$ implies

$$p_k^0(\mathbf{x}) - \frac{C_1}{M} \geq p_k^0(\mathbf{x}) \left(1 - \frac{C_1}{C_1+1}\right) = \frac{p_k^0(\mathbf{x})}{C_1+1}.$$

This gives rise to the upper bound

$$\frac{(p_k^0(\mathbf{X}) - p_k(\mathbf{X}))^2}{p_k(\mathbf{X})} \leq \frac{C_1^2}{M} \mathbb{1}_{\{p_k^0(\mathbf{x}) \leq \frac{C_1+1}{M}\}} + \frac{C_1^2}{M^2} \cdot \frac{C_1+1}{p_k^0(\mathbf{x})} \mathbb{1}_{\{p_k^0(\mathbf{x}) \geq \frac{C_1+1}{M}\}}.$$

Taking the expectation over the right hand side yields

$$\frac{C_1^2}{M} \mathbb{P}_{\mathbf{X}} \left(p_k^0(\mathbf{x}) \leq \frac{C_1+1}{M}\right) + \frac{C_1^2(C_1+1)}{M^2} \int_{\{p_k^0(\mathbf{x}) \geq \frac{C_1+1}{M}\}} \frac{1}{p_k^0(\mathbf{x})}\, d\mathbb{P}_{\mathbf{X}}(\mathbf{x})$$

By the α-SVB condition the first term is upper bounded by

$$\frac{C_1^2}{M} \mathbb{P}_{\mathbf{X}} \left(p_k^0(\mathbf{x}) \leq \frac{C_1+1}{M}\right) \leq \frac{C_1^2 C}{M} \left(\frac{C_1+1}{M}\right)^{\alpha \wedge 1} \leq C \frac{(C_1+1)^{2+(\alpha\wedge 1)}}{M^{1+(\alpha \wedge 1)}}.$$

Applying Proposition 2.4.4 with $H = (C_1+1)/M$ to the second term yields the result. □

Now we have all the ingredients to complete the proof of the main theorem.

Proof of Theorem 2.3.3. Take $\delta = n^{-1}$ and $\epsilon = C_n = 1$ in Theorem 2.3.5. Using that d_τ is upper bounded by the sup-norm distance together with Lemma 2.3.8 gives

$$\begin{aligned} R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) &\leq 2 \left(\inf_{\mathbf{p} \in \mathcal{F}} R(\mathbf{p}_0, \mathbf{p}) + \Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}}) + \frac{3}{n}\right) \\ &+ 4 \cdot \frac{68B(s+1)\log(2^{2L+6} n(L+1)K^3 d^2 s^L) + 272B + (3/2)K(\log(n) + B)}{n}. \end{aligned} \tag{2.4.1}$$

Recall that $0 \leq \alpha \leq 1$ is the index from the SVB condition. We now choose $M = \lfloor cK^{\frac{(2+\alpha)\beta}{(1+\alpha)\beta+d}} n^{\frac{\beta}{(1+\alpha)\beta+d}} \rfloor$ for a small constant c chosen below. To apply Lemma 2.4.3, we need to show that $M \gg K$. To see this, observe that $R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) \leq B$ and therefore the convergence rate becomes trivial if $\phi_n \geq 1$. Using that $\phi_n = K^{\frac{(1+\alpha)\beta+(3+\alpha)d}{(1+\alpha)\beta+d}} n^{-\frac{(1+\alpha)\beta}{(1+\alpha)\beta+d}}$, this implies $K \leq n^{\frac{(1+\alpha)\beta}{(1+\alpha)\beta+(3+\alpha)d}} \leq n^{\frac{\beta}{\beta+2d}} \leq n^{\frac{\beta}{2d}}$. Hence, $K^{d-\beta} \ll n^{\beta}$ and thus also $M \gg K$.

For this choice of M, the network $\widetilde{\mathbf{q}}$ from Lemma 2.4.3 is in the network class $\mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s)$, where $L = 3\lceil \log_2(M)(d/\beta+1)\rceil(1+\lceil\log_2(d+\beta)\rceil)+\lfloor 40(\beta+2)^2\log_2(M)\rfloor + 2$, the maximum width of the hidden layers is bounded by $\lesssim Kc^{d/\beta}M^{d/\beta} = c^{d/\beta}n\phi_n$ and similarly $s \lesssim Kc^{d/\beta}M^{d/\beta}\log_2(M) = c^{d/\beta}n\phi_n\log_2(M)$. In particular, by taking c sufficiently small and using the depth synchronization property (2.A.3), $\widetilde{\mathbf{q}} \in \mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s)$, whenever $A(d,\beta)\log_2(n) \leq L \lesssim n\phi_n$, for a suitable constant $A(d,\beta)$, the maximum width is $\gtrsim n\phi_n$ and $s \asymp n\phi_n\log(n)$. We now apply Theorem 2.3.2 with $C_1 = 2K(4 + C_{Q,\beta,d})$. Using that $C_1 + 1 = 2K(4+C_{Q,\beta,d}) + 1 \leq 2K(5+C_{Q,\beta,d})$, we find

$$\inf_{\mathbf{p}\in\mathcal{F}} R(\mathbf{p}_0, \mathbf{p}) \leq 8CK^{3+\alpha}\frac{(5+C_{Q,\beta,d})^3}{M^{1+\alpha}}\Big(1 + \frac{\mathbb{1}_{\{\alpha<1\}}}{1-\alpha} + \log(M)\Big) \lesssim \phi_n\log(n).$$

Together with (2.4.1) and $s \asymp n\phi_n\log(n)$, the statement of Theorem 2.3.3 follows. □

2.4.2 Oracle inequality related results

In this section we prove Theorem 2.3.5. For $B > 0$, consider

$$R_{B,n}(\mathbf{p}_0, \widehat{\mathbf{p}}) := \mathbb{E}_{\mathcal{D}_n}\Big[\frac{1}{n}\sum_{i=1}^{n}\mathbf{Y}_i^\top\Big(B \wedge \log\Big(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\widehat{\mathbf{p}}(\mathbf{X}_i)}\Big)\Big)\Big].$$

The next proposition shows how this risk is related to the approximation error and the quantity $\Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}})$ defined in (2.2.2) that measures the empirical distance between an arbitrary estimator and an empirical risk minimizer.

Proposition 2.4.5. *For any estimator* $\widehat{\mathbf{p}} \in \mathcal{F}$,

$$R_{B,n}(\mathbf{p}_0, \widehat{\mathbf{p}}) \leq R_{\infty,n}(\mathbf{p}_0, \widehat{\mathbf{p}}) \leq \inf_{\mathbf{p}\in\mathcal{F}} R(\mathbf{p}_0, \mathbf{p}) + \Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}}).$$

Proof. The first inequality follows from $a \geq \min(a, b)$, for all $a, b \in \mathbb{R}$. To prove the second inequality, fix a $\mathbf{p}^* \in \mathcal{F}$. Using that $\Delta_n(\mathbf{p}_0, \mathbf{p}^*) \geq 0$ and

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n}\big[\mathbf{Y}_i^\top \log(\mathbf{p}^*(\mathbf{X}_i))\big] &= \mathbb{E}_{\mathcal{D}_n}\big[\mathbb{E}_{\mathcal{D}_n}[\mathbf{Y}_i^\top|\mathbf{X}_i]\log(\mathbf{p}^*(\mathbf{X}_i))\big] \\ &= \mathbb{E}_{\mathcal{D}_n}\big[\mathbf{p}_0(\mathbf{X}_i)^\top \log(\mathbf{p}^*(\mathbf{X}_i))\big],\end{aligned}$$

we get

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n}\Big[-\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i^\top \log(\widehat{\mathbf{p}}(\mathbf{X}_i))\Big] &\leq \mathbb{E}_{\mathcal{D}_n}\Big[-\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i^\top \log(\widehat{\mathbf{p}}(\mathbf{X}_i))\Big] + \Delta_n(\mathbf{p}_0, \mathbf{p}^*)\\ &= \mathbb{E}_{\mathcal{D}_n}\Big[-\frac{1}{n}\sum_{i=1}^n \mathbf{Y}_i^\top \log(\mathbf{p}^*(\mathbf{X}_i))\Big] + \Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}})\\ &= \mathbb{E}_{\mathbf{X}}\Big[-\mathbf{p}_0^\top(\mathbf{X})\log(\mathbf{p}^*(\mathbf{X}))\Big] + \Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}}).\end{aligned}$$

As this holds for all $\mathbf{p}^* \in \mathcal{F}$, we can take on the right hand side also the infimum over all $\mathbf{p}^* \in \mathcal{F}$. To complete the proof for the second inequality, we add to both sides $\mathbb{E}_{\mathcal{D}_n}[\mathbf{Y}_i^\top \log(\mathbf{p}_0(\mathbf{X}_i))] = \mathbb{E}_{\mathcal{D}_n}[\mathbf{p}_0(\mathbf{X}_i)^\top \log(\mathbf{p}_0(\mathbf{X}_i))]$. □

The truncation level B allows us to split the statistical risk into multiple parts that can be controlled separately. The following lemma provides a bound on the event that $p_k^0(\mathbf{X})$ is small.

Lemma 2.4.6. *Let $\mathcal{F}$ be a class of conditional class probabilities, $\widehat{\mathbf{p}}$ be any estimator taking values in $\mathcal{F}$, $(\overline{\mathbf{X}}, \overline{\mathbf{Y}})$ be a random pair with the same distribution as $(\mathbf{X}_1, \mathbf{Y}_1)$ and $C_n \in (0, n/e]$. Then, for any $i \in \{1, \cdots, n\}$, and any $k \in \{1, \cdots, K\}$, we have*

$$\left|\mathbb{E}_{\mathcal{D}_n,(\overline{\mathbf{X}},\overline{\mathbf{Y}})}\left[\overline{Y}_k \mathbb{1}_{\{p_k^0(\overline{\mathbf{X}}) \leq \frac{C_n}{n}\}}\Big(B \wedge \log\Big(\frac{p_k^0(\overline{\mathbf{X}})}{\widehat{p}_k(\overline{\mathbf{X}})}\Big)\Big)\right]\right| \leq \frac{C_n\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n}.$$

Proof. Since $\mathbf{p}_0, \widehat{\mathbf{p}} \in [0,1]^K$, we have

$$\log(p_k^0(\overline{\mathbf{X}})) \leq B \wedge \log\Big(\frac{p_k^0(\overline{\mathbf{X}})}{\widehat{p}_k(\overline{\mathbf{X}})}\Big) \leq B. \tag{2.4.2}$$

Using that $a \leq x \leq b$ implies $|x| \leq \max\{|a|, |b|\} \leq |a| + |b|$ and $Y_k \geq 0$, we can get an upper bound that does not depend on $\widehat{\mathbf{p}}$

$$\begin{aligned}&\left|\mathbb{E}_{\mathcal{D}_n,(\overline{\mathbf{X}},\overline{\mathbf{Y}})}\left[\overline{Y}_k \mathbb{1}_{\{p_k^0(\overline{\mathbf{X}}) \leq \frac{C_n}{n}\}}\Big(B \wedge \log\Big(\frac{p_k^0(\overline{\mathbf{X}})}{\widehat{p}_k(\overline{\mathbf{X}})}\Big)\Big)\right]\right|\\ &\leq \mathbb{E}_{(\overline{\mathbf{X}},\overline{\mathbf{Y}})}\left[\overline{Y}_k \mathbb{1}_{\{p_k^0(\overline{\mathbf{X}}) \leq \frac{C_n}{n}\}}\left|\log(p_k^0(\overline{\mathbf{X}}))\right|\right] + \mathbb{E}_{(\overline{\mathbf{X}},\overline{\mathbf{Y}})}\left[\overline{Y}_k \mathbb{1}_{\{p_k^0(\overline{\mathbf{X}}) \leq \frac{C_n}{n}\}} B\right]\\ &= \mathbb{E}_{\overline{\mathbf{X}}}\left[p_k^0(\overline{\mathbf{X}}) \mathbb{1}_{\{p_k^0(\overline{\mathbf{X}}) \leq \frac{C_n}{n}\}}\left|\log(p_k^0(\overline{\mathbf{X}}))\right|\right] + \mathbb{E}_{\overline{\mathbf{X}}}\left[p_k^0(\overline{\mathbf{X}}) \mathbb{1}_{\{p_k^0(\overline{\mathbf{X}}) \leq \frac{C_n}{n}\}} B\right],\end{aligned}$$

where the last equality follows from conditioning on $\overline{\mathbf{X}}$. Using that the function $u \mapsto u|\log(u)|$ is monotone increasing on $(0, e^{-1})$ and $n \geq eC_n$, yields

$$\left|\mathbb{E}_{\mathcal{D}_n,(\overline{\mathbf{X}},\overline{\mathbf{Y}})}\left[\overline{Y}_k \mathbb{1}_{\{p_k^0(\overline{\mathbf{X}}) \leq \frac{C_n}{n}\}}\Big(B \wedge \log\Big(\frac{p_k^0(\overline{\mathbf{X}})}{\widehat{p}_k(\overline{\mathbf{X}})}\Big)\Big)\right]\right| \leq \frac{C_n\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n}.$$

□

Corollary 2.4.7. *Under the conditions of Lemma 2.4.6 it holds that*

$$-\frac{C_n\log(n/C_n)}{n}\leq \mathbb{E}_{\mathcal{D}_n,(\overline{\mathbf{X}},\overline{\mathbf{Y}})}\left[\overline{Y}_k\mathbb{1}_{\{p_k^0(\overline{\mathbf{X}})\leq\frac{C_n}{n}\}}\Big(B\wedge\log\Big(\frac{p_k^0(\overline{\mathbf{X}})}{\widehat{p}_k(\overline{\mathbf{X}})}\Big)\Big)\right]\leq\frac{C_nB}{n}.$$

Proof. The lower and upper bound can be obtained from (2.4.2), $\overline{Y}_k\geq 0$ and the fact that $u\mapsto u\log(u)$ is monotone decreasing on $(0,e^{-1})$. □

Both Lemma 2.4.6 and Corollary 2.4.7 do not require that the random pair $(\overline{\mathbf{X}},\overline{\mathbf{Y}})$ is independent of the data. Specifically, they also hold in the case that $(\overline{\mathbf{X}},\overline{\mathbf{Y}})=(\mathbf{X}_i,\mathbf{Y}_i)$ for some $i\in\{1,\cdots,n\}$.

Proof of Theorem 2.3.5. For ease of notation set

$$\Big(B\wedge\log\Big(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\widehat{\mathbf{p}}(\mathbf{X}_i)}\Big)\Big)_{\geq C_n/n}$$

to denote the vector with coefficients

$$\mathbb{1}_{\{p_k^0(\mathbf{X}_i)\geq\frac{C_n}{n}\}}\Big(B\wedge\log\Big(\frac{p_k^0(\mathbf{X}_i)}{\widehat{p}_k(\mathbf{X}_i)}\Big)\Big),\quad k=1,\ldots,K.$$

For i.i.d. random pairs $(\widetilde{\mathbf{X}}_i,\widetilde{\mathbf{Y}}_i)$, $i=1,\cdots,n$ with joint distribution $\mathbb{P}$ that are generated independently of the data sample define $\mathcal{D}_n':=\{(\mathbf{X}_i,\mathbf{Y}_i)_i,(\widetilde{\mathbf{X}}_i,\widetilde{\mathbf{Y}}_i)_i\}$. Then, for any $C_n>0$,

$$\begin{aligned}&|R_B(\mathbf{p}_0,\widehat{\mathbf{p}})-R_{B,n}(\mathbf{p}_0,\widehat{\mathbf{p}})|\\&=\Bigg|\mathbb{E}_{\mathcal{D}_n'}\Bigg[\frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K\widetilde{Y}_{i,k}\Big(B\wedge\log\Big(\frac{p_k^0(\widetilde{\mathbf{X}}_i)}{\widehat{p}_k(\widetilde{\mathbf{X}}_i)}\Big)\Big)\\&\qquad-\frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K Y_{i,k}\Big(B\wedge\log\Big(\frac{p_k^0(\mathbf{X}_i)}{\widehat{p}_k(\mathbf{X}_i)}\Big)\Big)\Bigg]\Bigg|\\&\leq (I)+(II)+(III),\end{aligned}\tag{2.4.3}$$

where

$$(I)=\Bigg|\mathbb{E}_{\mathcal{D}_n'}\Bigg[\frac{1}{n}\sum_{i=1}^n\Bigg(\widetilde{\mathbf{Y}}_i^\top\Big(B\wedge\log\Big(\frac{\mathbf{p}_0(\widetilde{\mathbf{X}}_i)}{\widehat{\mathbf{p}}(\widetilde{\mathbf{X}}_i)}\Big)\Big)_{\geq C_n/n}$$

$$- \mathbf{Y}_i^\top \Big(B \wedge \log\Big(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\widehat{\mathbf{p}}(\mathbf{X}_i)}\Big)\Big)_{\geq C_n/n}\Big)\Big]\Big|$$

$$(II) = \left|\mathbb{E}_{\mathcal{D}_n'}\left[\frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K \widetilde{Y}_{i,k}\mathbb{1}_{\{p_k^0(\widetilde{\mathbf{X}}_i)\leq \frac{C_n}{n}\}}\Big(B \wedge \log\Big(\frac{p_k^0(\widetilde{\mathbf{X}}_i)}{\widehat{p}_k(\widetilde{\mathbf{X}}_i)}\Big)\Big)\right]\right|$$

$$(III) = \left|\mathbb{E}_{\mathcal{D}_n'}\left[\frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K Y_{i,k}\mathbb{1}_{\{p_k^0(\mathbf{X}_i)\leq \frac{C_n}{n}\}}\Big(B \wedge \log\Big(\frac{p_k^0(\mathbf{X}_i)}{\widehat{p}_k(\mathbf{X}_i)}\Big)\Big)\right]\right|.$$

First we bound the terms (II) and (III). Applying Lemma 2.4.6 in total nK times with $(\overline{\mathbf{X}}, \overline{\mathbf{Y}}) = (\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{Y}}_i)$, yields

$$(II) \leq \frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K \frac{C_n\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n} = \frac{C_nK\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n}, \tag{2.4.4}$$

while taking $(\overline{\mathbf{X}}, \overline{\mathbf{Y}}) = (\mathbf{X}_i, \mathbf{Y}_i)$ in Lemma 2.4.6 yields

$$(III) \leq \frac{1}{n}\sum_{i=1}^n\sum_{k=1}^K \frac{C_n\big(\log\big(\frac{n}{C_n}\big) + B\big) + B\big)}{n} = \frac{C_nK\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n}. \tag{2.4.5}$$

Now we deal with the term (I). Due to the bound B and the indicator function

$$\mathbb{1}_{\{p_k^0(\mathbf{X}_i)\geq \frac{C_n}{n}\}}\Big(B \wedge \log\Big(\frac{p_k^0(\mathbf{X}_i)}{\widehat{p}_k(\mathbf{X}_i)}\Big)\Big)$$
$$= \mathbb{1}_{\{p_k^0(\mathbf{X}_i)\geq \frac{C_n}{n}\}}\Big(B \wedge \log\Big(\frac{p_k^0(\mathbf{X}_i)}{(C_ne^{-B}/n) \vee \widehat{p}_k(\mathbf{X}_i)}\Big)\Big). \tag{2.4.6}$$

Given a minimal (internal) δ-covering of $\log(\mathcal{F})$ with respect to the pseudometric d_τ, with $\tau = \log(C_ne^{-B}/n)$, denote the centers of the balls by $\mathbf{p}_\ell$. Then there exists a random ℓ^* such that

$$\Big\|\log\Big(\frac{C_ne^{-B}}{n}\Big) \vee \log(\widehat{\mathbf{p}}) - \log\Big(\frac{C_ne^{-B}}{n}\Big) \vee \log(\mathbf{p}_{\ell^*})\Big\|_\infty \leq \delta.$$

This together with (2.4.6) and using that $\mathbf{Y}$ is one of the K-dimensional standard basis vectors yields

$$(I) \leq \mathbb{E}_{\mathcal{D}_n'}\left[\left|\frac{1}{n}\sum_{i=1}^n G_{\ell^*}(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{Y}}_i, \mathbf{X}_i, \mathbf{Y}_i)\right|\right] + 2\delta, \tag{2.4.7}$$

where

$$G_{\ell^*}(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{Y}}_i, \mathbf{X}_i, \mathbf{Y}_i) := \\ \widetilde{\mathbf{Y}}_i^\top \Big(B \wedge \log\Big(\frac{\mathbf{p}_0(\widetilde{\mathbf{X}}_i)}{\mathbf{p}_{\ell^*}(\widetilde{\mathbf{X}}_i)}\Big)\Big)_{\geq C_n/n} - \mathbf{Y}_i^\top \Big(B \wedge \log\Big(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_{\ell^*}(\mathbf{X}_i)}\Big)\Big)_{\geq C_n/n}. \quad (2.4.8)$$

For all $\ell \in \{1, \cdots, \mathcal{N}_n\}$ define G_ℓ in the same way. Moreover, write

$$\mathbf{Z}_i := (\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{Y}}_i, \mathbf{X}_i, \mathbf{Y}_i).$$

In a next step, we apply Bernstein's inequality (Proposition 2.C.1) to $(G_\ell(\mathbf{Z}_i))_{i=1}^n$. Using that $(\mathbf{X}_i, \mathbf{Y}_i)$ and $(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{Y}}_i)$ have the same distribution, we get for the expectation of G_ℓ that

$$\mathbb{E}_{\mathcal{D}'_n}[G_\ell(\mathbf{Z}_i)] = 0.$$

To verify the assumptions of Bernstein's inequality, it remains to prove that

$$\mathbb{E}|G_\ell(\mathbf{Z}_i)|^m \leq m!(2B)^{m-2} R_B(\mathbf{p}_0, \mathbf{p}_\ell) 32B2^{-1}, \ \forall m \in \mathbb{N}_{\geq 2}, \quad (2.4.9)$$

such that, in the notation of Proposition 2.C.1, we have $v_i = R_B(\mathbf{p}_0, \mathbf{p}_\ell)32B$ and $U = 2B$. To show this moment bound, observe that any real numbers a, b satisfy $|a + b|^m \leq 2^m(|a|^m + |b|^m)$. Using moreover that $(\mathbf{X}_i, \mathbf{Y}_i)$ and $(\widetilde{\mathbf{X}}_i, \widetilde{\mathbf{Y}}_i)$ have the same distribution, the m-th absolute moment of G_ℓ is given by

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}'_n}\big[|G_\ell(\mathbf{Z}_i)|^m\big] \\ &= \mathbb{E}_{\mathcal{D}'_n}\Bigg[\Bigg|\widetilde{\mathbf{Y}}_i^\top \left(B \wedge \log\left(\frac{\mathbf{p}_0(\widetilde{\mathbf{X}}_i)}{\mathbf{p}_\ell(\widetilde{\mathbf{X}}_i)}\right)\right)_{\geq C_n/n} \\ &\qquad - \mathbf{Y}_i^\top \left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)_{\geq C_n/n}\Bigg|^m\Bigg] \\ &\leq 2^{m+1}\mathbb{E}_{\mathcal{D}_n}\left[\left|\mathbf{Y}_i^\top \left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)_{\geq C_n/n}\right|^m\right]. \end{aligned}$$

Triangle inequality gives

$$\begin{aligned} &\mathbb{E}_{\mathcal{D}_n}\left[\left|\mathbf{Y}_i^\top \left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)_{\geq C_n/n}\right|^m\right] \\ &\leq \mathbb{E}_{\mathcal{D}_n}\left[\left(\mathbf{Y}_i^\top \left|\left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)_{\geq C_n/n}\right|\right)^m\right], \end{aligned}$$

where for a vector $\mathbf{v}$, $|\mathbf{v}|$ denotes the absolute value coefficient-wise. Since $\mathbf{Y}$ is one of the standard basis vectors, it holds that $Y_k \in \{0,1\}$, and Y_kY_j is equal to 0 when $j \neq k$ and equal to Y_k when $k = j$. Using this observation together with conditioning on $\mathbf{X}_i$ yields

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_n}\left[\left(\mathbf{Y}_i^\top \left|\left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)_{\geq C_n/n}\right|\right)^m\right] \\
&= \mathbb{E}_{\mathcal{D}_n}\left[\mathbf{Y}_i^\top \left|\left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)_{\geq C_n/n}\right|^m\right] \\
&= \mathbb{E}_{\mathbf{X}_i}\left[\mathbf{p}_0^\top(\mathbf{X}_i) \left|\left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)_{\geq C_n/n}\right|^m\right] \\
&\leq \mathbb{E}_{\mathbf{X}_i}\left[\mathbf{p}_0^\top(\mathbf{X}_i) \left|B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right|^m\right],
\end{aligned}$$

where we used for the last inequality that for every set Ω, each $A \subseteq \Omega$, every function $\theta : \Omega \to \mathbb{R}$ and every $m \in \mathbb{N}_{\geq 2}$ it holds that $|\mathbb{1}_A \theta|^m = (\mathbb{1}_A)^m |\theta|^m = \mathbb{1}_A |\theta|^m \leq |\theta|^m$. Combining the previous displays and applying Lemma 2.3.7, we get that

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_n'}[|G_\ell(\mathbf{Z}_i)|^m] \\
&\leq 2^{m+1}\mathbb{E}_{\mathbf{X}_i}\left[\mathbf{p}_0^\top(\mathbf{X}_i) \left|B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right|^m\right] \\
&\leq 2^{m+1}C_{m,B}\mathbb{E}_{\mathbf{X}_i}\left[\mathbf{p}_0^\top(\mathbf{X}_i) \left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X}_i)}{\mathbf{p}_\ell(\mathbf{X}_i)}\right)\right)\right] = 2^{m+1}C_{m,B}R_B(\mathbf{p}_0, \mathbf{p}_\ell),
\end{aligned} \tag{2.4.10}$$

where $C_{m,B}$ is given by

$$C_{m,B} = \max\left\{m!, \frac{B^m}{B-1}\right\}.$$

Since $B \geq 2$, we get that $B/(B-1) \leq 2$ and $C_{m,B} \leq \max\{m!, 2B^{m-1}\} \leq 2m!B^{m-1}$. Together with (2.4.10) this yields

$$\mathbb{E}_{\mathcal{D}_n'}[|G_\ell(\mathbf{Z}_i)|^m] \leq 2^{m+1}C_{m,B}R_B(\mathbf{p}_0, \mathbf{p}_\ell) \leq m!(2B)^{m-2}R_B(\mathbf{p}_0, \mathbf{p}_\ell)32B2^{-1},$$

completing the proof for the moment bound (2.4.9).

Now define $z_\ell := \sqrt{n^{-1}68B\log(\mathcal{N}_n)} \vee \sqrt{\mathbb{E}_{(\mathbf{X},\mathbf{Y})}[\mathbf{Y}^\top(B \wedge \log(\mathbf{p}_0(\mathbf{X})/\mathbf{p}_\ell(\mathbf{X})))]}$. Since $B \geq 2$, Lemma 2.3.4 guarantees that the truncated Kullback-Leibler risk is always nonnegative, so z_ℓ is well defined. Define $z^* = z_{\ell^*}$, that is,

$$z^* = \sqrt{\frac{68B\log(\mathcal{N}_n)}{n}} \vee \sqrt{\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\mathbf{Y}^\top\left(B \wedge \log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\mathbf{p}_{\ell^*}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]},$$

where we also condition on the dataset $\mathcal{D}_n$. To upper bound z^*, we split the truncated empirical risk

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\mathbf{Y}^\top\left(B\wedge\log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\mathbf{p}_{\ell^*}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]\\
&=\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\sum_{k=1}^K Y_k\mathbb{1}_{\{p_k^0(\mathbf{X})\leq\frac{C_n}{n}\}}\left(B\wedge\log\left(\frac{p_k^0(\mathbf{X})}{p_{\ell^*,k}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]\\
&+\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\sum_{k=1}^K Y_k\mathbb{1}_{\{p_k^0(\mathbf{X})\geq\frac{C_n}{n}\}}\left(B\wedge\log\left(\frac{p_k^0(\mathbf{X})}{p_{\ell^*,k}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right].
\end{aligned}$$

Using the property of the δ-cover, Equation (2.4.6) and the fact that $\mathbf{Y}$ is a standard basis vector, it holds that

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\sum_{k=1}^K Y_k\mathbb{1}_{\{p_k^0(\mathbf{X})\geq\frac{C_n}{n}\}}\left(B\wedge\log\left(\frac{p_k^0(\mathbf{X})}{p_{\ell^*,k}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]\\
&\leq\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\sum_{k=1}^K Y_k\mathbb{1}_{\{p_k^0(\mathbf{X})\geq\frac{C_n}{n}\}}\left(B\wedge\log\left(\frac{p_k^0(\mathbf{X})}{\widehat{p}_k(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]+\delta.
\end{aligned}$$

On the other hand, applying Corollary 2.4.7, with $(\overline{\mathbf{X}},\overline{\mathbf{Y}})=(\mathbf{X},\mathbf{Y})$, K times for $\widehat{\mathbf{p}}$ and K times with $\widehat{\mathbf{p}}$ replaced by $\mathbf{p}_{\ell^*}$, yields

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\sum_{k=1}^K Y_k\mathbb{1}_{\{p_k^0(\mathbf{X})\leq\frac{C_n}{n}\}}\left(B\wedge\log\left(\frac{p_k^0(\mathbf{X})}{p_{\ell^*,k}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]\\
&\leq\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\sum_{k=1}^K Y_k\mathbb{1}_{\{p_k^0(\mathbf{X})\leq\frac{C_n}{n}\}}\left(B\wedge\log\left(\frac{p_k^0(\mathbf{X})}{\widehat{p}_k(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]\\
&+\frac{C_nK\big(\log\big(\frac{n}{C_n}\big)+B\big)}{n}.
\end{aligned}$$

Define

$$V:=\sqrt{\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\mathbf{Y}^\top\left(B\wedge\log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\widehat{\mathbf{p}}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]}.$$

Combining the previous inequalities, we get that

$$\sqrt{\mathbb{E}_{\mathcal{D}_N,(\mathbf{X},\mathbf{Y})}\left[\mathbf{Y}^\top(B\wedge\log\left(\frac{\mathbf{p}_0(\mathbf{X})}{\mathbf{p}_{\ell^*}(\mathbf{X})}\right)\right)\middle|\mathcal{D}_n\right]}\leq V+\sqrt{\delta+\frac{C_nK\big(\log\big(\frac{n}{C_n}\big)+B\big)}{n}},$$

where we also used the elementary inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for all $a, b \geq 0$. Hence,

$$z^* \leq \sqrt{\frac{68B\log(\mathcal{N}_n)}{n}} + V + \sqrt{\delta + \frac{C_nK\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n}}. \tag{2.4.11}$$

The term $\sqrt{n^{-1}68B\log(\mathcal{N}_n)}$ is chosen such that in (2.4.13) and (2.4.14) below the equations balance out. Now define

$$T := \max_{\ell} \left| \sum_{i=1}^{n} \frac{G_\ell(\mathbf{Z}_i)}{z_\ell} \right|.$$

The Cauchy-Schwarz inequality gives us that $\mathbb{E}_{\mathcal{D}'_n}[VT] \leq \sqrt{\mathbb{E}_{\mathcal{D}'_n}[V^2]\mathbb{E}_{\mathcal{D}'_n}[T^2]}$. Noticing that $\mathbb{E}_{\mathcal{D}'_n}[V^2] = R_B(\mathbf{p}_0, \widehat{\mathbf{p}})$, we get from (2.4.3), (2.4.4), (2.4.5), (2.4.7) and (2.4.11) that

$$\begin{aligned}
&|R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) - R_{B,n}(\mathbf{p}_0, \widehat{\mathbf{p}})| \\
&\leq \frac{1}{n}\sqrt{R_B(\mathbf{p}_0, \widehat{\mathbf{p}})}\sqrt{\mathbb{E}_{\mathcal{D}'_n}[T^2]} \\
&+ \frac{1}{n}\left(\sqrt{\frac{68B\log(\mathcal{N}_n)}{n}} + \sqrt{\delta + \frac{C_nK\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n}}\right)\mathbb{E}_{\mathcal{D}'_n}[T] \\
&+ 2\delta + \frac{2C_nK\big(\log\big(\frac{n}{C_n}\big) + B\big)}{n}.
\end{aligned} \tag{2.4.12}$$

The next step in the proof derives bounds on $\mathbb{E}_{\mathcal{D}'_n}[T]$ and $\mathbb{E}_{\mathcal{D}'_n}[T^2]$. Using an union bound it holds that

$$\begin{aligned}
\mathbb{P}(T \geq t) &= \mathbb{P}\left(\max_{\ell} \left| \sum_{i=1}^{n} \frac{G_\ell(\mathbf{Z}_i)}{z_\ell} \right| \geq t\right) = \mathbb{P}\left(\bigcup_{\ell=1}^{\mathcal{N}_n} \left(\left| \sum_{i=1}^{n} \frac{G_\ell(\mathbf{Z}_i)}{z_\ell} \right| \geq t \right)\right) \\
&\leq \sum_{\ell=1}^{\mathcal{N}_n} \mathbb{P}\left(\left| \sum_{i=1}^{n} G_\ell(\mathbf{Z}_i) \right| \geq tz_\ell \right).
\end{aligned}$$

We already showed that $G_\ell(\mathbf{Z}_i)$ satisfies the conditions of Bernstein's inequality (Proposition 2.C.1) with $v_i = R_B(\mathbf{p}_0, \mathbf{p}_\ell)32B$ and $U = 2B$. Bernstein's inequality applied to the last term gives

$$\begin{aligned}\mathbb{P}\left(T \geq t\right) &\leq \sum_{\ell=1}^{\mathcal{N}_n} \mathbb{P}\left(\left|\sum_{i=1}^{n} G_\ell(\mathbf{Z}_i)\right| \geq t z_\ell\right) \\ &\leq \sum_{\ell=1}^{\mathcal{N}_n} 2\exp\left(-\frac{(tz_\ell)^2}{2nR_B(\mathbf{p}_0,\mathbf{p}_\ell)32B + 4Btz_\ell}\right) \\ &= 2\mathcal{N}_n \exp\left(-\frac{t^2}{2n\frac{R_B(\mathbf{p}_0,\mathbf{p}_\ell)32B}{z_\ell^2} + 4B\frac{t}{z_\ell}}\right).\end{aligned}$$

Since $z_\ell \geq \sqrt{R_B(\mathbf{p}_0,\mathbf{p}_\ell)}$ it holds that $z_\ell^2 \geq R_B(\mathbf{p}_0,\mathbf{p}_\ell)$. As probabilities are in the interval $[0,1]$, this gives us that

$$\mathbb{P}\left(T \geq t\right) \leq 1 \wedge 2\mathcal{N}_n \exp\left(-\frac{t^2}{64Bn + 4B\frac{t}{z_\ell}}\right).$$

If $t \geq \sqrt{68Bn\log(\mathcal{N}_n)}$, then since $z_\ell \geq \sqrt{n^{-1}68B\log(\mathcal{N}_n)}$ it holds that

$$\exp\left(-\frac{t^2}{64Bn + 4B\frac{t}{z_\ell}}\right) \leq \exp\left(-\frac{t\sqrt{\log(\mathcal{N}_n)}}{\sqrt{68Bn}}\right).$$

For every nonnegative random variable X with finite expectation one has $\mathbb{E}[X] = \int_0^\infty \mathbb{P}(X \geq t)\,dt$. Therefore,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n'}[T] &\leq \sqrt{68Bn\log(\mathcal{N}_n)} + \int_{\sqrt{68Bn\log(\mathcal{N}_n)}}^{\infty} 2\mathcal{N}_n \exp\left(-\frac{t\sqrt{\log(\mathcal{N}_n)}}{\sqrt{68Bn}}\right) dt \\ &= \sqrt{68Bn\log(\mathcal{N}_n)} + \sqrt{\frac{272Bn}{\log(\mathcal{N}_n)}}.\end{aligned} \tag{2.4.13}$$

Since T is nonnegative, $\mathbb{P}(T^2 \geq u) = \mathbb{P}(T \geq \sqrt{u})$, so using the same arguments as before we get that

$$\mathbb{E}_{\mathcal{D}_n'}[T^2] \leq 68Bn\log(\mathcal{N}_n) + \int_{68Bn\log(\mathcal{N}_n)}^{\infty} 2\mathcal{N}_n \exp\left(-\sqrt{\frac{u\log(\mathcal{N}_n)}{68Bn}}\right) du.$$

Substitution $s = \sqrt{u}$ and integration by parts gives us that $(1/2)\int_a^\infty e^{-\sqrt{u}b}\,du = \int_{\sqrt{a}}^\infty se^{-sb}\,ds = (\sqrt{a}b+1)e^{-\sqrt{a}b}/b^2$ and consequently

$$\mathbb{E}_{\mathcal{D}_n'}[T^2] \leq 68Bn\log(\mathcal{N}_n) + 544Bn, \tag{2.4.14}$$

where we also used that $\mathcal{N}_n \geq e$ and thus $(\log(\mathcal{N}_n)+1)/\log(\mathcal{N}_n) \geq 2$.

Combining (2.4.13), (2.4.14) with (2.4.12), using twice that $2xy \leq x^2 + y^2$ for all real numbers x, y, and using that $\log(\mathcal{N}_n) \geq 1$, we get that

$$\begin{aligned} |R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) - R_{B,n}(\mathbf{p}_0, \widehat{\mathbf{p}})| \leq \sqrt{R_B(\mathbf{p}_0, \widehat{\mathbf{p}})}\sqrt{\frac{68B\log(\mathcal{N}_n)+544B}{n}} + 3\delta \\ + \frac{102B\log(\mathcal{N}_n)+272B}{n} + \frac{3C_nK\big(\log\big(\frac{n}{C_n}\big)+B\big)}{n}. \end{aligned} \tag{2.4.15}$$

Setting $a = R_B(\mathbf{p}_0, \widehat{\mathbf{p}})$, $b = R_{B,n}(\mathbf{p}_0, \widehat{\mathbf{p}})$,

$$c = \sqrt{\frac{17B\log(\mathcal{N}_n)+134B}{n}},$$

and

$$d = \frac{102B\log(\mathcal{N}_n)+272B+3C_nK\big(\log\big(\frac{n}{C_n}\big)+B\big)}{n} + 3\delta,$$

we get from (2.4.15) that $|a-b| \leq 2\sqrt{ac}+d$. Since the excess risk is always nonnegative we can apply Proposition 2.C.2. This gives us for any $0 < \epsilon \leq 1$

$$\begin{aligned} R_B(\mathbf{p}_0, \widehat{\mathbf{p}}) &\leq (1+\epsilon)\left(R_{B,n}(\mathbf{p}_0, \widehat{\mathbf{p}}) + 3\delta\right) \\ &+ (1+\epsilon)\left(\frac{102B\log(\mathcal{N}_n)+272B+3C_nK\big(\log\big(\frac{n}{C_n}\big)+B\big)}{n}\right) \\ &+ \frac{(1+\epsilon)^2}{\epsilon}\cdot\frac{17B\log(\mathcal{N}_n)+136B}{n}. \end{aligned}$$

Proposition 2.4.5 gives $R_{B,n}(\mathbf{p}_0, \widehat{\mathbf{p}}) \leq \inf_{\mathbf{p}\in\mathcal{F}} R(\mathbf{p}_0, \mathbf{p}) + \Delta_n(\mathbf{p}_0, \widehat{\mathbf{p}})$. Substituting this in the previous equation and observing that $(1+\epsilon)/\epsilon \geq 2$, $1/\epsilon \geq 1$ and $0 < 1-\epsilon \leq 1$ for $\epsilon \in (0, 1]$ yields the assertion of the theorem. □

Appendix Chapter 2

2.A Basic network properties and operations

In this section we state elementary properties of network classes and introduce small networks that are capable of approximating multiplication operations based on similar results in [127].

2.A.1 Embedding properties of neural network function classes

This section extends the results in [127] to arbitrary output activation function.

Enlarging: Let $\mathbf{m}$ and $\mathbf{m}'$ be two width-vectors of the same length and let $s, s' > 0$. If $\mathbf{m} \leq \mathbf{m}'$ component-wise, $m_{L+1} = m'_{L+1}$ and $s \leq s'$, then

$$\mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m}, s) \subseteq \mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m}', s'). \tag{2.A.1}$$

This rule allows us to simplify the neural network architectures. For example we can simplify a network class by embedding it in a class for which all hidden layers have the same width.

Composition: Let $\mathbf{f} \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s_1)$ and let $\mathbf{g}$ be a network in $\mathcal{F}_{\boldsymbol{\psi}}(L', \mathbf{m}', s_2)$, with $m_{L+1} = m'_0$. For a vector $\mathbf{v} \in \mathbb{R}^{m_{L+1}}$, define the composed network $\mathbf{g} \circ \sigma_{\mathbf{v}}(\mathbf{f})$. Then

$$\mathbf{g} \circ \sigma_{\mathbf{v}}(\mathbf{f}) \in \mathcal{F}_{\boldsymbol{\psi}}\big(L + L' + 1, (m_0, \cdots, m_{L+1}, m'_1, \cdots, m'_{L'+1}), s_1 + s_2 + |\mathbf{v}|_0\big). \tag{2.A.2}$$

The following rule allows us to synchronize the depths of neural networks.

Depth synchronization: For any positive integer a,

$$\mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m}, s) \subset \mathcal{F}_{\boldsymbol{\psi}}(L + a, (\underbrace{m_0, \cdots, m_0}_{a \text{ times}}, \mathbf{m}), s + am_0). \tag{2.A.3}$$

To identify simple neural network architectures, we can combine the depth synchronization and enlarging properties. When there exist $c \geq m_0$ and $b > 0$, such that

$s = cL + b$, and L^* is an upper bound on L, combining the previous two properties yields

$$\mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m}, s) \subset \mathcal{F}_{\boldsymbol{\psi}}(L^*, \mathbf{m}', cL + m_0(L^* - L) + b) \subset \mathcal{F}_{\boldsymbol{\psi}}(L^*, \mathbf{m}', cL^* + b),$$

where the width vector $\mathbf{m}'$ has length $L^* + 2$ and can be chosen as $(m_0, m', m', \cdots m', m_{L+1})$ with m' equal to the largest coefficient of $\mathbf{m}$.

Parallelization: Let $\mathbf{m}$, $\mathbf{m}'$ be two width vectors such that $m_0 = m_0'$ and let $\mathbf{f} \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m})$ and $\mathbf{g} \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}')$. Define the parallelized network $\mathbf{h}$ as $\mathbf{h} := (\mathbf{f}, \mathbf{g})$. Then

$$\mathbf{h} \in \mathcal{F}_{\mathrm{id}}(L, (m_0, m_1 + m_1', \cdots, m_{L+1} + m_{L+1}'). \tag{2.A.4}$$

Proposition 2.A.1 (Removal of inactive nodes)**.** *It holds that*

$$\mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m}, s) = \mathcal{F}_{\boldsymbol{\psi}}(L, (m_0, m_1 \wedge s, \cdots, m_L \wedge s, m_{L+1}), s).$$

For this property, the output function plays no role and the proof in [127] carries over.

The following equation gives the number of parameters in a fully connected network in $\mathcal{F}_{\boldsymbol{\psi}}(L, \mathbf{m})$:

$$\sum_{j=0}^{L} (m_j + 1)m_{j+1} - m_{L+1}. \tag{2.A.5}$$

This will be used further on as an upper bound on the number of active parameters in sub-networks.

2.A.2 Scaling numbers

We constraint all neural network parameters to be bounded in absolute value by one. To build neural networks with large output values we construct small rescaling networks.

Proposition 2.A.2. *For any real number C there exists a network* $\mathrm{Scale}_C \in \mathcal{F}_{\mathrm{id}}(\lceil \log_2(|C|)\rceil + (\lceil \log_2(|C|)\rceil - 1), (1, 2, 1, 2, 1, \cdots, 1, 2, 1), 4\lceil \log_2(|C|)\rceil)$ *such that* $\mathrm{Scale}_C(x) = C(x)_+$.

Proof. Set

$$W_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mathbf{v}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \text{and} \quad W_1 = (1, 1).$$

The network $W_1 \sigma_{\mathbf{v}_1} W_0 x$ computes $x \mapsto 2(x)_+$. This network has exactly one hidden layer, one input node, one output node and two nodes in the hidden layer. It uses four nonzero-parameters. Composing $\lceil \log_2(|C|) \rceil$ of these networks, using the composition rule (2.A.2), where we take the output layer of one network to be the input layer of the next one with shift vector zero, yields a network in the right network class computing $x \mapsto 2^{\lceil \log_2(|C|) \rceil}(x)_+$. Replacing the last weight matrix by $(C2^{-\lceil \log_2(|C|) \rceil}, C2^{-\lceil \log_2(|C|) \rceil})$ yields the result. □

2.A.3 Negative numbers

For negative input, the ReLU activation without shift returns zero. As a result, many network constructions output zero for negative input. Using that $x = \sigma(x) - \sigma(-x)$, the next result shows existence of a neural network function that extends the original network function as an even (or odd) function to negative input values.

Proposition 2.A.3. *Assume $f \in \mathcal{F}_{\mathrm{id}}(L, (m_0, m_1, \cdots, m_L, 1), s)$ and $f(\mathbf{x}) = 0$ whenever $x_j \leq 0$ for some index $j \in \{1, \cdots, m_0\}$. Then there exist neural networks*

$$f^{\pm} \in \mathcal{F}_{\mathrm{id}}(L, (m_0, 2m_2, \cdots, 2m_L, 1), 2s),$$

such that $x_j \mapsto f^+(\mathbf{x})$ is an even function, $x_j \mapsto f^-(\mathbf{x})$ is an odd function and $f^{\pm}(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x}$ with $x_j \geq 0$.

Proof. Take two neural networks in the class $\mathcal{F}_{\mathrm{id}}(L, (m_0, m_1, \cdots, m_L, 1), s)$ in parallel: The original network f to deal with the positive part and the second network to deal with the negative part. This second network can be build from the first network f by multiplying the j-th column vector of W_0 by -1 and multiplying the output of the network by ± 1. The parallelized network computes then $f^{\pm}$. □

The extension to more than one output is straightforward. Following the same construction as in the previous section, all that has to be done is multiplying the corresponding rows of the weight matrix in the output layer of the neural network by either -1, 1 of 0 depending on how we wish to extend the function. More precisely, if we have $m_0^- \leq m_0$ input coefficients x_j for which $x_j \leq 0$ implies $f(\mathbf{x}) = 0$, we can find neural networks

$$\mathbf{f}^{\pm} \in \mathcal{F}_{\mathrm{id}}(L, (m_0, 2^{m_0^-} m_2, \cdots, 2^{m_0^-} m_L, m_{L+1}), 2^{m_0^-} s),$$

such that $x_j \mapsto \mathbf{f}^+(\mathbf{x})$ is an even function and $x_j \mapsto \mathbf{f}^-(\mathbf{x})$ is an odd function for all of the m_0^- indices j. This network can be constructed using $2^{m_0^-}$ parallel networks.

2.B Neural networks approximating the logarithm

Theorem 2.4.1 assumes $M \geq 2$. We use this throughout the proof without further mentioning.

2.B.1 Taylor approximation

Set

$$T_c^\kappa(x) = \log(c) + \sum_{\gamma=0}^{\kappa} x^\gamma \sum_{\alpha=\gamma\vee 1}^{\kappa} \binom{\alpha}{\gamma} \frac{c^{-\gamma}(-1)^{1-\gamma}}{\alpha} = \sum_{\gamma=0}^{\kappa} x^\gamma c_\gamma.$$

Proposition 2.B.1. *For all $\kappa = 0, 1, \dots$ and every $c > 0$, we have that*

$$\big|\log(x) - T_c^\kappa(x)\big| \leq \frac{1}{\kappa+1}\left|\frac{x-c}{x \wedge c}\right|^{\kappa+1},$$

where the sum in T_c^κ is defined as zero if $\kappa = 0$. Moreover, if $0 < x \leq c$, we also have that $T_c^\kappa(x) \leq \log(c)$.

Proof. We claim that T_c^κ is equal to the k-th order Taylor approximation of the logarithm. First we show that from this claim the statements of the proposition follow. The α-th derivative of the logarithm is $\log^{(\alpha)}(x) = (\alpha-1)!(-1)^{\alpha+1}x^{-\alpha}$. Thus, the k-th order Taylor approximation of the logarithm around the point c is given by

$$\log(c) + \sum_{\alpha=1}^{\kappa} \frac{(x-c)^\alpha(-1)^{\alpha+1}}{\alpha c^\alpha}. \tag{2.B.1}$$

By the mean value theorem, the remainder is bounded by

$$\frac{1}{\kappa+1}\left|\frac{x-c}{s}\right|^{\kappa+1},$$

for some s between x and c. Now since the function $1/s$ on $(0,\infty)$ is decreasing, its maximum is obtained at the left boundary, that is, $x \wedge c$, which yields the first claim of the proposition. Now we show that $T_c^\kappa \leq \log(c)$ whenever $0 < x \leq c$. When $\kappa = 0$, the sum in (2.B.1) disappears and the result follows immediately. When $\kappa \geq 1$, notice that $(x-c)$ is always negative. Hence the product $(x-c)^\alpha(-1)^{\alpha+1}$ is negative for all α, so together with the case $\kappa = 0$ this yields $T_c^\kappa(x) \leq \log(c)$, for $0 < x \leq c$.

It remains to prove that T_c^κ is the k-th order Taylor approximation of the logarithm around the point c. Writing the Taylor approximation as a linear combination of monomials gives

$$\log(c) + \sum_{\alpha=1}^{\kappa} \frac{(x-c)^\alpha(-1)^{\alpha+1}}{\alpha c^\alpha} = \sum_{\gamma=0}^{\kappa} x^\gamma \bar{c}_\gamma,$$

for suitable coefficients $\bar{c}_\gamma$. Using this expression we can obtain the coefficients $\bar{c}_\gamma$ for $\gamma \geq 1$ by evaluating the derivatives at $x = 0$:

$$\frac{d^\gamma}{dx^\gamma} \log(c) + \sum_{\alpha=1}^{\kappa} \frac{(x-c)^\alpha (-1)^{\alpha+1}}{\alpha c^\alpha} \Bigg|_{x=0} = \gamma! \bar{c}_\gamma.$$

This gives us that

$$\bar{c}_\gamma = \sum_{\alpha=\gamma}^{\kappa} \frac{(\alpha-1)!(-c)^{\alpha-\gamma}(-1)^{\alpha+1}}{\gamma!(\alpha-\gamma)!c^\alpha} = \sum_{\alpha=\gamma}^{\kappa} \binom{\alpha}{\gamma} \frac{c^{-\gamma}(-1)^{1-\gamma}}{\alpha}.$$

For $\bar{c}_0$ we get

$$\bar{c}_0 = \log(c) + \sum_{\alpha=1}^{\kappa} \frac{(\alpha-1)!(-c)^{\alpha}(-1)^{\alpha+1}}{(\alpha)!c^\alpha} = \log(c) + \sum_{\alpha=1}^{\kappa} \frac{(-1)}{\alpha}.$$

Hence $\sum_\gamma^\kappa x^\gamma \bar{c}_\gamma = \sum_\gamma^\kappa x^\gamma c_\gamma = T_c^\kappa(x)$, proving the claim. □

Next we establish a bound on the sum of the coefficients c_γ of T_c^κ in the case $c \leq e$. For $\gamma \geq 1$, we bound c_γ by

$$|c_\gamma| \leq \sum_{\alpha=\gamma}^{\kappa} \binom{\alpha}{\gamma} \frac{(1 \wedge c)^{-\gamma}}{\alpha} \leq (1 \wedge c)^{-\kappa} \sum_{\alpha=\gamma}^{\kappa} \binom{\alpha}{\gamma}.$$

Since also

$$|c_0| \leq |\log(c)| + \sum_{\alpha=1}^{\kappa} \frac{1}{\alpha} \leq |\log(c)| + \sum_{\alpha=1}^{\kappa} \binom{\alpha}{0},$$

this shows that the sum of the coefficients is bounded by

$$\sum_{\gamma=0}^{\kappa} |c_\gamma| \leq |\log(c)| + (1 \wedge c)^{-\kappa} \sum_{\gamma=0}^{\kappa} \sum_{\alpha=1 \wedge \gamma}^{\kappa} \binom{\alpha}{\gamma} \leq |\log(c)| + (1 \wedge c)^{-\kappa} \sum_{\gamma=0}^{\kappa} \sum_{\alpha=\gamma}^{\kappa} \binom{\alpha}{\gamma}.$$

The double sum can be rewritten as the sum of all the entries in the rows $0, \cdots, \kappa$ of Pascal's triangle. From the binomial theorem we know that summing over the α-th row of Pascal's triangle gives 2^α. Combined with $|\log(c)| \leq (1 \wedge c)^{-1}$ for $0 < c \leq e$, this gives

$$\sum_{\gamma=0}^{\kappa} |c_\gamma| \leq (\kappa+1)2^{\kappa+1}(1 \wedge c)^{-(\kappa \vee 1)} \leq (\kappa+1)2^{\kappa+1}(1 \wedge c)^{-\kappa-1}, \text{ for all } 0 < c \leq e. \quad (2.B.2)$$

Applying the softmax function to an approximation g of the logarithm involves the exponential function and requires a bound for $|e^{g(x)} - x|$ with $x > 0$. By the mean value theorem $|e^{g(x)} - e^{\log(x)}| = e^s|g(x) - \log(x)|$ for a suitable s between $\log(x)$ and $g(x)$. The next proposition provides such a bound.

Proposition 2.B.2. *For all $\lambda \geq 1$, define*

$$\mathcal{D}_\lambda := \left[\frac{\lambda^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M}, \frac{(\lambda+1)^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M}\right].$$

If $[a,b] \subset \mathcal{D}_\lambda$, then it holds for any $x \in [a,b]$ and any $\omega \leq \log\left(\frac{(\lambda+1)^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M}\right)$, that

$$e^\omega|T_b^{\lfloor\beta\rfloor}(x) - \log(x)| \leq \frac{1}{M}.$$

Proof. First notice that on $(0,\infty)$ the logarithm is strictly increasing and is infinitely times continuously differentiable. For real numbers a, b and a positive integer j, $a^j - b^j = (a-b)\sum_{i=1}^j a^{j-i}b^{i-1}$. Applied to $a = \lambda+1$ and $b = \lambda$, this gives $(\lambda+1)^j - \lambda^j \leq j(\lambda+1)^{j-1}$ and thus for $x \in [a,b] \subseteq \mathcal{D}_\lambda$, we get that

$$|x-b| \leq b-a \leq \frac{(\lambda+1)^{\lceil\beta\rceil} - \lambda^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M} \leq b\frac{\lceil\beta\rceil}{\lambda+1}.$$

Substituting this in the bound from Proposition 2.B.1 and using that $x \geq a$ gives

$$|T_b^{\lfloor\beta\rfloor}(x) - \log(x)| \leq \frac{1}{\lceil\beta\rceil}\left|\frac{\lceil\beta\rceil(\lambda+1)^{\lfloor\beta\rfloor}}{a2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M}\right|^{\lceil\beta\rceil}.$$

Since $a \in \mathcal{D}_\lambda$,

$$\begin{aligned}|T_b^{\lfloor\beta\rfloor}(x) - \log(x)| &\leq \frac{1}{\lceil\beta\rceil}\left|\frac{\lceil\beta\rceil(\lambda+1)^{\lfloor\beta\rfloor}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M}\cdot\frac{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M}{\lambda^{\lceil\beta\rceil}}\right|^{\lceil\beta\rceil} \\ &= \lceil\beta\rceil^{\lfloor\beta\rfloor}\left|\frac{(\lambda+1)^{\lfloor\beta\rfloor)}}{\lambda^{\lceil\beta\rceil}}\right|^{\lceil\beta\rceil}.\end{aligned}$$

Multiplying both sides with an exponential, noticing that the exponential function is strictly increasing, and applying the upper bound on ω given in the statement of the proposition yields

$$\begin{aligned}e^\omega|T_b^{\lfloor\beta\rfloor}(x) - \log(x)| &\leq \frac{(\lambda+1)^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M}\left|\frac{(\lambda+1)^{\lfloor\beta\rfloor)}}{\lambda^{\lceil\beta\rceil}}\right|^{\lceil\beta\rceil} \\ &= \frac{1}{2^{\lceil\beta\rceil^2}M}\left(\frac{\lambda+1}{\lambda}\right)^{\lceil\beta\rceil^2}.\end{aligned}$$

Since $(\lambda+1)\lambda^{-1}$ is positive and decreasing for $\lambda \geq 1$, we can upper bound the last display by $1/M$. □

2.B.2 Partition of unity

So far we have bounded the approximation error on subintervals. As we work with ReLU functions, indicator functions of intervals are impractical to use, because they are discontinuous. Instead we create a partition of unity consisting of continuous piecewise linear functions for an interval that contains the interval $[M^{-1}, 1-M^{-1}]$.

Define R as the smallest integer sucht that

$$\frac{(\frac{R}{2} + 2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil} - \frac{3}{4})^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M} \geq 1 - \frac{1}{M}.$$

Rewriting this equation yields

$$\begin{aligned} R &= \lceil * \rceil 2^{\lceil\beta\rceil+1}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil}(M-1)^{\frac{1}{\lceil\beta\rceil}} - 2\left(2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil} - \frac{3}{4}\right) \\ &\leq 2^{\lceil\beta\rceil+1}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil}M^{\frac{1}{\lceil\beta\rceil}}. \end{aligned}$$

Now we define sequences $(a_r)_{r=1,\cdots,R}$ and $(b_r)_{r=1,\cdots,R-1}$ as follows

$$a_r := \frac{(2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil} + \frac{r}{2} - \frac{3}{4})^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M},$$

$$b_r := \frac{(2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil} + \frac{r}{2} - \frac{1}{2})^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M},$$

and for ease of notation define $b_0 = a_1$ and $b_R = a_R$. Notice that $[M^{-1}, 1-M^{-1}] \subseteq [a_1, a_R] \subseteq [M^{-1}, 1+M^{-1}]$.

Next we define a family of functions $(F_r)_{r=2,3,\cdots,R}$ and $(H_r)_{r=1,2,\cdots,R}$ on the interval $[a_1, a_R]$. For $r = 2, \cdots, R$ define the function F_r to be zero outside of the interval $[a_{r-1}, a_r]$ and to be a linear interpolation between the value one at the point b_{r-1} and the value zero at the boundaries of this interval. In the same way define for $r = 2, \cdots, R-1$ the function H_r, but with support on the interval $[b_{r-1}, b_r]$ and with interpolation point a_r. Define H_1 to be the linear interpolation between the value one at the point a_1 and the value zero at b_1 and let it be zero outside this interval. Finally define H_R as the linear interpolation between the value one at the point b_R and the value zero at b_{R-1} and set it to zero outside of this interval.

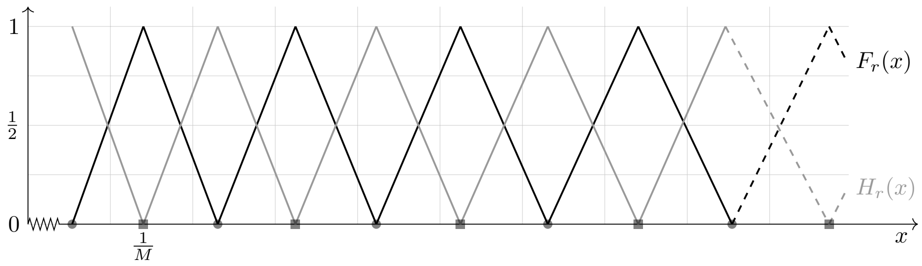


Figure 2.B.1: The first few functions $F_r(x)$ and $H_r(x)$ when $\beta \in (1, 2]$. The points a_r are marked with circles, while the points b_r are denoted by squares.

By construction it holds that

$$\sum_{r=2}^{R} F_r(x) + \sum_{r=1}^{R} H_r(x) = 1, \quad \text{for all } x \in [a_1, a_R].$$

Figure 2.B.1 gives the first few functions F_r and H_r in the case that $\beta \in (1, 2]$.

We can construct a ReLU network that exactly represents the functions F_r and H_r. This construction is a modification of the construction of continuous piecewise linear functions as used in [160]. This modification assures that the parameters are bounded by one.

Proposition 2.B.3. *For each function F_r and H_r their exists a network $U_{F_r}, U_{H_r} \in \mathcal{F}_{\text{id}}(L, \mathbf{m}, s)$, with $L = 3((1 + \lceil\beta\rceil)^2 + \lfloor\log_2(M\lceil\beta\rceil^{\lfloor\beta\rfloor})\rfloor)$, $\mathbf{m} = (1, 3, 3, \cdots, 3, 1)$ and $s = 8((1 + \lceil\beta\rceil)^2 + \log_2(M\lceil\beta\rceil^{\lfloor\beta\rfloor}))$, such that $F_r(x) = U_{F_r}(x)$ and $H_r(x) = U_{H_r}(x)$ for all $x \in [a_1, a_R]$.*

Proof. The functions F_r and H_r, $r = 2, \cdots, R$, are piecewise linear functions, consisting of four pieces each. This means that these function can be perfectly represented as a linear combination of three ReLU functions. The interpolation points provide the values of the shift vectors. Writing this out for F_r gives

$$F_r(x) = \frac{\sigma(x - a_{r-1})}{b_{r-1} - a_{r-1}} + \left(\frac{1}{b_{r-1} - a_{r-1}} + \frac{1}{a_r - b_{r-1}}\right)\sigma(x - a_{r-1}) + \frac{\sigma(x - a_{r-1})}{a_r - b_{r-1}}.$$

For H_r, $r = 2, \cdots, R$ this can be done in a similar way. For H_1 and H_R we actually only need one ReLU function. The networks weights in this construction are greater than one. The difference between two consecutive points a_r and b_r can be lower bounded by using that for $x, y \geq 0$: $(x+y)^{\lceil\beta\rceil} - x^{\lceil\beta\rceil} \geq y^{\lceil\beta\rceil}$. Because of

$$\frac{(2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil})^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M} - \frac{(2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil} - \frac{1}{4})^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M} \geq \frac{(\frac{1}{4})^{\lceil\beta\rceil}}{2^{\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M},$$

we can upper bound all the network weights by

$$2^{1+2\lceil\beta\rceil+\lceil\beta\rceil^2}\lceil\beta\rceil^{\lfloor\beta\rfloor}M, \tag{2.B.3}$$

which is the inverse of the lower bound on the smallest difference between two consecutive points multiplied by two. Dividing the multiplicative constants by this bound and combining (2.A.2) the resulting network with the $\mathrm{Scale}_C(x)$ network from Proposition 2.A.2 with C equal to (2.B.3) yields a network with the required output and parameters bounded by one. The network class is simplified by using the depth-synchronization (2.A.3) followed by the enlarging property of neural networks (2.A.1). □

The previous partition yields an approximation $T^\beta : [a_1, a_R] \to \mathbb{R}$ of the logarithm on the entire interval $[a_1, a_R]$ via

$$T^\beta(x) := \sum_{r=2}^{R} F_r(x) T_{a_r}^{\lfloor\beta\rfloor}(x) + \sum_{r=1}^{R} H_r(x) T_{b_r}^{\lfloor\beta\rfloor}(x). \tag{2.B.4}$$

This function depends on M through the sequence of points a_r and b_r.

We can now derive the same type of error bound as in Lemma 2.B.2 for all $x \in [0,1]$. For this, define the projection $\pi : [0,1] \to [a_1, a_R]$, that maps $x \in [0,1]$ to itself, if it is already in the interval $[a_1, a_R]$, and to the closest boundary point otherwise.

Lemma 2.B.4. *For all $x \in [0,1]$, we have $|e^{T^\beta(\pi(x))} - x| \leq M^{-1}$.*

Proof. First consider $x \in (a_1, a_R]$. By construction there exists a unique $r^* \in \{2, 3, \cdots, R\}$ and a unique $\bar{r} \in \{1, \cdots, R\}$ such that $x \in (a_{r^*-1}, a_{r^*}]$, and $x \in (b_{\bar{r}-1}, b_{\bar{r}}]$. By the mean value theorem and (2.B.4),

$$\begin{aligned}
\left| e^{T^\beta(x)} - x \right| &\leq e^\xi \left| T^\beta(x) - \log(x) \right| \\
&= e^\xi \left| \sum_{r=2}^{R} F_r(x) T_{a_r}^{\lfloor\beta\rfloor}(x) + \sum_{r=1}^{R} H_r(x) T_{b_r}^{\lfloor\beta\rfloor}(x) - \log(x)(F_{r^*}(x) + H_{\bar{r}}(x)) \right| \\
&\leq F_{r^*}(x) e^\xi \left| T_{a_{r^*}}^{\lfloor\beta\rfloor}(x) - \log(x) \right| + H_{\bar{r}}(x) e^\xi \left| T_{b_{\bar{r}}}^{\lfloor\beta\rfloor}(x) - \log(x) \right|,
\end{aligned}$$

where ξ is some number between $T^\beta(x)$ and $\log(x)$. We now want to apply Proposition 2.B.2. For this we need to find a $\lambda \geq 1$ such that $[a_{r^*-1}, a_{r^*}] \cup [b_{\bar{r}-1}, b_{\bar{r}}] \in \mathcal{D}_\lambda$ and

$\xi \leq \max_{y \in \mathcal{D}_\lambda} \log(y)$, with $\mathcal{D}_\lambda$ as defined by that proposition. Because of our choice of the sequences of points a_r and b_r,

$$\lambda := \max\left\{\frac{r^*}{2} + 2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil} - \frac{3}{4}, \frac{\bar{r}}{2} + 2^{\lceil\beta\rceil}\lceil\beta\rceil^{\lfloor\beta\rfloor/\lceil\beta\rceil} - \frac{1}{2}\right\} - 1$$

satisfies $\lambda \geq 1$, since $r^* \geq 2$ and $\bar{r} \geq 1$. Furthermore this choice of λ guarantees that $[a_{r^*-1}, a_{r^*}] \cup [b_{\bar{r}-1}, b_{\bar{r}}] \subseteq \mathcal{D}_\lambda$. For the bound on ξ, notice that $x \in [a_{r^*-1}, a_{r^*}] \cup [b_{\bar{r}-1}, b_{\bar{r}}]$ and that $T^\beta(x) = F_{r^*}(x)T_{a_{r^*}}^{\lfloor\beta\rfloor}(x) + H_{\bar{r}}(x)T_{b_{\bar{r}}}^{\lfloor\beta\rfloor}(x)$. Combined with the second statement of Proposition 2.B.1, that is $T_c^\kappa \leq \log(c)$ for $0 < c \leq x$, and together with $F_{r^*}(x) + H_{\bar{r}}(x) = 1$, this yields $\xi \leq \max\{\log(a_{r^*}), \log(b_{\bar{r}})\}$. Thus we can apply Proposition 2.B.2 and obtain

$$F_{r^*}(x)e^\xi \left|T_{a_{r^*}}^{\lfloor\beta\rfloor}(x) - \log(x)\right| + H_{\bar{r}}(x)e^\xi \left|T_{b_{\bar{r}}}^{\lfloor\beta\rfloor}(x) - \log(x)\right| \leq F_{r^*}(x)\frac{1}{M} + H_{\bar{r}}(x)\frac{1}{M} = \frac{1}{M},$$

completing the proof for $x \in [a_1, a_R]$.

When $x \in [0, a_1]$, notice that $0 < a_1 < M^{-1}$ and $T^\beta(\pi(x)) = T_{b_1}^{\lfloor\beta\rfloor}(a_1)$. Hence by Proposition 2.B.1 together with $b_1 = M^{-1}$, we get that $T^\beta(\pi(x)) \leq \log(M^{-1})$ proving that both x and $e^{T^\beta(\pi(x))}$ are in $[0, M^{-1}]$. Thus the conclusion also holds for $x \in [0, a_1]$.

For $a_R \geq 1$, the proof follows from $[0,1] \subseteq ([0, a_1] \cup [a_1, a_R])$. Thus it remains to study $a_R < 1$. Consider $x \in [a_R, 1]$. Using that $1 - M^{-1} \leq a_R < 1$ and that $T^\beta(\pi(x)) = T_{b_R}^{\lfloor\beta\rfloor}(a_R) = T_{a_R}^{\lfloor\beta\rfloor}(a_R)$ yields $T^\beta(\pi(x)) = \log(a_R)$. This gives us that both x and $e^{T^\beta(\pi(x))}$ are in $[a_R, 1] \subset [1 - M^{-1}, 1]$, which immediately yields the required bound. □

Network Construction

The following result shows how to approximate multiplications with deep ReLU networks. This is required later to construct neural networks mimicking the Taylor-approximation T^β considered in the previous section.

Lemma 2.B.5 (Lemma A.3. of [127]). *For every $\eta \in \mathbb{N}_{\geq 1}$ and $D \in \mathbb{N}_{\geq 1}$, there exists a network $Mult_\eta^D \in \mathcal{F}_{\mathrm{id}}((\eta + 5)\lceil\log_2(D)\rceil, (D, 6D, 6D, \cdots, 6D, 1))$, such that $Mult_\eta^D \in [0,1]$ and*

$$\left|Mult_\eta^D(x_1, \cdots, x_D) - \prod_{i=1}^{D} x_i\right| \leq 3^D 2^{-\eta}, \quad \textit{for all } (x_1, \cdots, x_D) \in [0,1]^D.$$

Moreover $\mathrm{Mult}_\eta^D(x)=0$ *if one of the coefficients of* $\mathbf{x}$ *is zero.*

Remark 2.B.6. Using (2.A.5) the number of parameters in the neural network Mult_η^D is bounded by $((\eta+5)\lceil\log_2(D)\rceil+1)42D^2 \leq (\eta+5)126D^2\log_2(D)$.

We now have all the required ingredients to finish the proof of Theorem 2.4.1:

Proof of Theorem 2.4.1. Since $a_1=\sigma(0\cdot x+a_1)$, the projection π can be written in terms of ReLU functions as

$$\pi(x)=\max\big(a_1,\min(x,a_R)\big)=\sigma(0\cdot x+a_1)+\sigma(x-a_1)-\sigma(x-a_R).$$

For $a_R\leq 1$, all network parameters are bounded by one and this defines a neural network in $\mathcal{F}_{\mathrm{id}}(1,(1,3,1),8)$. When $a_R>1$, we replace $\sigma(x-a_R)$ with $\sigma(x-1)$ as we are only interested in input in the interval $[0,1]$. Having thus obtained a value in the interval $[a_1,a_R]$, we can, for any $r\in\{1,\cdots,R\}$, apply the network U_{F_r} from Proposition 2.B.3 to it. Using depth synchronization (2.A.3) and parallelization (2.A.4), we can combine the network U_{F_r} with a parallel network that forwards the input value to obtain a network in the network class $\mathcal{F}_{\mathrm{id}}(L,\mathbf{m},s)$, with $L=4\big((1+\lceil\beta\rceil)^2+\log_2(M\lceil\beta\rceil^{\lfloor\beta\rfloor})\big)$, $\mathbf{m}=(1,3,1,4,\cdots,4,2)$ and $s=13\big((1+\lceil\beta\rceil)^2+\log_2(M\lceil\beta\rceil^{\lfloor\beta\rfloor})\big)$, that maps $x\in[0,1]$ to $(F_r(\pi(x)),\pi(x))$. The next step is to construct a network that approximates $F_r(x)T^\beta_{a_r}(x)$. Since $a_r\in[M^{-1},1+M^{-1}]$, (2.B.2) allows us, for $\gamma=1,\cdots,\lfloor\beta\rfloor$, to use the network $\mathrm{Mult}_\eta^{\gamma+1}$ with input vector $(F_r(\pi(x)),\pi(x),\cdots,\pi(x))$ to compute approximately the function $F_r(\pi(x))\pi(x)^\gamma$, and multiply its output with $c_\gamma/\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1}M^{\lceil\beta\rceil}$. For each $\gamma\in\{1,\cdots,\lfloor\beta\rfloor\}$ we have a network that approximately computes the function $x\mapsto F_r(\pi(x))\pi(x)^\gamma c_\gamma/\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1}M^{\lceil\beta\rceil}$. We now consider the network that computes these functions in parallel and combines this with a single shallow hidden node network to approximately compute $F_r(\pi(x))c_0/\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1}M^{\lceil\beta\rceil}$. Making use of parallelization (2.A.4), depth synchronization (2.A.3) and Remark 2.B.6, this yields a network $G_{F_r}\in\mathcal{F}_{\mathrm{id}}(L^*,(1,6(\lceil\beta\rceil)^2,\cdots,6(\lceil\beta\rceil)^2,1),s^*)$, with

$$\begin{aligned}L^*&=4((1+\lceil\beta\rceil)^2+\log_2(M\lceil\beta\rceil^{\lfloor\beta\rfloor}))+2(\eta+5)\log_2(\lceil\beta\rceil)\\ s^*&=13((1+\lceil\beta\rceil)^2+\log_2(M\lceil\beta\rceil^{\lfloor\beta\rfloor}))+(\eta+5)\log_2(\lceil\beta\rceil)126(\lceil\beta\rceil)^3\end{aligned}$$

such that

$$\left|G_{F_r}(x)-F_r(\pi(x))\sum_{\gamma=0}^{\lfloor\beta\rfloor}\frac{c_\gamma}{\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1}M^{\lceil\beta\rceil}}\pi(x)^\gamma\right|\leq 3^{\lceil\beta\rceil}2^{-\eta}.$$

Due to the normalization constant $\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1}M^{\lceil\beta\rceil}$ it holds that $G_{F_r}(x)\in[-1,1]$ when $\pi(x)$ is in the support of F_r. If $\pi(x)$ is outside the support of F_r, then Lemma 2.B.5

guarantees that $G_{F_r}(x) = 0$. Similarly for F_r replaced by H_r, we can construct deep ReLU networks G_{H_r} with the same properties.

Using the R networks G_{H_r} and $R-1$ networks G_{F_r} in parallel together with the observation that each x can be in the support of at most one F_r and one H_r, this yields a deep ReLU network with output $\sum_{r=2}^{R} G_{F_r}(x) + \sum_{r=1}^{R} G_{H_r}(x)$, such that

$$\left|\sum_{r=2}^{R} G_{F_r}(x) + \sum_{r=1}^{R} G_{H_r}(x) - \frac{T^{\beta}(\pi(x))}{\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1} M^{\lceil\beta\rceil}}\right| \leq 3^{\lceil\beta\rceil} 2^{-\eta+1}.$$

In the next step we compose the network construction with a scaling network. For this we use the scaling network from Proposition 2.A.2 with constant $C = \lceil\beta\rceil 2^{\lfloor\beta\rfloor+1} M^{\lceil\beta\rceil}$. Since the input can be negative we use two of those networks in parallel as described in Proposition 2.A.3. This gives us a network

$$\widetilde{G} \in \mathcal{F}_{\mathrm{id}}\Big(L^* + 4\log_2\left(\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1} M^{\lceil\beta\rceil}\right), \mathbf{m}^*, 2Rs^* + 16\log_2\left(\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1} M^{\lceil\beta\rceil}\right)\Big),$$

where $\mathbf{m}^* = (1, 12R(\lceil\beta\rceil)^2, \cdots, 12R(\lceil\beta\rceil)^2, 1)$, such that

$$\left|\widetilde{G}(x) - T^{\beta}(\pi(x))\right| \leq \lceil\beta\rceil 2^{\lfloor\beta\rfloor+2} M^{\lceil\beta\rceil} 3^{\lceil\beta\rceil} 2^{-\eta}.$$

Setting $\eta = \lceil\log_2(\lceil\beta\rceil 2^{\lfloor\beta\rfloor+2} M^{\lceil\beta\rceil+1} 3^{\lceil\beta\rceil})\rceil$, this is upper bounded by M^{-1}. Applying the triangle inequality, the mean value theorem and Lemma 2.B.4 yields

$$\left|e^{\widetilde{G}(x)} - x\right| \leq \left|e^{\widetilde{G}(x)} - e^{T^{\beta}\log(\pi(x))}\right| + \left|e^{T^{\beta}\log(\pi(x))} - x\right| \leq \frac{e^{2/M}}{M} + \frac{1}{M} \leq \frac{4}{M}, \quad (2.B.5)$$

where the term $e^{2/M}$ comes from noticing that $|\widetilde{G}(x) - T^{\beta}\log(\pi(x))| \leq M^{-1}$, $|T^{\beta}\log(\pi(x)) - \log(1)| \leq M^{-1}$ and triangle inequality.

To derive the lower bound $G(x) \geq \log(4/M)$, we construct a network that computes the maximum between $\widetilde{G}(x)$ and $\log(4/M)$. Since $M \geq 1$ implies $|\log(4/M)|/\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1} M^{\lceil\beta\rceil} \leq 1$, we can achieve this by adding one additional layer before the scaling. This layer can be written as

$$\sigma\Big(x - \frac{\log(4/M)}{\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1} M^{\lceil\beta\rceil}}\Big) + \frac{\log(4/M)}{\lceil\beta\rceil 2^{\lfloor\beta\rfloor+1} M^{\lceil\beta\rceil}}\sigma(1). \quad (2.B.6)$$

Applying the scaling as before yields a network $G(x) = \max\{\widetilde{G}(x), \log(4/M)\}$ that is in the same network class as $\widetilde{G}(x)$. For the upper bound notice that if $G(x) = \widetilde{G}(x)$,

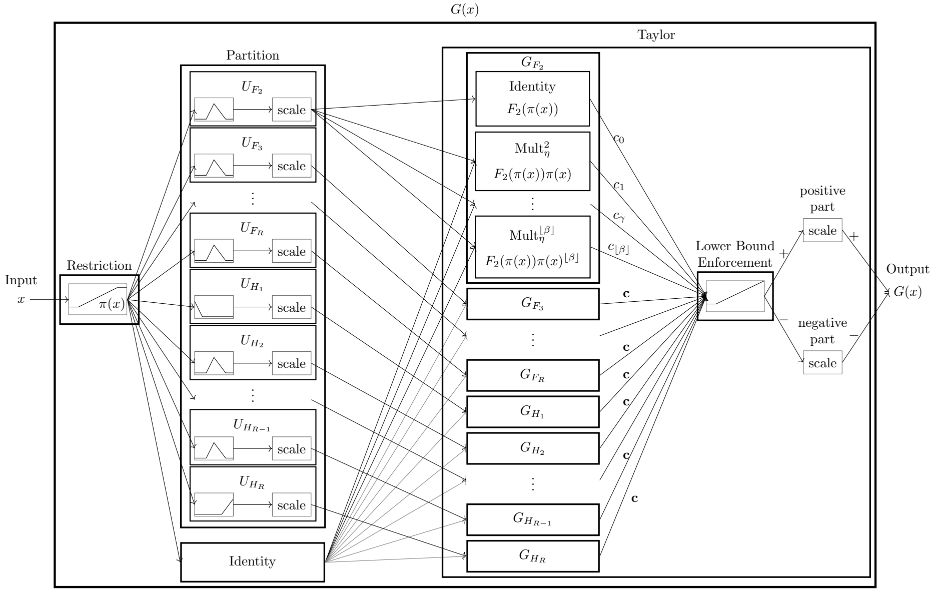


Figure 2.B.2: The construction of the logarithm approximation network G of Theorem 2.4.1 from subnetworks. The difference between the networks G and $\widetilde{G}$ is the single layer which enforces the lower bound, which is not present in the network $\widetilde{G}$.

then the bound follows from (2.B.5). When $G(x) = \log(4/M)$, then $\widetilde{G}(x) \leq \log(4/M)$, so (2.B.5) implies that $x \leq 8/M$. Hence

$$\left|e^{G(x)} - x\right| = \left|\frac{4}{M} - x\right| \leq \frac{4}{M}.$$

The network size as given in the theorem is an upper bound on the network size obtained here, which is allowed by the depth-synchronization followed by the enlarging property, and is done in order to simplify the expressions. □

Figure 2.B.2 shows the main substructures of the deep ReLU network construction in this proof.

2.C Further technicalities

Proposition 2.C.1 (Bernstein's inequality)**.** *For independent random variables* $(Z_i)_{i=1}^n$ *with zero mean and moment bounds* $\mathbb{E}|Z_i|^m \leq \frac{1}{2}m!U^{m-2}v_i$ *for* $m = 2, 3, \ldots$ *and* $i = 1, \ldots, n$ *for some constants* U *and* v_i*, we have*

$$\mathbb{P}\left(\left|\sum_{i=1}^n Z_i\right| > x\right) \leq 2e^{-\frac{x^2}{2v+2Ux}}, \quad \textit{for } v \geq \sum_{i=1}^n v_i.$$

This formulation of Bernstein's inequality is based on the formulation in Lemma 2.2.11 of [148]. The proof can be found in [16].

The next elementary inequality generalizes Lemma 10 of [127].

Lemma 2.C.2. *If* a, b, c, d *are real numbers,* $a \geq 0$*, such that* $|a - b| \leq 2\sqrt{a}c + d$*, then, for each* $\epsilon \in (0, 1]$*,*

$$(1-\epsilon)(b-d) - \frac{(1-\epsilon)^2}{\epsilon}c^2 \leq a \leq (1+\epsilon)(b+d) + \frac{(1+\epsilon)^2}{\epsilon}c^2.$$

Proof. First notice that $|a-b| \leq 2\sqrt{a}c+d$ if and only if $-2\sqrt{a}c-d \leq a-b \leq 2\sqrt{a}c+d$. Using that $2xy \leq x^2 + y^2$ for all $x, y \in \mathbb{R}$, we get for $x := \sqrt{a}\sqrt{\epsilon}/\sqrt{1+\epsilon}$ and $y := c\sqrt{1+\epsilon}/\sqrt{\epsilon}$, that

$$2\sqrt{a}c = 2xy \leq x^2 + y^2 = \frac{\epsilon a}{1+\epsilon} + \frac{(1+\epsilon)c^2}{\epsilon}$$

and therefore

$$a - b \leq \frac{\epsilon a}{1+\epsilon} + \frac{(1+\epsilon)c^2}{\epsilon} + d.$$

Rearranging the terms yields the upper bound of the lemma. For the lower bound notice that if $\epsilon = 1$, then the lower bound is zero, and holds since $a \geq 0$. For $\epsilon \in (0, 1)$ using the same argument but now with $x = \sqrt{a}\sqrt{\epsilon}/\sqrt{1-\epsilon}$ and $y = c\sqrt{1-\epsilon}/\sqrt{\epsilon}$, gives

$$a - b \geq -\frac{\epsilon a}{1-\epsilon} - \frac{(1-\epsilon)c^2}{\epsilon} - d.$$

Rearranging the terms yields the lower bound of the proposition. □

The number a is required to be nonnegative as otherwise $\sqrt{a}$ would not be a real number. In the statement in [127] the constants a, b, c, d are all required to be positive. However since the inequality $2xy \leq x^2 + y^2$ holds for all real numbers x, y the positivity constraint is not necessary. However, when c and d are negative the term $2\sqrt{a}c + d$ is negative, and no pair a, b exists such that the condition is satisfied.

Recall that $d_\tau(\mathbf{f},\mathbf{g}) := \sup_{\mathbf{x}\in\mathcal{D}} \max_{k=1,\cdots,K} |(\tau \vee f_k(\mathbf{x})) - (\tau \vee g_k(\mathbf{x}))|$. Observe that $d_\tau(\mathbf{f},\mathbf{g}) = 0$ does not imply $\mathbf{f} = \mathbf{g}$, which is why d_τ is not a metric. The next lemma shows that this, however, defines a pseudometric.

Lemma 2.C.3. *Let $\mathbf{f},\mathbf{g},\mathbf{h} : \mathcal{D} \to \mathbb{R}^K$, then for every $\tau \in \mathbb{R}$:*

(i) $d_\tau(\mathbf{f},\mathbf{g}) \geq 0$

(ii) $d_\tau(\mathbf{f},\mathbf{f}) = 0$

(iii) $d_\tau(\mathbf{f},\mathbf{g}) = d_\tau(\mathbf{g},\mathbf{f})$

(iv) $d_\tau(\mathbf{f},\mathbf{g}) \leq d_\tau(\mathbf{f},\mathbf{h}) + d_\tau(\mathbf{h},\mathbf{g})$.

Proof. (i), (ii) and (iii) follow immediately. (iv) follows from applying triangle inequality to the $\|\cdot\|_\infty$ norm,

$$\begin{aligned} d_\tau(\mathbf{f},\mathbf{g}) &= \Big\| \max_{k=1,\cdots,K} |(\tau \vee f_k(\cdot)) - (\tau \vee g_k(\cdot))| \Big\|_\infty \\ &\leq \Big\| \max_{k=1,\cdots,K} |(\tau \vee f_k(\cdot)) - (\tau \vee h_k(\cdot))| \Big\|_\infty \\ &+ \Big\| \max_{k=1,\cdots,K} |(\tau \vee h_k(\cdot)) - (\tau \vee g_k(\cdot))| \Big\|_\infty \\ &= d_\tau(\mathbf{f},\mathbf{h}) + d_\tau(\mathbf{h},\mathbf{g}). \end{aligned}$$

□

Lemma 2.C.4. *If $\mathcal{G}$ is a function class of functions from $\mathcal{D}$ to $[0,\infty)^K$, then for all $\delta > 0$ and $\tau > 0$*

$$\mathcal{N}\big(\delta, \log(\mathcal{G}), d_{\log(\tau)}(\cdot,\cdot)\big) \leq \mathcal{N}\big(\delta\tau, \mathcal{G}, d_\tau(\cdot,\cdot)\big).$$

Proof. Let $\delta > 0$. Denote by $(\mathbf{g}_j)_{j=1}^{\mathcal{N}_n}$ the centers of a minimal internal $\delta\tau$-covering of $\mathcal{G}$ with respect to d_τ and let $\mathbf{g} \in \mathcal{G}$. By the cover property, there exist a $j \in \{1,\cdots,\mathcal{N}_n\}$ such that $d_\tau(\mathbf{g},\mathbf{g}_j) \leq \delta\tau$.

The derivative of $\log(u)$ is $1/u$, so the logarithm is Lipschitz on $[\tau,\infty)$ with Lipschitz constant τ^{-1}. Applying this to $d_{\log(\tau)}(\log(\mathbf{g}),\log(\mathbf{g}_j))$, noticing that $\max\{\log(\tau),\log(x)\} \in [\log(\tau),\infty)$ for $x \in [0,\infty)$, yields

$$\begin{aligned} &\max_{\mathbf{x}\in\mathcal{D}} \max_{k=1,\cdots,K} |(\log(\tau) \vee \log(g_k(\mathbf{x}))) - (\log(\tau) \vee \log(g_{j,k}(\mathbf{x})))| \\ &\leq \tau^{-1} \max_{\mathbf{x}\in\mathcal{D}} \max_{k=1,\cdots,K} |(\tau \vee g_k(\mathbf{x})) - (\tau \vee g_{j,k})(\mathbf{x}))| \\ &\leq \tau^{-1}\delta\tau = \delta. \end{aligned}$$

Since $\mathbf{g} \in \mathcal{G}$ was arbitrary, this means that for all $\mathbf{g} \in \mathcal{G}$ there exists a $j \in \{1,\cdots,\mathcal{N}_n\}$ such that $d_{\log(\tau)}(\log(\mathbf{g}),\log(\mathbf{g}_j)) \leq \delta$. Hence $(\log(\mathbf{g}_j))_{j=1}^{\mathcal{N}_n}$ is a δ-cover for $\log(\mathcal{G})$ with respect to $d_{\log(\mathcal{T})}$. Since the $\mathbf{g}_j$ are in $\mathcal{G}$, the $\log(\mathbf{g}_j)$ are in $\log(\mathcal{G})$, thus this cover is an internal cover. Since $\mathcal{N}(\delta, \log(\mathcal{G}), d_{\log(\mathcal{T})}(\cdot,\cdot))$ is the minimal number of balls with center in $\log(\mathcal{G})$ required to cover $\log(\mathcal{G})$. This proves the assertion. □

Proof of Lemma 2.3.7. Let $\mathbf{p}, \mathbf{q} \in \mathcal{S}^k$. Thus, $\sum_{k=1}^K p_k = 1$, $\sum_{k=1}^K q_k = 1$ and

$$\sum_{k=1}^K p_k \left(B \wedge \log\left(\frac{p_k}{q_k}\right)\right) = \sum_{k=1}^K \left(p_k \left(B \wedge \log\left(\frac{p_k}{q_k}\right)\right) - p_k + q_k\right). \tag{2.C.1}$$

Suppose for the moment that for any $k = 1, \cdots, K$,

$$p_k \left(B \wedge \log\left(\frac{p_k}{q_k}\right)\right) - p_k + q_k \geq \frac{1}{C_{m,B}} p_k \left|B \wedge \log\left(\frac{p_k}{q_k}\right)\right|^m, \tag{2.C.2}$$

with $C_{m,B} := \max\{m!, B^m/(B-1)\}$. Applying this inequality to each term on the right hand side of (2.C.1) gives

$$\sum_{k=1}^K p_k \left(B \wedge \log\left(\frac{p_k}{q_k}\right)\right) \geq \sum_{k=1}^K \frac{1}{C_{m,B}} p_k \left|B \wedge \log\left(\frac{p_k}{q_k}\right)\right|^m.$$

Since $C_{m,B} > 0$, multiplying both sides of the inequality with $C_{m,B}$ yields the claim.

It remains to proof (2.C.2). First we consider the case that $p_k = 0$. By considering the limit we get that $0 \log^m(0) = 0$, for $m = 1, 2, \cdots$. Thus the right hand side of (2.C.2) is equal to 0, while the left hand side is equal to q_k. Since $q_k \geq 0$, this proves (2.C.2) for this case.

Assume now that $p_k > 0$. Dividing both sides by p_k yields

$$B \wedge \log\left(\frac{p_k}{q_k}\right) - 1 + \frac{q_k}{p_k} \geq \frac{1}{C_{m,B}} \left|B \wedge \log\left(\frac{p_k}{q_k}\right)\right|^m.$$

If $p_k/q_k \geq e^B$ the inequality follows immediately. It remains to study the case that $p_k/q_k < e^B$. In this case one can always replace $B \wedge \log(p_k/q_k)$ by $\log(p_k/q_k)$. Introducing the new variable $u = q_k/p_k$ and replacing $C_{m,B}$ by $C > 0$ gives rise to a function

$$H_{C,m}(u) = u - 1 - \log(u) - |\log(u)|^m/C.$$

It remains to show that $H_{C_{m,B},m}(u) \geq 0$ for all $u \geq e^{-B}$. Obviously, $H_{C,m}(1) = 0$ for all C, so we only have to consider $u \neq 1$. Consider first $u > 1$ and $C = m!$. Using the substitution $u = e^s$ gives

$$m!e^s - m!(s+1) - s^m.$$

Substituting the power series for the exponential function leads to

$$m!\sum_{n=0}^{\infty}\frac{s^n}{n!}-m!(1+s)-s^m=m!\sum_{n=2}^{m-1}\frac{s^n}{n!}+m!\sum_{n=m+1}^{\infty}\frac{s^n}{n!}>0,$$

where the last strict inequality holds because $u>1$ and thus $s>0$. Thus $H_{m!,m}(u)\geq 0$ for $u>1$.

For $u\in(e^{-b},1)$, dividing by $u-\log(u)-1$ gives us the following constraint on the constant C :

$$C\geq \sup_{u\in(e^{-B},1)}\frac{|\log(u)|^m}{u-\log(u)-1}. \tag{2.C.3}$$

This division can be done since $u-\log(u)-1>0$ when $u>0$, $u\neq 1$ and zero if and only if $u=1$, which for example can be shown by observing the sign of the derivative.

Define $C_{<1}$ as $C_{<1}:=B^m/(B-1)$. Since $|\log(u)|^m/(u-\log(u)-1)$ is strictly decreasing on $(0,1)$, see Proposition 2.C.5 (II), it follows for $u\in[e^{-B},1)$ that $|\log(u)|^m/(u-\log(u)-1)\leq B^m/(e^{-B}+B-1)$. Now since $B>1$, it follows that $B^m/(u+B-1)$ is also strictly decreasing on $[0,1]$. Hence on $[0,e^{-B}]$ we have $B^m/(e^{-B}+B-1)\leq B^m/(u+B-1)\leq C_{<1}$, thus $C_{<1}$ satisfies (2.C.3).

Now notice that $C_{m,B}=\max\{C_{<1},m!\}$. Consequently $H_{C_{m,B},m}(u)\geq 0$, for all $u\geq e^{-B}$, proving (2.C.2). □

For all $m=2,3,\ldots$ define the function $F_m:(0,\infty)\to[0,\infty)$ as

$$F_m(u):=\frac{|\log^m(u)|}{u-\log(u)-1}.$$

Since $u-\log(u)-1\geq 0$, this function indeed takes only positive values. Furthermore since $u-\log(u)-1=0$ only when $u=1$ this is the only possible singularity/discontinuity of this function. The next result derives some properties of the function $F_m(u)$.

Proposition 2.C.5. *If* $m=2,3,\cdots$, *then*
(i) $\lim_{u\to 1}F_2(u)=2$ *and* $\lim_{u\to 1}F_m(u)=0$ *for* $m>2$
(ii) $F_m(u)$ *is strictly decreasing on* $(0,1)$.

Proof. (i): For $u=1$, it holds that $(u-\log(u)-1)=0$ and $|\log^m(u)|=0$. Applying L'Hopital's rule twice yields the desired result.

(ii): The L'Hopital's like rule for monotonicity, see [112] or Lemma 2.2 in [4], states that a function f/g on an interval (a,b), satisfying $g'\neq 0$ and either $f(a)=0=g(a)$ or $f(b)=0=g(b)$, is strictly increasing/decreasing if f'/g' is strictly increasing/decreasing on (a, b). For $f(u) = |\log^m(u)|$ and $g(u) = u - \log(u) - 1$, we have

$$\frac{f'(u)}{g'(u)} = \frac{m \log(u)|\log^{m-2}(u)|}{u - 1}$$

and for $\bar{f}(u) = m \log(u)|\log^{m-2}(u)|$ and $\bar{g}(u) = u - 1$, we obtain

$$\frac{\bar{f}'(u)}{\bar{g}'(u)} = \frac{(m-1)m|\log^{m-2}(u)|}{u}.$$

On $u \in (0, 1)$, $\bar{f}'(u)/\bar{g}'(u)$ is strictly decreasing. Applying the L'Hopital's like rule for monotonicity twice yields the statement. □

Proof of Lemma 2.3.4. The inequality $\mathrm{KL}_2(P, Q) \leq \mathrm{KL}_B(P, Q)$ follows direct from the definition of the truncated Kullback-Leibler divergence. Write $P = P^a + P^s$ for the Lebesgue decomposition of P with respect to Q such that $P^a \ll Q$. The Lebesgue decomposition ensures existence of a set A with $P^a(A) = 0 = P^s(A^c)$. For $x \in A$, we define $dP/dQ(x) := +\infty$. For the dominating measure $\mu = (P + Q)/2$, denote by p, p^a, p^s, q the μ-densities of P, P^a, P^s, Q, respectively. Since $p^s q = 0$,

$$\begin{aligned} H^2(P, Q) &= \int \big(p^a + p^s - \sqrt{p^a q}\big) \\ &\leq \int_{0<p^a/q\leq e^2} \big(p^a - \sqrt{p^a q}\big) + \int_{p^a/q>e^2} p^a + \int p^s. \end{aligned}$$

For every $u \in \mathbb{R}$, we have $1 - u \leq e^{-u}$ and hence $e^u - 1 \leq ue^u$. Substituting $u = \log(\sqrt{y})$ yields $\sqrt{y} - 1 \leq \sqrt{y}\log(\sqrt{y})$ and therefore $y - \sqrt{y} \leq y\log(\sqrt{y}) = y\log(y)/2$. With $y = p^a/q$, we find,

$$H^2(P, Q) \leq \int_{0<p^a/q\leq e^2} \frac{p^a}{2q} \log\Big(\frac{p^a}{q}\Big) q + \int_{p^a/q>e^2} p^a + \int p^s.$$

The other direction works similarly. Second order Taylor expansion around one gives for $y > 0$, $y\log(y) \leq y - 1 + \frac{1}{2}(y-1)^2/(y \wedge 1)$. For $y = \sqrt{x}$, we find $x\log(x) = 2\sqrt{x} \cdot \sqrt{x}\log(\sqrt{x}) \leq 2(x - \sqrt{x}) + (1 \vee \sqrt{x})(\sqrt{x} - 1)^2$. Consequently, for each $B \geq 0$,

$$\begin{aligned} \mathrm{KL}_B(P, Q) &= \int_{p^a/q\leq e^B} \frac{p^a}{q} \log\Big(\frac{p^a}{q}\Big) q + B \int_{dP/dQ>e^B} dP \\ &\leq 2e^{B/2} H^2(P, Q) + 2\int_{p^a/q\leq e^B} p - \sqrt{pq} + B\int_{dP/dQ>e^B} dP. \end{aligned}$$

If $\int_{p^a/q \leq e^B} p^a - \sqrt{p^a q} \leq 0$, we can use that $H^2(P,Q) \geq \frac{1}{2}\int_{p/q \geq e^B}(\sqrt{p}-\sqrt{q})^2 \geq \frac{1}{2}\int_{p/q \geq e^B} p(1-e^{-B/2})^2$ and hence

$$\mathrm{KL}_B(P,Q) \leq 2\Big(e^{B/2} + (1-e^{-B/2})^{-2}\Big)H^2(P,Q).$$

Otherwise, if $\int_{p^a/q \leq e^B} p^a - \sqrt{p^a q} > 0$, we can upper bound

$$\begin{aligned}
\mathrm{KL}_B(P,Q) &\leq 2e^{B/2}H^2(P,Q) + B(1-e^{-B/2})^{-1}\int_{p^a/q \leq e^B} p - \sqrt{pq} \\
&\quad + B\int_{dP/dQ > e^B} dP \\
&\leq 2e^{B/2}H^2(P,Q) + B(1-e^{-B/2})^{-1}\int p - \sqrt{pq} \\
&= \Big(2e^{B/2} + B(1-e^{-B/2})^{-1}\Big)H^2(P,Q).
\end{aligned}$$

The result now follows by observing that since $B \geq 2$, both $B(1-e^{-B/2})^{-1}$ and $2(1-e^{-B/2})^{-2}$ are less than $2e^{B/2}$. □

Proposition 2.C.6. *Recall that Φ denotes the softmax function. The function $\log(\mathbf{\Phi}(\cdot)) : \mathbb{R}^K \to \mathbb{R}^K$ satisfies $|\log(\mathbf{\Phi}(\mathbf{x})) - \log(\mathbf{\Phi}(\mathbf{y}))|_\infty \leq K\|\mathbf{x}-\mathbf{y}\|_\infty$.*

Proof. Consider the composition of the logarithm with the softmax function, that is,

$$\left(\log\left(\frac{e^{x_1}}{\sum_{j=1}^K e^{x_j}}\right), \cdots, \log\left(\frac{e^{x_K}}{\sum_{j=1}^K e^{x_j}}\right)\right).$$

It holds for $k, i \in \{1, \cdots, K\}$, $i \neq k$ that

$$\begin{aligned}
\frac{\partial}{\partial x_k}\log\left(\frac{e^{x_k}}{\sum_{j=1}^K e^{x_j}}\right) &= 1 - \frac{e^{x_k}}{\sum_{j=1}^K e^{x_j}}, \\
\frac{\partial}{\partial x_k}\log\left(\frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}\right) &= -\frac{e^{x_k}}{\sum_{j=1}^K e^{x_j}}.
\end{aligned}$$

The partial derivatives are bounded in absolute value by one. The combined log-softmax function is therefore Lipschitz continuous (w.r.t to $\|\cdot\|_\infty$ norm for vectors) with Lipschitz constant bounded by K. □

Proof of Lemma 2.3.8. We start proving the first bound. Notice that $g \in \log(\mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s))$ means that there exists a ReLU network $f_g \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$ such that $g(\mathbf{x}) = \log(\mathbf{\Phi}(f_g(\mathbf{x})))$. By Lemma 5 of [127] it holds that $\mathcal{N}(\delta/(2K), \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s), \|\cdot\|_\infty) \leq (4\delta^{-1}K(L+1)V^2)$. Let $\delta > 0$. Denote by $(\mathbf{f}_j)_{j=1}^{\mathcal{N}_n}$ the centers of a minimal $\delta/(2K)$-covering of $\mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$ with respect to $\|\cdot\|_\infty$. Triangle inequality gives that for each $\mathbf{f}_j$ there exists a $\widehat{\mathbf{f}_j} \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$ such that $(\widehat{\mathbf{f}_j})_{j=1}^{\mathcal{N}_n}$ is an interior δ/K-cover of $\mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$. Let $\mathbf{g} \in \log(\mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s))$. By the cover property, there exists a $j \in \{1, \cdots, \mathcal{N}_n\}$ such that $\|f_g - \widehat{f_j}\| \leq \delta/K$. Proposition 2.C.6 yields:

$$\|\mathbf{g} - \log(\Phi(\widehat{\mathbf{f}_j}))\|_\infty = \|\log(\Phi(\mathbf{f}_g)) - \log(\Phi(\widehat{\mathbf{f}_j}))\|_\infty \leq K\|f_g - \widehat{f_j}\|_\infty \leq \delta.$$

Since $\mathbf{g} \in \log(\mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s))$ was arbitrary and $\widehat{f_j} \in \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s)$ for $j = 1, \cdots, \mathcal{N}_n$, this means that $(\log(\Phi(\widehat{\mathbf{f}_j}))$ is an internal δ-cover for $\log(\mathcal{F}_{\mathbf{\Phi}}(L, \mathbf{m}, s))$ with respect to $\|\cdot\|_\infty$. Hence

$$\begin{aligned}\mathcal{N}(\delta, \log(\mathcal{F}_\Phi(L, \mathbf{m}, s)), \|\cdot\|_\infty) &\leq \mathcal{N}(\delta/(2K), \mathcal{F}_{\mathrm{id}}(L, \mathbf{m}, s), \|\cdot\|_\infty) \\ &\leq (4\delta^{-1}K(L+1)V^2).\end{aligned}$$

Now we consider the second bound of the lemma. Using that $m_0 = d$, $m_{L+1} = K$ and by removing inactive nodes, Proposition 2.A.1, we get that $m_\ell \leq s$ for $s = 1, \cdots, L$, and thus

$$V \leq dKs^L 2^{L+2}.$$

Substituting this in the first bound and taking the logarithm yields the result. □

Proposition 2.C.7. *Consider binary classification ($K = 2$) for the conditional class probabilities $p_1(\mathbf{x}) = (3|x_1 + x_2 - 1|^8)/4$ and $p_2(\mathbf{x}) = 1 - p_1(\mathbf{x})$. If $\mathbf{X}$ is uniformly distributed on $[0,1]^2$, then*

$$\mathbb{P}_{\mathbf{X}}\left(p_1(\mathbf{X}) \leq t\right) = 2\left(\frac{4t}{3}\right)^{\frac{1}{8}} - \left(\frac{4t}{3}\right)^{\frac{1}{4}}.$$

If the distribution of $\mathbf{X}$ is given by the density $(x_1, x_2) \mapsto 3|x_1 + x_2 - 1|$, then

$$\mathbb{P}_{\mathbf{X}}\left(p_1(\mathbf{X}) \leq t\right) = 3\left(\frac{4t}{3}\right)^{\frac{1}{4}} - 2\left(\frac{4t}{3}\right)^{\frac{3}{8}}.$$

Proof. By rewriting the inequality $p_1(\mathbf{X}) \leq t$, we get for both cases that

$$\mathbb{P}_{\mathbf{X}}\left(p_1(\mathbf{X}) \leq t\right) = \mathbb{P}_{\mathbf{X}}\left((3|x_1 + x_2 - 1|^8)/4 \leq t\right)$$

$$= \mathbb{P}_{\mathbf{X}}\left(1-\left(\frac{4t}{3}\right)^{\frac{1}{8}} \leq x_1+x_2 \leq 1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}\right).$$

First we consider the case of uniform design. In this case, we find

$$\begin{aligned}
&\mathbb{P}_{\mathbf{X}}\left(1-\left(\frac{4t}{3}\right)^{\frac{1}{8}} \leq x_1+x_2 \leq 1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}\right)\\
&= \int_0^1 \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2}^{1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2} 1 dx_1 dx_2 - \int_0^{\left(\frac{4t}{3}\right)^{\frac{1}{8}}} \int_1^{1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2} 1 dx_1 dx_2\\
&- \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}}^1 \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2}^0 1 dx_1 dx_2\\
&= 2\left(\frac{4t}{3}\right)^{\frac{1}{8}} - \frac{1}{2}\left(\frac{4t}{3}\right)^{\frac{1}{4}} - \frac{1}{2}\left(\frac{4t}{3}\right)^{\frac{1}{4}}.
\end{aligned}$$

Here, the second and third double integral are correction terms that compensate for the regions where the first double integral integrates over values outside $[0,1]^2$.

To prove the second part of the statement, consider the case that the distribution of $\mathbf{X}$ is given by the density $(x_1,x_2) \mapsto 3|x_1+x_2-1|$. In this case we have that

$$\begin{aligned}
&\mathbb{P}_{\mathbf{X}}\left(1-\left(\frac{4t}{3}\right)^{\frac{1}{8}} \leq x_1+x_2 \leq 1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}\right)\\
&= \int_0^1 \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2}^{1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2} 3|x_1+x_2-1| dx_1 dx_2\\
&- \int_0^{\left(\frac{4t}{3}\right)^{\frac{1}{8}}} \int_1^{1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2} 3|x_1+x_2-1| dx_1 dx_2\\
&- \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}}^1 \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2}^0 3|x_1+x_2-1| dx_1 dx_2\\
&= \int_0^1 \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2}^{1-x_2} 3(-x_1-x_2+1) dx_1 dx_2\\
&+ \int_0^1 \int_{1-x_2}^{1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2} 3(x_1+x_2-1) dx_1 dx_2\\
&- \int_0^{\left(\frac{4t}{3}\right)^{\frac{1}{8}}} \int_1^{1+\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2} 3(x_1+x_2-1) dx_1 dx_2
\end{aligned}$$

$$
\begin{aligned}
&- \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}}^{1} \int_{1-\left(\frac{4t}{3}\right)^{\frac{1}{8}}-x_2}^{0} 3(-x_1 - x_2 + 1) dx_1 dx_2 \\
&= \frac{3}{2}\left(\frac{4t}{3}\right)^{\frac{1}{4}} + \frac{3}{2}\left(\frac{4t}{3}\right)^{\frac{1}{4}} - \left(\frac{4t}{3}\right)^{\frac{3}{8}} - \left(\frac{4t}{3}\right)^{\frac{3}{8}} .
\end{aligned}
$$

Again, the correction terms occur because we integrate over values outside $[0,1]^2$. □

Acknowledgments

We thank the two referees and the editor of the Electronic Journal of Statistics for many helpful suggestions.

Chapter 3

A supervised deep learning method for nonparametric density estimation

Abstract

Nonparametric density estimation is an unsupervised learning problem. In this chapter we propose a two-step procedure that casts the density estimation problem in the first step into a supervised regression problem. The advantage is that we can afterwards apply supervised learning methods. Compared to the standard nonparametric regression setting, the proposed procedure creates, however, dependence among the training samples. To derive statistical risk bounds, one can therefore not rely on the well-developed theory for i.i.d. data. To overcome this, we prove an oracle inequality for this specific form of data dependence. As an application, it is shown that under a compositional structure assumption on the underlying density, the proposed two-step method achieves convergence rates that are faster than the standard nonparametric rates. A simulation study illustrates the finite sample performance.

3.1 Introduction

Machine learning distinguishes between supervised and unsupervised learning tasks [21, 99]. In the supervised framework, the dataset consists of input-output pairs. No outputs are observed in the unsupervised setting. For supervised learning, classical examples are regression and classification; for unsupervised learning, commonly en-

This chapter is based on: Thijs Bos and Johannes Schmidt-Hieber (2023). A supervised deep learning method for nonparametric density estimation. Preprint arXiv:2306.10471. The research has been supported by the NWO/STAR grant 613.009.034b and the NWO Vidi grant VI.Vidi.192.021.

countered problems are density estimation and clustering. The apparent difference between supervised and unsupervised tasks results in methods that either apply to the supervised or to the unsupervised framework. Of course, neural nets can be applied in both scenarios but the underlying methodology is unrelated: In the supervised context, deep learning is applied to reconstruct the function mapping the inputs to the outputs; in the unsupervised framework, neural networks are employed for feature extraction, e.g. by making use of variational autoencoders [68].

In this chapter, we show how unsupervised multivariate density estimation can be cast into a supervised regression problem. For that, we generate suitable response variables from the data in a first step. Rewriting the problem as supervised learning task allows us to borrow strength from supervised learning methods. We demonstrate this by fitting deep ReLU networks. In the theoretical deep learning literature, it has been shown that supervised deep networks can outperform other methods if the target function exhibits some compositional structure. Making the link to supervised learning allows us to exploit this property also for density estimation. This is highly desirable as a compositional structure is frequently imposed in modelling of densities. Examples include copula models [1, 101] and Bayesian network models [78], see also Section 3.4.

Theorem 3.3.1 is the main theoretical contribution and establishes an oracle inequality for supervised regression methods applied to nonparametric density estimation. The key technical difficulty in the proof is to deal with the dependence incurred by generating the response variables in the first step of the proposed method. To control the dependence, we use a Poissonization argument. Applying the derived oracle inequality, we show in Theorem 3.3.4 that deep ReLU networks can obtain fast convergence rates, given that the underlying density has a compositional structure. For sufficiently smooth densities the convergence rates are, up to logarithmic factors in the sample size, the same as the recently obtained minimax rates in the nonparametric regression model under compositional structure on the regression function, [127]. But there are also smoothness regimes where the convergence rate is slower by a polynomial order in the sample size if compared to the nonparametric regression case. This is due to the first step in the construction of the estimator that transforms the density estimation problem into a supervised regression problem. But still then there are scenarios where the convergence rate is considerably faster than doing off-the-shelf kernel density estimation without taking the underlying compositional structure of the density into account.

The proposed two-step procedure is related to Lindsey's method which transforms parametric estimation in exponential families into a Poisson regression problem [91, 90, 45]. The first step of this method discretizes the sample space into disjoint bins. The bin counts follow a multinomial distribution that is then approximated by the Poisson distribution. Assuming Poisson distributed bin counts, maximum likelihood estimation of the parameters results then in a Poisson regression problem. A benefit of Lindsey's transformation is that the normalization constant of the exponential family vanishes. This constant is an integral over the entire domain and hard to compute in high dimensions [97, 48]. While Lindsey's method returns one observation per bin and has been formulated for exponential families, the proposed method in this chapter focuses on nonparametric densities and artificially creates a supervised dataset by computing a response vector for each of the datapoints. Approximation of the bin counts by the Poisson distribution occurs in our approach in the proof.

This chapter is structured as follows. Section 3.2 describes the construction of suitable response variables from the data. In Section 3.3 we present a suitable oracle inequality for non-i.i.d. data. Furthermore, we provide convergence rates in the case that the regression estimator is a deep neural network and the underlying density are compositional functions. In Section 3.4 we shortly discuss some density models that exhibit compositional structure. A small (exploratory) simulation is provided in Section 3.5. All proofs are deferred to the Appendix.

3.1.1 Notation

We denote vectors and vector valued functions by bold letters. For a vector $\mathbf{x} = (x_1, \ldots, x_k)^\top$ we define $|\mathbf{x}|_\infty = \max_{i=1,\ldots,k} |x_i|$, $|\mathbf{x}|_1 = \sum_{i=1}^k |x_i|$. and $|\mathbf{x}|_0 = \sum_{i=1}^k \mathbb{1}_{\{x_i \neq 0\}}$. For partial derivatives we use multi-index notation, that is, if $\boldsymbol{\alpha} \in \{0, 1, 2, \ldots\}^d$ we use the notation $\partial^{\boldsymbol{\alpha}} := \partial_{x_1}^{\alpha_1} \ldots \partial_{x_d}^{\alpha_d}$. We denote the supremum norm of a function $f : \mathcal{D} \to \mathbb{R}$ by $\|f\|_\infty = \sup_{\mathbf{x} \in \mathcal{D}} |f(\mathbf{x})|$. As commonly defined in nonparametric statistics, for a real number $x \in \mathbb{R}$, $\lfloor x \rfloor$ is the largest integer $< x$ and $\lceil x \rceil$ is the smallest integer $\geq x$. The minimum and maximum of two numbers x, y are also written as $x \wedge y$ and $x \vee y$. For two sequences $(a_n)_n$ and $(b_n)_n$, we write $a_n \lesssim b_n$ if there exists a constant C such that $a_n \leq C b_n$ for all n. Moreover, $a_n \asymp b_n$ means that $a_n \lesssim b_n$ and $b_n \lesssim a_n$. If no basis is specified, then $\log = \ln$.

3.2 Conversion into a supervised learning problem

We consider nonparametric density estimation on the hypercube $[0, 1]^d$, where we observe $2n$ i.i.d. vectors $\mathbf{X}_i \in [0, 1]^d$ which are distributed according to an unknown density f_0 from a nonparametric class. The density estimation problem is to recover this density f_0 from the data $(\mathbf{X}_i)_{i=1}^{2n}$. Here the sample size $2n$ is chosen for notational convenience, as we will do data splitting. Half of the data are used to compute an undersmoothed kernel density estimator. From that we construct response variables Y_i for the remaining data. In a last step, we fit a neural network to the resulting regression problem. The response variables Y_i can be interpreted as noisy versions of

$f_0(\mathbf{X}_i)$, such that the regression estimator then yields an estimator for the underlying density f_0. It is convenient to denote the n data points used for the kernel density estimator by $\mathbf{X}_1', \ldots, \mathbf{X}_n'$, and the n data points for the regression step by $\mathbf{X}_1, \ldots, \mathbf{X}_n$.

The multivariate kernel density estimator based on the subsample $\mathbf{X}_1', \ldots, \mathbf{X}_n'$ with $\mathbf{X}_\ell' = (X_{\ell,1}', \ldots, X_{\ell,d}')^\top$ is defined by

$$\widehat{f}_{\mathrm{KDE}}(\mathbf{x}) := \frac{1}{nh_n^d} \sum_{\ell=1}^{n} \prod_{r=1}^{d} K\left(\frac{X_{\ell,r}' - x_r}{h_n}\right), \tag{3.2.1}$$

with h_n the bandwidth and $K : \mathbb{R} \to \mathbb{R}$ the kernel. We choose a sequence h_n satisfying $(\log(n)/n)^{1/d} \leq h_n \leq 2(\log(n)/n)^{1/d}$ and such that h_n^{-1} is a positive integer for all $n > 1$. Existence of such a sequence is guaranteed by Lemma 3.3.2. The fact that h_n^{-1} is a positive integer allows us to partition $[0, 1]$ into h_n^{-1} disjoint intervals of length h_n. This construction undersmooths and does not require knowledge of the true smoothness.

For $i = 1, \ldots, n$, define

$$Y_i := \widehat{f}_{\mathrm{KDE}}(\mathbf{X}_i). \tag{3.2.2}$$

Setting $\epsilon_i := Y_i - f_0(\mathbf{X}_i)$, we obtain the regression model

$$Y_i = f_0(\mathbf{X}_i) + \epsilon_i, \quad i = 1, \ldots, n. \tag{3.2.3}$$

Although the notation seems to suggest that this is the standard nonparametric regression framework, all data points depend on the underlying kernel density estimator $\widehat{f}_{\mathrm{KDE}}$. The pairs $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ are henceforth dependent and thus not i.i.d. To deal with this dependence is the main technical challenge in the analysis of the proposed method.

The least squares estimator $\widehat{f}_n$ over a function class $\mathcal{F}$ for the density f_0 is defined as any global minimizer of the least squares loss

$$\widehat{f}_n \in \operatorname*{arg\,min}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(\mathbf{X}_i))^2.$$

Due to the nonconvex energy landscape, neural network training usually does not find the global minimum. The difference between training error of the estimator and training error of the global minimum is commonly referred to as optimization error. For any estimator $\widehat{f}$ taking values in a function class $\mathcal{F}$, and data generated from the nonparametric regression model with regression function f_0, we consider here the optimization error

$$\Delta_n(\widehat{f}, f_0) := \mathbb{E}_{f_0}\left[\frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{f}(\mathbf{X}_i))^2 - \inf_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(\mathbf{X}_i))^2\right], \tag{3.2.4}$$

where the expectation is taken over the full data set, making $\Delta_n(\widehat{f}, f_0)$ deterministic.

The risk of an estimator $\widetilde{f}$ is given by

$$R(\widetilde{f}, f_0) := \mathbb{E}_{f_0,\mathbf{X}}\left[(\widetilde{f}(\mathbf{X}) - f_0(\mathbf{X}))^2\right] = \int \mathbb{E}_{f_0}\left[(\widetilde{f}(\mathbf{x}) - f_0(\mathbf{x}))^2\right] f_0(\mathbf{x})\, d\mathbf{x}. \tag{3.2.5}$$

Here $\mathbf{X} \overset{d}{=} \mathbf{X}_1$ is independent of the data and $\mathbb{E}_{f_0,\mathbf{X}}$ is the expectation with respect to the joint distribution of $\mathbf{X}$ and the data set. We denote by $\mathbb{E}_{\mathbf{X}}$ the expectation with respect to $\mathbf{X}$.

3.3 Main results

We assume that the density f_0 belongs to the class of β-Hölder smooth function on $\mathbb{R}^d$ with support on $[0,1]^d$. For $\beta > 0$ and $\mathcal{D} \subset \mathbb{R}^d$, the ball of β-Hölder functions with radius Q is defined as

$$\begin{aligned} C_d^\beta(\mathcal{D}, Q) := \Bigg\{ f : \mathcal{D} \subseteq \mathbb{R}^d \to \mathbb{R} : &\sum_{\boldsymbol{\gamma}:|\boldsymbol{\gamma}|_1 < \beta} \|\partial^{\boldsymbol{\gamma}} f\|_\infty \\ &+ \sum_{\boldsymbol{\gamma}:|\boldsymbol{\gamma}|_1 = \lfloor \beta \rfloor} \sup_{\mathbf{x},\mathbf{y} \in \mathcal{D}, \mathbf{x} \neq \mathbf{y}} \frac{|\partial^{\boldsymbol{\gamma}} f(\mathbf{x}) - \partial^{\boldsymbol{\gamma}} f(\mathbf{y})|}{|\mathbf{x} - \mathbf{y}|_\infty^{\beta - \lfloor \beta \rfloor}} \leq Q \Bigg\}, \end{aligned} \tag{3.3.1}$$

where $\|\cdot\|_\infty$ denotes the supremum norm, $\partial^{\boldsymbol{\gamma}} = \partial_{x_1}^{\gamma_1} \ldots \partial_{x_d}^{\gamma_d}$, with $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_d) \in \{0, 1, 2, , \ldots\}^d$. The class of β-Hölder smooth densities on $\mathbb{R}^d$ and support on $[0,1]^d$ can subsequently be defined as

$$\mathcal{C}_d^\beta(Q) := \left\{ f \in C_d^\beta(\mathbb{R}^d, Q) : \operatorname{supp} f \subseteq [0,1]^d, \int_{[0,1]^d} f(\mathbf{x})\, d\mathbf{x} = 1, f \geq 0 \right\}.$$

The condition that the density f_0 is smooth on $\mathbb{R}^d$ instead of $[0,1]^d$ is imposed to avoid (technical) difficulties of the kernel density estimator near the boundary of $[0,1]^d$. (There is literature dealing with the behaviour of kernel estimators near boundaries, see for example Section 2.11 of [151].) We define the class of β-Hölder smooth densities on $[0,1]^d$ by restricting β-Hölder smooth densities on $\mathbb{R}^d$ to $[0,1]^d$,

$$\mathcal{C}_d^\beta([0,1]^d, Q) := \left\{ f : [0,1]^d \to \mathbb{R} : \text{there exists } h \in \mathcal{C}_d^\beta(Q) \text{ s.t. } f = h|_{[0,1]^d} \right\}.$$

A function $K : \mathbb{R} \to \mathbb{R}$ is said to be a (one-dimensional) kernel of order $\lfloor \beta \rfloor$ if $\int_{\mathbb{R}} K(u)\, du = 1$, $\int_{\mathbb{R}} |u|^{\lfloor \beta \rfloor + 1} K(u)\, du < \infty$, and if K has vanishing moments $\int_{\mathbb{R}} u^\ell K(u)\, du = 0$ for all $\ell = 1, \ldots, \lfloor \beta \rfloor$.

We state the oracle inequality for estimators taking values in an abstract function class $\mathcal{F}(F) \subseteq \{f : \|f\|_\infty \leq F\}$. Furthermore we denote by $\mathcal{N}_\mathcal{F}(\delta)$ the covering number of a class $\mathcal{F}(F)$ with respect to the supremum norm. More specifically, $\mathcal{N}_\mathcal{F}(\delta)$ is the minimum number of supremum norm balls with radius δ and centers contained in $\mathcal{F}$ that are necessary to cover $\mathcal{F}$.

Theorem 3.3.1. *Consider the density estimation model as defined in Section 3.2 with density f_0 in the Hölder class $\mathcal{C}_d^\beta([0,1]^d, Q)$. Let $\widehat{f}$ be any (regression) estimator based on the data $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ generated from (3.2.3) and taking values in the function class $\mathcal{F}(F)$, with $F \geq \max\{Q, 1\}$ and $\mathcal{N}_\mathcal{F}(\delta) \geq n$. If K is a kernel of order $\lfloor \beta \rfloor$ with support in $[-1,1]$ and $\|K\|_\infty < \infty$, then, for $n > e$, $(\log(n)/n)^{1/d} \leq h_n \leq 2(\log(n)/n)^{1/d}$ and h_n^{-1} a positive integer, there exist constants C_1, C_2, C_3 only depending on F, K, d, Q, β such that*

$$R(\widehat{f}, f_0) \leq C_1 \frac{\log^2(n) \log(\mathcal{N}_\mathcal{F}(\delta))}{n} + C_2\delta + C_3 \left(\frac{\log(n)}{n}\right)^{\frac{2\beta}{d}} + \frac{16}{3}\Delta_n(\widehat{f}, f_0) + \frac{20}{3} \inf_{f \in \mathcal{F}} \mathbb{E}_\mathbf{X}\left[(f(\mathbf{X}) - f_0(\mathbf{X}))^2\right].$$

As common for oracle inequalities, the upper bound contains an approximation term, a complexity term involving the metric entropy, and the optimization error $\Delta_n(\widehat{f}, f_0)$. For neural networks and other parametrizable function classes, the metric entropy $\log(\mathcal{N}_\mathcal{F}(\delta))$ depends only logarithmically on δ and one can choose $\delta = 1/n$, making the $C_2\delta$ term negligibly small.

Additionally, the bound contains the term $C_3(\log(n)/n)^{2\beta/d}$ that is due to the bandwidth choice $h_n \asymp (\log(n)/n)^{1/d}$ and a term of the order $h_n^{2\beta}$ that can be traced back to Proposition 3.6.3. To decrease the order of the $C_3(\log(n)/n)^{2\beta/d}$ term, it is tempting to aim for a smaller bandwidth $h_n \ll n^{-1/d}$. However, even if the data points are equally spaced in $[0,1]^d$, the distance of two neighboring data points is $n^{-1/d}$. Thus for bandwidth $h_n \ll n^{-1/d}$, it follows from the definition of the kernel density estimator in (3.2.1) that the estimated density degenerates into separate spikes centered around the data points, the generated response variables Y_i become much larger than the true density $f(\mathbf{X}_i)$, and consequently the two-step method that we propose will not work anymore.

The assumption $\mathcal{N}_\mathcal{F}(\delta) \geq n$ in the previous theorem is imposed for convenience and holds for all common nonparametric classes $\mathcal{F}$. The bound is still valid if we replace $\mathcal{N}_\mathcal{F}(\delta)$ by $\mathcal{N}_\mathcal{F}(\delta) \vee n$.

The following lemma shows that a bandwidth h_n with the imposed properties exists.

Lemma 3.3.2. *If $n > 1$, then there exists a h_n, such that $(\log(n)/n)^{1/d} \leq h_n \leq 2(\log(n)/n)^{1/d}$ and h_n^{-1} is a positive integer.*

3.3.1 Neural networks

We study the effect of fitting a deep ReLU network in the regression step of the proposed two-step procedure. We rely on the mathematical formulation of deep neural networks introduced in [127] and briefly recall the details for completeness of the exposition. The rectified linear unit (ReLU) activation function is $\sigma(x) := \max\{x, 0\}$. For any vectors $\mathbf{v} = (v_1, \ldots, v_r)^\top, \mathbf{y} = (y_1, \ldots, y_r)^\top \in \mathbb{R}^r$, we define the shifted activation function $\sigma_{\mathbf{v}}\mathbf{y} := (\sigma(y_1 - v_1), \ldots, \sigma(y_r - v_r))^\top$. The number of hidden layers is denoted by L and the width of the layers is denoted by the width vector $\mathbf{p} = (p_0, \ldots, p_{L+1}) \in \mathbb{N}^{L+2}$. A network with network architecture $(L, \mathbf{p})$ is any function of the form

$$\mathbf{f} : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{L+1}}, \ \mathbf{x} \mapsto \mathbf{f}(\mathbf{x}) = W_L \sigma_{\mathbf{v}_L} W_{L-1} \sigma_{\mathbf{v}_{L-1}} \ldots W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}, \tag{3.3.2}$$

where W_j is a $p_{j+1} \times p_j$ weight matrix and $\mathbf{v}_j \in \mathbb{R}^{p_j}$ is a shift vector. We use the convention that $\mathbf{v}_0 := (0, \ldots, 0)^\top \in \mathbb{R}^{p_0}$. Denote the maximum entry norm of a matrix W by $\|W\|_\infty$. The class of ReLU networks with architecture $(L, \mathbf{p})$ and parameters bounded in absolute value by one is

$$\mathcal{F}(L, \mathbf{p}) := \left\{ \mathbf{f} \text{ is of the form of (3.3.2) } : \max_{j \in \{0, \ldots, L\}} \|W_j\|_\infty \vee |\mathbf{v}_j|_\infty \leq 1 \right\}.$$

For a matrix W denote the counting norm (number of non-zero entries) by $\|W\|_0$. We are interested in sparsely connected networks where the number of non-zero or active parameters is small compared to the total number of parameters. For this we define the class of s-sparse networks, that are bounded in uniform norm by F, as

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

Definition 3.3.3 (Two-stage neural network density estimator)**.** The two-stage neural network estimator is defined as follows. In the first step, we generate response variables as in (3.2.3) and in the second step, we fit a neural network from the class $\mathcal{F}(L, \mathbf{p}, s, F)$ to the augmented sample $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$.

3.3.2 Structural constraints: compositions of functions

Deep neural networks are built by computing individual layers. Previously derived statistical theory has shown that they are well-suited to pick up compositional structure in the regression function, [71, 113, 11, 127, 77]. In this chapter we follow the composition structure introduced in [127] and impose it on the multivariate density f_0, that is, we assume that $f_0 = g_q \circ g_{q-1} \circ \ldots \circ g_1 \circ g_0$, with $g_i : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}$.

Denote by $g_i = (g_{ij})^\top_{j=1,\ldots,d_{i+1}}$ the components of g_i and let t_i be the maximal number of variables on which each of the g_{ij} depends. It always holds that $t_i \leq d_i$ and for certain models t_i can be much smaller than d_i. Section 3.4 provides examples of densities where this is the case. As we consider density estimation on $[0,1]^d$, it follows that $d_0 = d$, $a_0 = 0$, $b_0 = 1$ and $d_{q+1} = 1$. Since g_{ij} depends on t_i variables, we also interpret it as a function $[a_i, b_i]^{t_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}$ whenever this is convenient. Denote by α_i the smoothness of each of the functions g_{ij}. Then $g_{ij} \in C^{\alpha_i}_{t_i}([a_i,b_i]^{t_i}, Q_i)$ and the space of compositions of these smooth functions is given by

$$\begin{aligned}\mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\alpha}, Q') := \Big\{ & f = g_q \circ \ldots \circ g_0 : g_i = (g_{ij})_j : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}, \\ & g_{ij} \in C^{\alpha_i}_{t_i}([a_i,b_i]^{t_i}, Q'), \text{ for some } |a_i|, |b_i| \leq Q' \Big\}.\end{aligned} \tag{3.3.3}$$

If two functions $h, g : \mathbb{R} \to \mathbb{R}$ have respective smoothness $\alpha_h, \alpha_g \leq 1$ then it follows from the definition of the Hölder space that the composition $f := g \circ h$ has smoothness $\alpha_h \alpha_g$. For $\alpha_h > 1$ or $\alpha_f > 1$, this is not necessarily true anymore. It turns out that the convergence rates for a compositional function in $\mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\alpha}, Q')$ are governed by a notion of effective smoothness indices which are defined as

$$\alpha_i^* := \alpha_i \prod_{\ell = i+1}^{q} (\alpha_\ell \wedge 1).$$

Indeed, in the nonparametric regression model with i.i.d. observations the minimax estimation rate is up to $\log(n)$-terms

$$\phi_n := \max_{i=0,\ldots,q} n^{-\frac{2\alpha_i^*}{2\alpha_i^* + t_i}}, \tag{3.3.4}$$

cf. [127]. A function can be represented as a composition in different ways. In the function representation $f = g_q \circ \ldots \circ g_0$, the α_i, t_i and the components $g_0, \ldots, g_q$ are not identifiable. Since we are only interested in estimating the density f_0 this does not constitute a problem.

The oracle inequality in Theorem 3.3.1 together with the approximation and covering entropy bound results for deep ReLU networks from [127] yields a convergence rate result for the proposed two-stage neural networks estimator.

Theorem 3.3.4 (Main convergence rates results). *Consider the multivariate density estimation model as defined in Section 3.2 with density f_0 in the class $\mathcal{C}^\beta_d([0,1]^d, Q) \cap \mathcal{G}(q, \mathbf{d}, \mathbf{t}, \boldsymbol{\alpha}, Q)$. For K a kernel of order $\lfloor \beta \rfloor$ with support in $[-1,1]$ and $\|K\|_\infty < \infty$, let $\widehat{f}_n$ be an estimator, based on the data $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$ generated from (3.2.3), taking values in the network class $\mathcal{F}(L, (p_0, \ldots, p_{L+1}), s, F)$ with parameters satisfying*

(i) $F \geq \max\{Q, 1\}$,
(ii) $\mathcal{N}_{\mathcal{F}}(n^{-1}) \geq n$,
(iii) $\sum_{i=1}^{q} \log_2(4t_i \vee 4\alpha_i) \log_2(n) \leq L \lesssim n\phi_n$,
(iv) $n\phi_n \lesssim \min_{i=1,\ldots,L,} p_i$,
(v) $s \asymp n\phi_n \log(n)$.

If $n > e$, then there exist constants C_4, C_5 only depending on $q, \mathbf{d}, \boldsymbol{\alpha}, \mathbf{t}, F, \beta, K$ and the implicit constants in (iii), (iv) and (v), such that if the optimization error satisfies $\Delta_n(\widehat{f}_n, f_0) \leq C_4 L \max(\phi_n \log^4(n), n^{-2\beta/d})$, we have

$$R(\widehat{f}_n, f_0) \leq C_5 L \max\Big(\phi_n \log^4(n), n^{-2\beta/d}\Big).$$

Any admissible compositional structure $f = g_q \circ \ldots \circ g_0$ leads to an upper bound on the risk. The estimator achieves therefore the fastest convergence rate among all possible representations.

Choosing depth $L \asymp \log(n)$, the convergence rate for the learned network $\widehat{f}$ is thus $\phi_n + n^{-2\beta/d}$, up to $\log(n)$-factors. The $n^{-2\beta/d}$-term is due to the kernel density estimator in the first step and already occurs in the general oracle inequality, see also the discussion after Theorem 3.3.1. Assuming that the true density f_0 is β-Hölder smooth without any further structural assumption, the minimax rate is up to $\log(n)$-factors, the standard nonparametric rate $n^{-2\beta/(2\beta+d)}$, [137].

If the density exhibits a compositional structure, it is now of interest to understand which of the two terms ϕ_n and $n^{-2\beta/d}$ will drive the convergence rate. If the compositional structure is strong enough to make ϕ_n small but β is small compared to d, then $n^{-2\beta/d}$ dominates the convergence rate. This is faster than the standard nonparametric rate $n^{-2\beta/(2\beta+d)}$ but still suffers from the curse of dimensionality.

If $2\beta \geq d$, then $n^{-2\beta/d} = O(n^{-1})$. Since $\phi_n \gg n^{-1}$, the rate is in this case always of order ϕ_n (up to log-factors). The condition $2\beta \geq d$ appears frequently in the literature on nonparametric statistics and empirical risk minimization. For $d = 1$, $2\beta > 1$ is known to be a necessary condition for nonparametric density estimation and nonparametric regression to be asymptotically equivalent if all densities are bounded from below [107, 116]. This condition seems also necessary to ensure that the nonparametric least squares estimator achieves the nonparametric rate, see e.g. Section 6.1 in [130]. Barron [8] showed that shallow neural networks can circumvent the curse of dimensionality under a Fourier criterion. A sufficient, but not necessary condition for this Fourier criterion to be finite is that the partial derivatives up to the least integer β such that $2\beta \geq d + 2$ are square-integrable, see Example 15 in Section IX of [7].

In the next section, we provide more explicit examples of densities that satisfy the compositional assumption and attain the convergence rate ϕ_n.

3.4 Examples of multivariate densities with compositional structure

Compositional structures arise naturally in density modelling. One possibility to see this is to rewrite the joint density f as a product

$$f(x_1,\ldots,x_d)=f(x_d|x_1,\ldots,x_{d-1})\cdot\ldots\cdot f(x_2|x_1)f(x_1).$$

Each factor $f(x_i|x_1,\ldots,x_{i-1})$ is a function of i variables. But the effective number of variables can be much smaller under conditional independence of the variables. When $\mathbf{X}=(X_1,\ldots,X_d)^\top$ is generated for instance from a Markov chain, X_i only depends on X_{i-1} and the density is a product of bivariate conditional densities

$$f(x_1,\ldots,x_d)=f(x_d|x_{d-1})\cdot\ldots\cdot f(x_2|x_1)f(x_1). \tag{3.4.1}$$

Such a structure could occur if the individual data vectors are recordings from a time series, that is, every observation $\mathbf{X}_i=(X_{i,1},\ldots,X_{i,d})^\top$ contains measurements of the same quantity taken at d different times instances. We now assume that the density is of the form

$$f(x_1,\ldots,x_d)=\prod_{I\in\mathcal{R}}\psi_I(x_I), \tag{3.4.2}$$

with $\mathcal{R}\subseteq\{S\subset\{1,\ldots,d\},|S|\leq r\}$, r a given number, $x_I=(x_i)_{i\in I}$, and ψ_I nonnegative functions. Observe that $|\mathcal{R}|\leq\sum_{s=1}^r\binom{d}{s}$.

Lemma 3.4.1. *Consider a density f of the form (3.4.2). If all the functions ψ_I in the decomposition satisfy $\psi_I\in C_{r_I}^{\gamma}([0,1]^{r_I},Q)$ for some $r_I\leq r$, then the density f can be rewritten as a composition $g_1\circ g_0$ of the form (3.3.3), with $(d_0,d_1)=(d,|\mathcal{R}|)$, $(t_0,t_1)=(r,|\mathcal{R}|)$, $(\alpha_0,\alpha_1)=(\gamma,\zeta)$, and ζ arbitrarily large.*

Under the combined conditions of Lemma 3.4.1 and Theorem 3.3.4, the proposed two-step density estimator achieves, up to $\log(n)$-factors, the convergence rate

$$n^{-\frac{2\gamma}{2\gamma+r}}\vee n^{-\frac{2\beta}{d}}, \tag{3.4.3}$$

with β the (global) Hölder smoothness of the joint density f. If $\beta=\gamma$, that is, the effective smoothness γ coincides with the global Hölder smoothness β of f, then the achieved rate is $n^{-\frac{2\gamma}{2\gamma+r}}$ if $\gamma\geq(d-r)/2$ and $n^{-\frac{2\beta}{d}}$ if $\gamma\leq(d-r)/2$.

Next, we discuss three examples of models that are of the form (3.4.2).

Independent variables

If $\mathbf{X} = (X_1, \dots, X_d)$ is a vector containing independent random variables, the joint density is given by

$$f(x_1, \dots, x_d) = \prod_{i=1}^{d} f_i(x_i), \tag{3.4.4}$$

where f_i is the marginal density of X_i. We assume that f_i is γ_i-Hölder. If we are unaware of the independence and simply use multivariate kernel density estimators, we will suffer from the curse of dimensionality as demonstrated for Gaussian densities and Gaussian kernels in Chapter 7 of [131].

Observe that (3.4.4) is of the form (3.4.2), with $\mathcal{R}$ the set of singletons. Thus under the combined conditions of Lemma 3.4.1 and Theorem 3.3.4, we get, up to $\log(n)$-factors, the convergence rate $n^{-2\gamma/(2\gamma+1)} \vee n^{-2\beta/d}$, with β the (global) Hölder smoothness of the joint density f. The construction in Lemma 3.4.1 implies that $\beta \geq \gamma = \min_{i=1,\dots,d} \gamma_i$. The next result shows that in this case we necessarily have equality $\beta = \gamma$. In other words the smoothness of the joint density f has to be equal to the (effective) smoothness of the least smooth marginal density.

Lemma 3.4.2. *Consider a density f of the form* (3.4.4)*. If one of the marginal densities f_i is at most α-Hölder smooth, then f is at most α-Hölder smooth.*

Graphical models

Let $(X_1, \dots, X_d)$ be a d-dimensional random vector. An undirected graphical model (or Markov random field) is defined by a graph with d nodes representing the d random variables. In this graph, no edge between node i and j is drawn if and only if X_i, X_j are conditionally independent given all the other variables $\{X_1, \dots, X_d\} \setminus \{X_i, X_j\}$. A clique in a graph is any fully connected subgraph. When the joint density $f(x_1, \dots, x_d)$ is strictly positive with respect to a σ-finite product measure, the Hammersley-Clifford theorem states that

$$f(x_1, \dots, x_d) = \prod_{C \in \mathcal{C}} \psi_C(x_C), \tag{3.4.5}$$

where $\mathcal{C}$ is the set of all cliques in the graph and ψ_C are suitable functions called potentials [19, 85]. As we consider densities supported on $[0,1]^d$, one can take as dominating product measure the uniform distribution on $(0,1)^d$ and the condition requires that the density is strictly positive on $(0,1)^d$. There is no clear link between the potentials and marginal densities.

Assuming that the true density f_0 satisfies (3.4.5) with largest clique size r and all potentials having Hölder smoothness γ, Lemma 3.4.1 implies that, under the conditions of Theorem 3.3.4, the two-step estimator is able to exploit the underlying low-dimensional structure and achieves the rate $n^{-2\gamma/(2\gamma+r)} \vee n^{-\frac{2\beta}{d}}$, up to $\log(n)$-factors.

Bayesian networks

Bayesian network models are widely used to model for instance medical expert systems [78, 59] and causal relationships [109]. As in the previous section, consider a d-dimensional random vector $(X_1, \ldots, X_d)$. In a Bayesian network, the dependence relationships of the variables are encoded in a directed acyclic graph with nodes $\{1, \ldots, d\}$ [109, 78, 21, 79]. A directed acyclic graph (DAG) is a directed graph that contains no cycles, meaning one cannot visit the same node twice by following a path along the direction of the edges. The parents $\mathrm{pa}(i)$ of a node i are all nodes that have an edge pointing to node i.

The DAG underlying a Bayesian network is constructed such that each variable X_i is conditionally independent of all other variables given the parents $X_{\mathrm{pa}(i)} := \{X_j : j \in \mathrm{pa}(i)\}$ in the graph. The joint density can now be written as product of conditional densities

$$f(x_1, \ldots, x_d) = f_d\big(x_d | x_{\mathrm{pa}(d)}\big) \cdot \ldots \cdot f_1\big(x_1 | x_{\mathrm{pa}(1)}\big). \tag{3.4.6}$$

In particular, if $X_1, \ldots, X_d$ are generated from a Markov chain, the corresponding DAG is $X_1 \to X_2 \to \ldots \to X_d$. Thus $\mathrm{pa}(j) = \{j-1\}$ for $j > 1$, and we recover (3.4.1).

Assuming that the true density f_0 satisfies (3.4.6), that no node in the DAG has more than r parents, and that all conditional densities $f_d\big(x_i | x_{\mathrm{pa}(i)}\big)$ have Hölder smoothness γ, Lemma 3.4.1 shows that, under the conditions of Theorem 3.3.4, the two-step estimator achieves the rate of convergence $n^{-2\gamma/(2\gamma+r)} \vee n^{-\frac{2\beta}{d}}$, up to $\log(n)$-factors.

3.4.1 Copulas

Copulas are widely employed to model dependencies between variables and to construct multivariate distributions, [104, 28, 35]. Denote by F the multivariate distribution, with marginals $F_1(x_1), \ldots, F_d(x_d)$ and density f. Sklar's theorem states that there exists a (unique) d-dimensional copula C (a multivariate distribution function with uniformly distributed marginals on $[0,1]$) such that $F(\mathbf{x}) = C(F_1(x_1), \ldots, F_d(x_d))$. The density f can then be rewritten by the chain rule as

$$f(\mathbf{x}) = c\big(F_1(x_1), \ldots, F_d(x_d)\big) \prod_{i=1}^{d} f_i(x_i), \tag{3.4.7}$$

where $f_i(x_i) = F_i'(x_i)$ is the marginal density with respect to x_i and c is the density of C (assuming that all these densities exist). For a reference, see Section 2.3 of [104].

Lemma 3.4.3. *Consider a density f of the form (3.4.7). If $c \in C_d^{\gamma_c}([0,1]^d, Q_c)$ and $f_i \in C_1^{\gamma_i}([0,1], Q_i)$, for $i = 1, \dots, d$, then, the density f can be rewritten as a composition $g_2 \circ g_1 \circ g_0$ of the form (3.3.3), with $(d_0, d_1, d_2) = (d, 2d, d+1)$, $(t_0, t_1, t_2) = (1, d, d+1)$, $(\alpha_0, \alpha_1, \alpha_2) = (\min_{i=1,\dots,d} \gamma_i, \gamma_c, \gamma)$, and γ arbitrarily large.*

Assume that the true density is of the form (3.4.7), that all marginals f_i have the same Hölder smoothness $\gamma_1 = \ldots = \gamma_d$, that $\beta = \gamma_c \wedge \gamma_1$, and that all the conditions on the kernel and the network architecture underlying Theorem 3.3.4 are satisfied. Applying the decomposition of the density in Lemma 3.4.3, Theorem 3.3.4 yields the convergence rate $n^{-2\gamma_1/(2\gamma_1+1)} \vee n^{-2\gamma_c/(2\gamma_c+d)} \vee n^{-2\beta/d}$, up to $\log(n)$-factors. When $\gamma_c/d \geq \gamma_1 \geq (d-1)/2$, the convergence rate becomes $n^{-2\gamma_1/(2\gamma_1+1)}$ (up to $\log(n)$-factors). The minimax estimation rate for β-Hölder smooth functions without compositional assumption is $n^{-2\beta/(2\beta+d)}$. If the copula c is smoother than the marginals, in the sense that $\gamma_c > \gamma_1 = \beta$, then the obtained convergence rate is faster than the standard nonparametric rate.

As example, consider the d-variate Farlie-Gumbel-Morgenstern copula family with parameter vector $\boldsymbol{\theta}$, which has copula density

$$c_{\boldsymbol{\theta}}(u_1, \dots, u_d) = 1 + \sum_{r=2}^{d} \sum_{1 \leq j_1 < \cdots < j_r \leq d} \theta_{j_1 \dots j_r} \prod_{k=1}^{r} (1 - 2u_{j_k}),$$

for a parameter vector $\boldsymbol{\theta}$ satisfying

$$1 + \sum_{r=2}^{d} \sum_{1 \leq j_1 < \cdots < j_r \leq d} \theta_{j_1 \dots j_r} \prod_{k=1}^{r} \xi_{j_k} \geq 0 \quad \text{for all } \xi_{j_k} \in \{-1, 1\},$$

[64, 39, 43]. The double summation sums over all $2^d - d - 1$ subsets of $\{1, \dots, d\}$ with at least two elements. Since the input of the copula comes from the distribution functions of the marginals, it holds that $(u_1, \dots, u_d) \in [0,1]^d$. This implies $v_j := (1 - 2u_j) \in [-1, 1]$, and by Lemma 3.7.1, $\mathbf{v} \mapsto \prod_{k=1}^{r} v_{j_k} \in C_r^{\gamma}([-1,1]^r, 2^r)$, for all $\gamma \geq r+1$. Together with the chain rule this yields $\mathbf{u} \mapsto \prod_{k=1}^{r}(1 - 2u_{j_k}) \in C_r^{\gamma}([-1,1]^r, 4^r)$. The derivative of a sum is the sum of the derivatives and therefore the triangle inequality implies that when $|\boldsymbol{\theta}|_\infty \leq 1$ then $c_{\boldsymbol{\theta}} \in C_r^{\gamma}([-1,1]^d, (2^d - d)4^d)$, for all $\gamma \geq d+1$. So for this family of copulas, the effective smoothness of the composition is determined by the smoothness of the marginals. If γ_m is the Hölder smoothness of the least smooth marginal, then $\phi_n = n^{-2\gamma_m/(2\gamma_m+1)}$. This shows that if $\beta = \gamma_m$, then under the conditions of Theorem 3.3.4, the convergence rate of the proposed two-step estimator is, up to $\log(n)$-factors, $n^{-2\gamma_m/(2\gamma_m+1)}$ whenever $\gamma_m \geq (d-1)/2$ and $n^{-2\gamma_m/d}$ whenever $\gamma_m \leq (d-1)/2$.

Explicit low-dimensional copula structures can be imposed using the fact that a d-dimensional copula density factorizes into a product of $d(d-1)/2$ bivariate (conditional) copula densities [101, 14, 1, 36]. The key ingredient in this argument is to successively rewrite the conditional densities using the formula $f_{X|Y}(x|y) = c_{X,Y}(F_X(x), F_Y(y))f_X(x)$, where $c_{X,Y}$ denotes the bivariate copula density of (X, Y). The decomposition into bivariate copulas is not unique. Already for three variables (X, Y, Z), there are two possible decompositions, namely

$$f_{X|Y,Z}(x|y,z) = c_{X,Y|Z}\big(F_{X|Z}(x|z), F_{Y|Z}(y|z) \,|\, z\big) f_{X|Z}(x|z)$$

and a second decomposition that interchanges the roles of y and z. The so-called simplifying assumption [136, 101, 36] states that all the bivariate copulas in the decomposition are independent of the conditioned variables, in other words

$$c_{i,j|k}(F_{i|k}(x_i|x_k), F_{j|k}(x_j|x_k)|x_k) = c_{i,j|k}\big(F_{i|k}(x_i|x_k), F_{j|k}(x_j|x_k)\big).$$

For the remainder of this section, we will assume that the simplifying assumption holds.

A way to define such decompositions is by relying on regular vines, [101, 14, 1, 36]. A vine on d variables $X_1, \ldots, X_d$ is a set of trees $(T_1, \ldots, T_r)$, such that the nodes of the first tree T_1 are $u_1, \ldots, u_d$. The nodes of the tree T_i, for $i = 2, \ldots, r$, are (a subset of) the edges of the tree T_{i-1}. For a regular vine it furthermore holds that $r = d - 1$, that two edges in a tree can only be joined by an edge in the next tree if these edges share a common node and that the set of nodes of T_i has to be equal to the set of edges of T_{i-1}

Any regular vine on $(X_1, \ldots, X_d)$ defines a factorization of a d-dimensional copula, by associating a bivariate copula density to each edge in any of the trees. Copulas defined in this way are called vine-copulas.

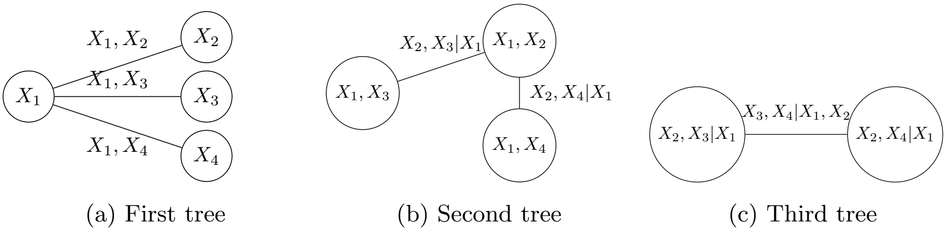
Figure 3.4.1 shows an example of a regular vine with four variables. Regular vines such as the one in Figure 3.4.1, where each tree has one node that has an edge to all other nodes in that tree, are known as canonical-vines [1] or C-vines [36]. The density corresponding to a canonical vine on d variables (up to renumbering the variables) is given by

$$\prod_{k=1}^{d} f_k(x_k) \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{j,j+i|1,\ldots,j-1}\big(F(x_j|x_1, \ldots, x_{j-1}), F(x_{j+i}|x_1, \ldots, x_{j-1})\big)$$

Another type of regular vine is the D-vine, [1, 36]. In a D-vine no node in any tree is connected to more than two edges. Figure 3.5.2 shows the first tree of a D-vine on d

variables. The density corresponding to a D-vine on d variables (up to renumbering the variables) is given by

$$\prod_{k=1}^{d} f_k(x_k)$$
$$\cdot \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{i,i+j|i+1,\ldots,i+j-1}\big(F(x_i|x_{i+1},\ldots,x_{i+j-1}), F(x_{i+j}|x_{i+1},\ldots,x_{i+j-1})\big).$$



(a) First tree (b) Second tree (c) Third tree

Figure 3.4.1: Example of a regular vine on four variables. Another example is given in Figure 3.5.2.

If two random variables X_1, X_2 are conditionally independent given X_3, then $c_{1,2|3} = 1$. If such conditional independence relations hold, one can simplify the vine-structure. For example consider the vine on four variables in Figure 3.4.1. In the (very simplified) case that X_2 and X_3 are independent given X_1, that X_2 and X_4 are independent given X_1, and that X_3 and X_4 are independent given X_1 and X_2, only the bivariate copulas on the edges of the first tree (Figure 3.4.1a) appear in the decomposition, cf. Section 3 of [1]. More generally, suppose that there exists a canonical vine on d variables such that the bivariate (conditional) copulas associated with all the trees except the first one are equal to one, then under the simplifying assumption, the decomposition becomes

$$f(\mathbf{x}) = \prod_{k=1}^{d} f_k(x_k) \prod_{i=2}^{d} c_{1,i}\big(F(x_1), F(x_i)\big). \tag{3.4.8}$$

Here we use that X_1 is the root of the first tree, which can always be achieved by renumbering the variables. In the case of a D-vine the decomposition (up to renumbering) becomes

$$f(\mathbf{x}) = \prod_{k=1}^{d} f_k(x_k) \prod_{i=1}^{d-1} c_{i,i+1}\big(F(x_i), F(x_{i+1})\big). \tag{3.4.9}$$

Vine copulas where the bivariate copulas associated with all trees except the first one are equal to the independence copula can be interpreted as Markov tree models [26, 70].

Lemma 3.4.4. *Consider a density f of the form (3.4.8) or (3.4.9). If $f_i \in C_1^{\gamma_{m,i}}([0,1], Q_{m,i})$, for $i = 1, \dots, d$, and all bivariate copulas are in $C_2^{\gamma_c}([0,1]^2, Q_c)$, then, the function f can be written as a composition $g_2 \circ g_1 \circ g_0$, with $(d_0, d_1, d_2) = (d, 2d, 2d-1)$, $(t_0, t_1, t_2) = (1, 2, 2d-1)$, $(\alpha_0, \alpha_1, \alpha_2) = (\min_{1 \le i \le d} \gamma_{m,i}, \gamma_c, \gamma)$, where γ is arbitrarily large.*

If we assume that $\gamma_c = \beta = \min_{1 \le i \le d} \gamma_{m,i}$, then under the combined conditions of Theorem 3.3.4 and Lemma 3.4.4, the proposed two-step neural network estimator achieves the rate $n^{-2\beta/(2\beta+2)} \vee n^{-2\beta/d}$ up to $\log(n)$ factors. If $d > 2$, this rate is faster than the minimax rate without structure $n^{-2\beta/(2\beta+d)}$. Furthermore when $\beta \ge d/2 - 1$ the rate equals $n^{-2\beta/(2\beta+2)}$, up to $\log(n)$-factors. If instead of assuming that $\gamma_c = \beta = \min_{1 \le i \le d} \gamma_{m,i}$, we assume that $\gamma_c \ge 2\beta = 2\min_{1 \le i \le d} \gamma_{m,i}$, that is, the copulas have at least twice the Hölder smoothness of the marginals, then the rate becomes $n^{-2\beta/(2\beta+1)} \vee n^{-2\beta/d}$, up to $\log(n)$-factors.

3.4.2 Mixture distributions

If the true density is a mixture and all mixture components can be estimated by a fast convergence rate, it should be possible to also estimate the true density with a fast rate. Below we make this precise, assuming that the true density is of the form

$$f_0 = a_1 f_1 + \ldots + a_r f_r \tag{3.4.10}$$

with non-negative mixture weights $a_1, \ldots a_r$ summing up to one and densities f_j in the compositional Hölder space $\mathcal{G}(q_j, \mathbf{d}_j, \mathbf{t}_j, \boldsymbol{\alpha}_j, Q')$ defined in (3.3.3). Compositional spaces are not closed under linear combinations and therefore there is no natural embedding of f into the compositional spaces of the f_j's. As shown next, the convergence rate for estimation of f still coincides with the maximum among all convergence rates for estimation of individual mixture components f_j.

Theorem 3.4.5 (Convergence rates for mixture distributions)**.** *Consider the density estimation model as defined in Section 3.2 with density $f_0 = \sum_{i=1}^{r} a_i f_i$, where $a_1, \ldots a_r$*

are non-negative mixture weights summing up to one, and with $f_j \in \mathcal{C}_d^{\beta_j}([0,1]^d, Q) \cap \mathcal{G}(q_j, \mathbf{d}_j, \mathbf{t}_j, \boldsymbol{\alpha}_j, Q)$*, for* $j = 1, \ldots, r$*. Set* $\phi_n^\star = \max_{j=1,\ldots,r} \phi_{n,j}$*, where* $\phi_{n,j}$ *is the rate* (3.3.4) *for estimation of* f_j*. Set* $\beta = \min_{j=1,\ldots,r} \beta_j$*. For* K *a kernel of order* $\lfloor \beta \rfloor$ *with support in* $[-1,1]$ *and* $\|K\|_\infty < \infty$*, let* $\widehat{f}_n$ *be the two-stage density estimator defined in Definition 3.3.3 for the neural network class* $\mathcal{F}(L, (p_0, \ldots, p_{L+1}), s, F)$ *with parameters satisfying*

(i) $F \geq \max\{Q, 1\}$,

(ii) $\mathcal{N}_{\mathcal{F}}(n^{-1}) \geq n$,

(iii) $\max_{j=1,\ldots,r} \sum_{i=1}^{q_j} \log_2(4t_{i,j} \vee 4\alpha_{i,j}) \log_2(n) \leq L \lesssim n\phi_n^\star$,

(iv) $n\phi_n^\star \lesssim \min_{i=1,\ldots,L} p_i$,

(v) $s \asymp n\phi_n^\star \log(n)$.

If n *is large enough, then there exist constants* C_6, C_7 *only depending on* r*,* $(q_j, \mathbf{d}_j, \mathbf{t}_j, \boldsymbol{\alpha}_j)_{j=1}^r$*,* F*,* β*,* K *and the implicit constants in (iii), (iv) and (v) such that if the optimization error satisfies* $\Delta_n(\widehat{f}_n, f_0) \leq C_6 L \max(\phi_n^\star \log^4(n), n^{-\frac{2\beta}{d}})$*, we have*

$$R(\widehat{f}_n, f_0) \leq C_7 L \max\left(\phi_n^\star \log^4(n), n^{-\frac{2\beta}{d}}\right).$$

3.5 Simulations

3.5.1 Methods

In a numerical simulation study we compare the proposed two-step neural network method (named SD for Split Data) as described in Definition 3.3.3 to two other methods. The FD (full data) method follows the same construction as the two-step neural network method but uses for both steps the full dataset without sample splitting. Thus, we have twice as many data for the individual steps, but also incur additional dependence between the regression variables as each of the constructed response variables Y_i depends on the entire dataset (instead of only on the kernel dataset and the corresponding X_i from the regression set). The neural network based methods are moreover compared to a multivariate kernel density estimator (KDE).

As suggested by the theory, for the first step in the SD and FD method, the bandwidths for the kernel density estimator are chosen of the form $c_1(\log(n)/n)^{1/d}$ and $c_2(\log(2n)/2n)^{1/d}$. For the KDE method, the bandwidth is $c_3 n^{-1/(2\beta+d)}$. The constants c_1, c_2, c_3 are determined based on the average of the optimal bandwidths found by 50-fold cross-validation, taking as searchspace $[0.05, 1.1]$ with stepsize 0.005, on five independently generated datasets with sample size $n = 200$ from the true density. Taking $n = 200$ for the calibration is natural as it is the smallest sample size in the simulation environment.

3.5.2 Densities

For the different simulation settings, we generate data from five densities. These densities are called Naive Bayes mixing (NBm), Naive Bayes shifting (NBs), Binary Tree mixing (BTm), Binary Tree shifting (BTs) and Copula (C).

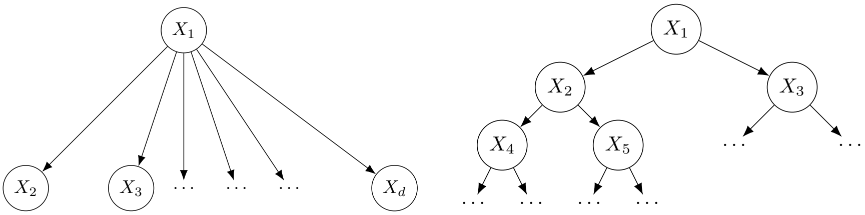
NBm, NBs, BTm, BTs

The densities (NBm) and (NBs) are so-called Naive Bayes networks [78] with DAGs displayed in Figure 3.5.1a and density factorization

$$f(x_1, \dots, x_d) = f_d(x_d|x_1) \cdot \dots \cdot f_2(x_2|x_1) f_1(x_1). \tag{3.5.1}$$

The densities (BTm) and (BTs) are Bayesian networks with DAGs displayed in Figure 3.5.1b and density factorization

$$f(x_1, \dots, x_d) = f_1(x_1) \prod_{j=2}^{d} f_j(x_j|x_{\lceil (j-1)/2 \rceil}). \tag{3.5.2}$$



(a) Naive Bayes DAG (b) Binary tree DAG

Figure 3.5.1: DAG for the Naive Bayes network (a) and the Bayesian network with binary tree structure (b).

For the density f_1 we use the exponential of a standard Brownian motion on $[0, 1]$, normalized such that f_1 integrates to one. We use two different types of conditional densities. The mixing conditional density has mixture weights from the conditioned variable,

$$f_j(x_j|x_i) = x_i h_j(x_j) + (1 - x_i) h_j(1 - x_j), \tag{3.5.3}$$

with h_j a density supported on $[0, 1]$. The shifting conditional density incorporates a shift determined by the conditioned variable,

$$f_j(x_j|x_i) = h_j\big(\max\{x_j - x_i/4, 0\}\big), \tag{3.5.4}$$

with h_j a density supported on the interval $[0, 3/4]$, so that the support of $f_j(\cdot|x_i)$ is ensured to lie in $[0, 1]$.

For the densities (NBm) and (BTm) all conditional densities $f_j(\cdot|\cdot)$ in the factorization are mixing densities (3.5.3). For the densities (NBs) and (BTs) the conditional densities $f_j(\cdot|\cdot)$ in the factorization are shifting densities (3.5.4) if j is divisible by 3 and mixing densities (3.5.3) otherwise.

It remains to choose the density h_j in (3.5.3) and (3.5.4). We consider scenarios containing both smooth and rough densities. For (NBm), (NBs), (BTm) and (BTs) and all j such that $j - 1$ is not divisible by 3, we set

$$h_j(x) = \Big(1 - \frac{2x-1}{d}\Big)\mathbf{1}\big(0 \leq x \leq 1\big). \tag{3.5.5}$$

Viewed as functions on $[0, 1]$, these densities have arbitrarily large Hölder smoothness. The densities take values between $1-1/d$ and $1+1/d$ ensuring that in higher dimensions the joint densities, which are products, neither become extremely small or large.

For (NBm) and (BTm) and all $j > 1$ such that $j - 1$ is divisible by 3, we take as densities h_j the exponential of the Brownian motion on $[0, 1]$, normalized such that h_j integrates to one. Brownian motion has Hölder smoothness $1/2 - \eta$ for any $\eta \in (0, 1/2)$, but is almost surely not 1/2-Hölder smooth [96]. This means that these densities have low regularity.

For (NBs) and (BTs) and all $j > 1$ such that $j - 1$ is divisible by 3, we take as densities h_j the paths of the exponential of the Brownian motion on $[0, 1]$ multiplied with the function $x \mapsto \rho(x) = \max(0, (4x/3)(1 - 4x/3))$ and normalized such that h_j integrates to one. Multiplication with ρ ensures that the support of these densities is in $[0, 3/4]$, as required in the definition (3.5.4).

The conditional densities f_j defined in (3.5.3) and (3.5.4) can be interpreted as compositional functions.

Lemma 3.5.1. *Consider the mixing conditional density f_j in* (3.5.3). *If $h_j \in C_1^{\gamma_j}([0, 1], Q)$, then f_j can be written as the composition $g_2 \circ g_1 \circ g_0$, with $(d_0, d_1, d_2) = (d, 3, 3)$, $(t_0, t_1, t_2) = (1, 1, 3)$, and $(\alpha_0, \alpha_1, \alpha_2) = (\gamma, \gamma_j, \zeta)$, with γ, ζ arbitrarily large.*

Lemma 3.5.2. *Consider the shifting conditional density f_j in* (3.5.4) *If $h_j \in C_1^{\gamma_j}([0, 3/4], Q)$, then f_j can be written as $g_1 \circ g_0$, with $(d_0, d_1) = (d, 1)$, $(t_0, t_1) = (2, 1)$, $(\alpha_0, \alpha_1) = (1, \gamma_j)$.*

The (NBm), (NBs), (BTm) and (BTs) joint densities are thus compositions where the components with low regularity are all univariate functions, making the rate ϕ_n dimensionless. The factorization in (3.5.1) and the composition of Lemma 3.4.1 combined with the composition in Lemma 3.5.1 shows this for the (NBm) model. The factorization in (3.5.1) and the composition Lemma 3.4.1 combined with the compositions in Lemma 3.5.1 and Lemma 3.5.2 show this for the (NBs) model. The factorization in (3.5.2) and the composition of Lemma 3.4.1 combined with Lemma 3.5.1 shows this for the (BTm) model and the factorization in (3.5.2) and the composition of Lemma 3.4.1 combined with the compositions in Lemma 3.5.1 and Lemma 3.5.2 show this for the (BTs) model.

Simulation setup for copula density model

For the copula model, the density (C) is associated to a D-vine copula. The first tree of the D-vine is depicted in Figure 3.5.2. We assume that this first tree captures all the dependencies between the variables. This means that X_i is conditionally independent of X_{i+j} given $X_{i+1}, \dots, X_{i+j-1}$ for all pairs (j, i) with $j \in \{2, \dots, d-1\}$ and $i \in \{1, \dots, d-j\}$.

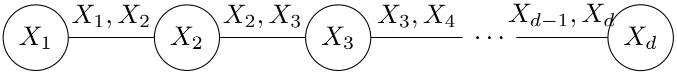


Figure 3.5.2: Structure of the first tree of the D-vine copula used in the simulation.

The bivariate copulas for the density (C) are chosen from the bivariate Farlie-Gumbel-Morgenstern copula family defined via the copula densities $c_{\theta,i,j}(F_i(x_i), F_j(x_j)) = 1 + \theta(1 - 2F_i(x_i))(1 - 2F_j(x_j))$, with parameter $|\theta| \leq 1$. As already shown in Section 3.4.1, these copulas have arbitrarily large Hölder smoothness. If $(i-1)/(d-2) \neq 1/2$, we use the parameter $\theta = -1 + 2(i-1)/(d-2)$ for the bivariate copula. Otherwise we use $\theta = 1/100$. The marginal densities are displayed in Figure 3.5.3. The smoothness of this density is determined by the square root, which has Hölder smoothness 1/2.

$$f_k(x) = \begin{cases} 1 + \frac{1}{2d} - \frac{1}{d}\sqrt{\frac{1}{4} - x}, & \text{if } 0 \leq x < \frac{1}{4} \\ 1 + \frac{1}{2d} - \frac{1}{d}\sqrt{x - \frac{1}{4}}, & \text{if } \frac{1}{4} \leq x < \frac{1}{2} \\ 1 - \frac{1}{2d} + \frac{1}{d}\sqrt{\frac{3}{4} - x}, & \text{if } \frac{1}{2} \leq x < \frac{3}{4} \\ 1 - \frac{1}{2d} + \frac{1}{d}\sqrt{x - \frac{3}{4}}, & \text{if } \frac{3}{4} \leq x \leq 1. \end{cases}$$

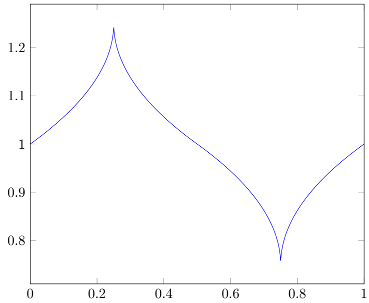


Figure 3.5.3: Marginal density $f_k(x)$ used in the simulated copula model.

The right panel of Figure 3.5.3 displays the graph for $d = 2$. This marginal density is appealing as it has a closed-form expression for the density and the c.d.f. The dependency on d of the marginals is to ensure that the marginal densities remain between $1 - 1/d$ and $1 + 1/d$ in order to prevent numerical instability. Since the Farlie-Gumbel-Morgenstern copula is infinitely smooth, we get from Lemma 3.4.4 that the effective smoothness of the joint density generated from this vine-copula approach is equal to $1/2$ and thus the rate ϕ_n in Theorem 3.3.4 becomes $n^{-1/2}$, up to $\log(n)$-factors.

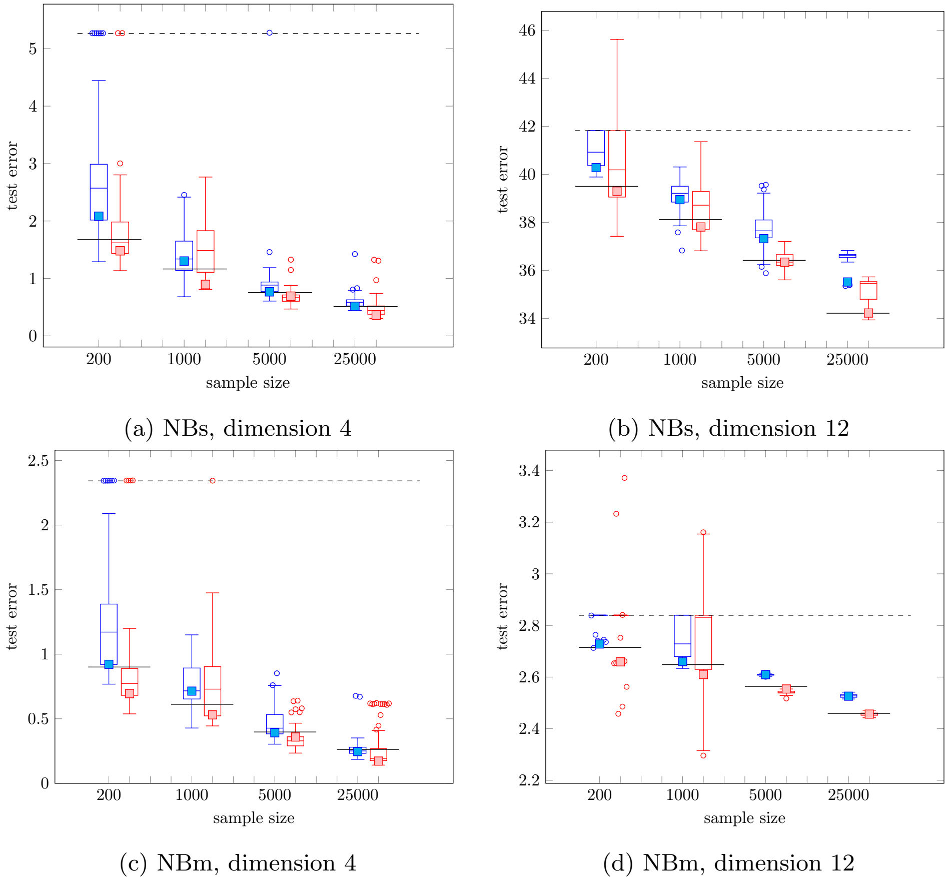
3.5.3 Neural network training setup

For both the SD and FD method, we train 50 neural networks on each training sample with width vector $\mathbf{p} = (d, \lceil (2n)^{1/2} \rceil, \lceil (2n)^{1/2} \rceil, \ldots, \lceil (2n)^{1/2} \rceil, 1)$ and depth $L = \lceil \log_2(2n) \rceil$. Since the derived convergence rate of the two-stage neural network estimator is $\phi_n = n^{\eta' - 1/2}$, for any $\eta' \in (0, 1/2)$, in the (NBm), (NBs), (BTm) and (BTs) settings and $\phi_n = n^{-1/2}$ for the (C) setting, this choice of the network width satisfies the bound in Theorem 3.3.4. The chosen depth is of the order $\log(n)$ suggested by the theory, but there might be a mismatch regarding the constants in the lower bound of Condition (iii) in Theorem 3.3.4. Since the proof of this result does not optimize the constants, we find it more appealing to work with the generic choice $L = \lceil \log_2(2n) \rceil$ in the simulations. Furthermore, Theorem 3.3.4 imposes a sparsity condition on the networks as well as a condition on the maximum norm of the parameters. In the simulation study we use ℓ_2-penalization on the weight matrices and the Glorot uniform initialization [49] to ensure that the parameter values do not become too large. Although these methods do not provide a hard guarantee that the condition on the maximum norm is satisfied, they work reasonably well in practice and the number of learned network parameters that are larger in absolute value than one is small compared to the total number of network parameters. We use pruning (using the TensorFlow model optimization package) to enforce sparsity. The fraction of zero network parameters is chosen as $1 - 2m \log(m) \phi_m / p$, with p the total number of network parameters and $m = 2n$ for the FD method and $m = n$ for the SD method.

The source code is available on GitHub [23].

3.5.4 Simulation Results

We compare the performance of all the methods on 10^6 test samples. This sample is only used for computing the test error and none of the methods has access to the test samples during training. Figures 3.5.4-3.5.6 report the test errors for the five different settings. In each of these settings we consider four data-sets, one of size 200, one of size 1000, one of size 5000 and one of size 25000.



(a) NBs, dimension 4

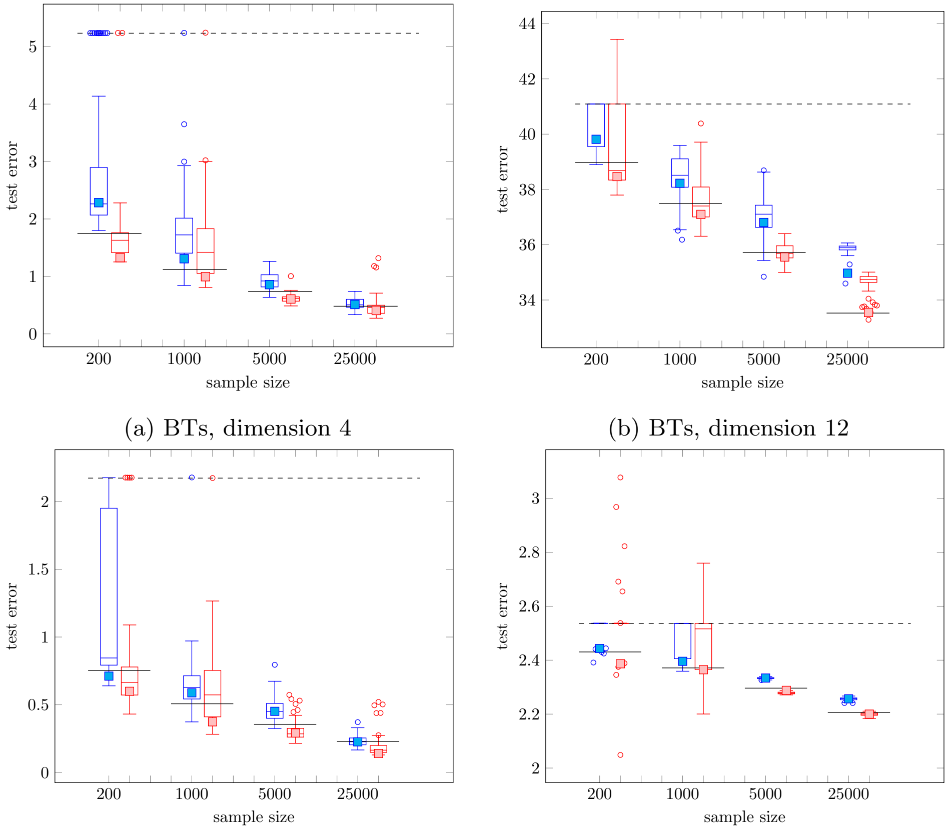
(b) NBs, dimension 12

(c) NBm, dimension 4

(d) NBm, dimension 12

Figure 3.5.4: Test errors for the naive Bayes model. SD in blue, FD in red, KDE black bars. The test error of the network with the lowest training error is indicated by the filled square. The black dashed line is the test error of the zero function. Notice that in the individual plots, the y-axis has different starting points.

For the smaller sample sizes there is in all models some degree of concentration of the test errors of the trained networks around the value of the test error for the zero function. The theory claims that among the sparsely connected networks that satisfy all the imposed conditions, the one with small training error should perform particularly well. To see whether there is an effect, we mark for every simulation setting



(a) BTs, dimension 4

(b) BTs, dimension 12

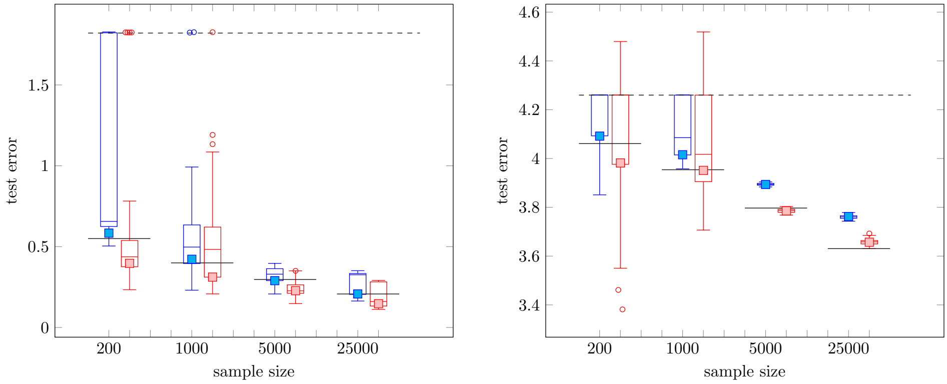
(c) BTm, dimension 4

(d) BTm, dimension 12

Figure 3.5.5: Test errors for the Bayesian network model. SD in blue, FD in red, KDE black bars. The test error of the network with the lowest training error is indicated by the filled square. The black dashed line is the test error of the zero function. Notice that in the individual plots, the y-axis has different starting points.

the test error of the network with the smallest training error by a filled square. This network is the one we use to compare the methods with each other in the discussion below.

To further investigate the relation between training error and test error, we plot for the (NBs) model in dimension four (Figure 3.5.7) and twelve (Figure 3.5.8) the



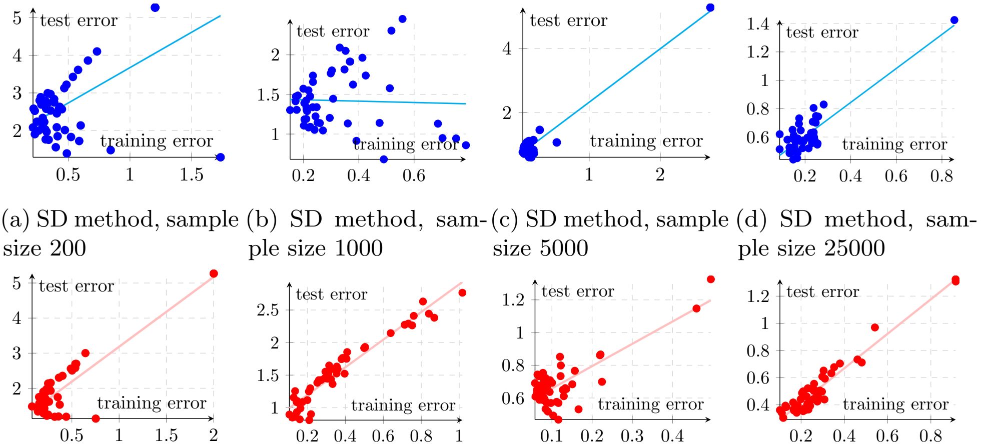
(a) Copula, dimension 4

(b) Copula, dimension 12

Figure 3.5.6: Test errors for the Copula model. SD in blue, FD in red, KDE black bars. The test error of the network with the lowest training error is indicated by the filled square. The black dashed line is the test error of the zero function. Notice that in the individual plots, the y-axis has different starting points.

training error versus the test error of all networks, for both the SD and FD method and for all the four considered sample sizes. The linear line displaying the least squares regression fit has positive slope, except for the SD method with sample size 1000 (in both dimensions four and twelve).

To estimate the joint density depending on four variables, the neural network fits based on the FD method with the lowest training error (indicated by red squares in the plots) seem to perform best for all sample sizes. For density estimation on $[0, 1]^{12}$, the picture is less clear as there are sample sizes for which the KDE method achieves a comparable or even better test error. The test error of the SD method is consistently higher. In dimension 4, it decreases, however, faster than the test errors of the FD and KDE method. The fact that networks with small training errors (filled squares in the plots) perform particularly well suggests that the performance of the FD method could be further increased by fitting much larger networks. Although sample splitting makes the theory tractable, we do not advise to use it in practice. While the idea to transform an unsupervised learning problem into a supervised learning problem and using supervised learning methods is appealing, we feel that considerable future effort is required to transform this into stable and efficient algorithms.



(a) SD method, sample size 200

(b) SD method, sample size 1000

(c) SD method, sample size 5000

(d) SD method, sample size 25000

(e) FD method, sample size 200

(f) FD method, sample size 1000

(g) FD method, sample size 5000

(h) FD method, sample size 25000

Figure 3.5.7: Scatterplot of the test error versus the training error for the (NBs) model in 4 dimensions. The line shows the linear least squares regression fit.

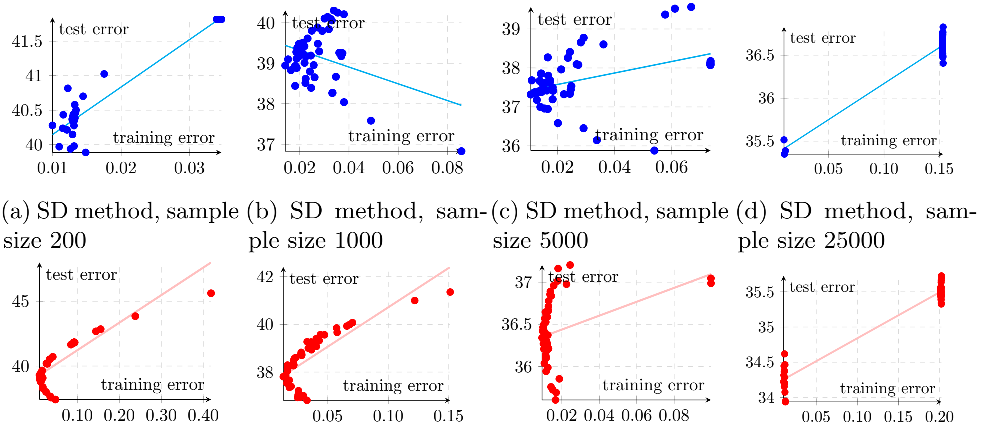
3.6 Proofs for Section 3.3

Proof of Lemma 3.3.2. For all $x \geq 0$, we have $x < 1 + x \leq e^x$ and thus $\log n/n < 1$ as well as $0 < u_n := 2(\log n/n)^{1/d} < 2$ for all $n > 1$. For all $y > 0$, one can find an integer r such that $y/2 \leq 2^r \leq y$. If $y < 2$, we must have $r \leq 0$. Thus, there exists $s \leq 0$ such that $u_n/2 \leq 2^s \leq u_n$. Set $h_n = 2^s$. Since $s \leq 0$, we must have $h_n^{-1} = 2^{-s}$, which is an integer. □

3.6.1 Proof of Theorem 3.3.1

The response variables Y_i in the regression model (3.2.3) are identically distributed, but they are not jointly independent as they all depend through the kernel density estimator on the subsample $(\mathbf{X}'_\ell)_{\ell=1}^n$.

To deal with the dependence induced by the kernel density estimator, we partition the hypercube $[0,1]^d$ into h_n^{-d} hypercubes with sidelength h_n. By construction h_n^{-1} is an integer and therefore no boundary issues arise. The centers of these h_n^{-d} hypercubes are given by the vectors $h_n(k_1 - 1/2, k_2 - 1/2, \ldots, k_d - 1/2)^\top \in [0,1]^d$ with $k_1, k_2, \ldots, k_d \in \{1, \ldots, h_n^{-1}\}$. By numbering these points (the specific numbering



(a) SD method, sample size 200

(b) SD method, sample size 1000

(c) SD method, sample size 5000

(d) SD method, sample size 25000

(e) FD method, sample size 200

(f) FD method, sample size 1000

(g) FD method, sample size 5000

(h) FD method, sample size 25000

Figure 3.5.8: Scatterplot of the test error versus the training error for the (NBs) model in 12 dimensions. The line shows the linear least squares regression fit.

of the points is irrelevant), we assign to each center an index in $\mathcal{J} := \{1, \ldots, h_n^{-d}\}$. The j-th bin $\mathcal{B}_j$ is then the $|\cdot|_\infty$-norm ball of radius $h_n/2$ around the j-th center $C(\mathcal{B}_j)$ in this index set. To avoid that boundary points are in two bins, we include a boundary point only if is not already included in a bin with smaller index in the ordering induced by $\mathcal{J}$. This construction gives a partition of $[0,1]^d$. As each bin is a hypercube with sidelength h_n, the Lebesgue measure is h_n^d (in $\mathbb{R}^d$). The neighborhood of a bin $\mathcal{B}_j$, denoted by $NB(\mathcal{B}_j)$, are all bins whose centers are at most $|\cdot|_\infty$-distance h_n away from the center of $\mathcal{B}_j$, in other words,

$$NB(\mathcal{B}_j) = \bigcup_{\ell : |C(\mathcal{B}_j) - C(\mathcal{B}_\ell)|_\infty \leq h_n} \mathcal{B}_\ell \tag{3.6.1}$$

(In two dimensions this neighborhood is also known as the Moore neighborhood).

We further subdivide the bins into equivalence classes. For all sufficiently large n, $h_n \leq 1/3$ and the hypercube $[0, 3h_n]^d$ contains exactly 3^d bins. Denote by $(j_s)_{s=1}^{3^d}$ the indices of these bins and define the index set $\mathcal{J}_s \subset \mathcal{J}$ by

$$\mathcal{J}_s := \Big\{ \ell \in \mathcal{J} : \frac{1}{3h_n}\big(C(\mathcal{B}_\ell) - C(\mathcal{B}_{j_s})\big) \in \mathbb{Z}^d \Big\}.$$

By construction, the sets $\mathcal{J}_s$ are mutually disjoint and $\bigcup_s \mathcal{J}_s = \mathcal{J}$.

Fix a $j \in \mathcal{J}$. Since the kernel K in the kernel density estimator has bandwidth h_n and support contained in $[-1,1]$, the point estimator $\widehat{f}_{\mathrm{KDE}}(\mathbf{x})$ only depends on the data points from the kernel data set $(\mathbf{X}'_\ell)_{\ell=1}^n$ that are in $NB(\mathcal{B}_j)$.

More generally, for two different indices $j, \widetilde{j} \in \mathcal{J}_s$, $j \neq \widetilde{j}$ and points $\mathbf{x}_1 \in \mathcal{B}_j$, $\mathbf{x}_2 \in \mathcal{B}_{\widetilde{j}}$, the point estimators $\widehat{f}_{\mathrm{KDE}}(\mathbf{x}_1)$ and $\widehat{f}_{\mathrm{KDE}}(\mathbf{x}_2)$ depend on $\{\mathbf{X}'_\ell : \mathbf{X}'_\ell \in NB(\mathcal{B}_j), \ell = 1, \ldots, n\}$ and $\{\mathbf{X}'_\ell : \mathbf{X}'_\ell \in NB(\mathcal{B}_{\widetilde{j}}), \ell = 1, \ldots, n\}$, respectively. The latter two sets are dependent if n is fixed (knowing that a data point is in one of the bins means that there can be at most $n-1$ in any of the other bins). If we instead assume that the sample size of the data set $(\mathbf{X}'_\ell)_{\ell=1}^n$ is not n but $\mathcal{M}$ with $\mathcal{M} \sim \text{Poisson}(n)$, then $\{\mathbf{X}'_\ell : \mathbf{X}'_\ell \in A, \ell = 1, \ldots, \mathcal{M}\}$ and $\{\mathbf{X}'_\ell : \mathbf{X}'_\ell \in B, \ell = 1, \ldots, \mathcal{M}\}$ are independent, whenever A and B are disjoint sets. This will formally be shown in the proof of Lemma 3.6.2. Using Poisson point process theory, we also show in the proof of Lemma 3.6.2 that $\widehat{f}_{\mathrm{KDE}}(\mathbf{x}_1)$ and $\widehat{f}_{\mathrm{KDE}}(\mathbf{x}_2)$ are independent.

The previously described strategy is known as Poissonization, cf. [148] Section 3.5.2., [47], [42] Section 8.3. In particular, we use the following inequality

Lemma 3.6.1. *For $\mathcal{M}$ and $\mathbf{X}'_1, \mathbf{X}'_2, \ldots$ as above, for any function h, and any measurable set A,*

$$\mathbb{P}\Big(\sum_{i=1}^n h(\mathbf{X}'_i) \in A\Big) \leq \sqrt{2e\pi n}\,\mathbb{P}\Big(\sum_{i=1}^{\mathcal{M}} h(\mathbf{X}'_i) \in A\Big).$$

Proof of Lemma 3.6.1. We have

$$\mathbb{P}\Big(\sum_{i=1}^n h(\mathbf{X}'_i) \in A\Big) = \mathbb{P}\Big(\sum_{i=1}^{\mathcal{M}} h(\mathbf{X}'_i) \in A \,\Big|\, \mathcal{M} = n\Big) \leq \frac{\mathbb{P}\big(\sum_{i=1}^{\mathcal{M}} h(\mathbf{X}'_i) \in A\big)}{\mathbb{P}(\mathcal{M} = n)}.$$

Since $\mathcal{M}$ is a Poisson(n) random variable we have that $\mathbb{P}(\mathcal{M} = n) = n^n e^{-n}/n!$. By Stirling's formula, see for example [120], $n! \leq \sqrt{2\pi n}(n/e)^n e^{1/(12n)} \leq \sqrt{2e\pi n}(n/e)^n$ and $1/\mathbb{P}(\mathcal{M} = n) \leq \sqrt{2e\pi n}$. □

While Poissonization removes dependence, the factor $\sqrt{2e\pi n}$ arises in the bounds.

Proving oracle inequalities for the risk $R(\widetilde{f}, f_0) := \mathbb{E}_{f_0,\mathbf{X}}[(\widetilde{f}(\mathbf{X}) - f_0(\mathbf{X}))^2]$ in the standard i.i.d. setting typically first shows an oracle inequality for the empirical risk $\widehat{R}_n(\widehat{f}, f_0)$ as

$$\widehat{R}_n(\widehat{f}, f_0) := \mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n \big(\widehat{f}(\mathbf{X}_i) - f_0(\mathbf{X}_i)\big)^2\right].$$

Here empirical refers to the fact that the estimator $\widehat{f}$ is evaluated at the data points $\mathbf{X}_1, \ldots, \mathbf{X}_n$. The derivation of an oracle inequality for the empirical risk can be further subdivided into several steps. The bound below refers to the step where our setting and the i.i.d. case differ the most. The proof (presented in Section 3.9) relies heavily on the construction of the bins above combined with Poissonization. Recall that $\varepsilon_i = Y_i - f(\mathbf{X}_i)$.

Lemma 3.6.2. *For any estimator $\widehat{f}$ as in Theorem 3.3.1, any fixed $f \in \mathcal{F}$ and $\log^2(n)\log(\mathcal{N}_{\mathcal{F}}(\delta)) \leq n$, it holds that*

$$\begin{aligned}
&\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^{n}\epsilon_i(\widehat{f}(\mathbf{X}_i) - f(\mathbf{X}_i))\right]\right| \\
&\leq 2^{d+6}14e^2\|K\|_{\infty}^{2d}F^3 3^{\frac{7d}{2}}\left(\sqrt{\widehat{R}_n(\widehat{f}, f_0)}\log(n)\sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}} + \log(n)\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n} + \delta\right) \\
&\quad + \frac{46F^2 2^d\|K\|_{\infty}^{d}}{n} + 8h_n^{2\beta}F^2d^{2\beta}\|K\|_{\infty}^{2d} + \frac{\mathbb{E}_{\mathbf{X}}[(f_0(\mathbf{X}) - f(\mathbf{X}))^2]}{4} + \frac{\widehat{R}_n(\widehat{f}, f_0)}{4}.
\end{aligned}$$

With this lemma in place, we can prove (see Section 3.9) the following bound on the empirical risk. This is similar to step (III) in the proof of the oracle inequality of Lemma 4 in [127].

Proposition 3.6.3. *For any estimator $\widehat{f}$ as in Theorem 3.3.1, any fixed $f \in \mathcal{F}$ and $\log^2(n)\log(\mathcal{N}_{\mathcal{F}}(\delta)) \leq n$,*

$$\begin{aligned}
\widehat{R}_n(\widehat{f}, f_0) &\leq \delta 2^{d+6}38e^2\|K\|_{\infty}^{2d}F^3 3^{\frac{9d}{2}} \\
&\quad + \frac{10}{3}\mathbb{E}_{\mathbf{X}}\left[(f(\mathbf{X}) - f_0(\mathbf{X}))^2\right] + \frac{8}{3}\Delta_n(\widehat{f}, f_0) \\
&\quad + 2^{d+6}38e^2\|K\|_{\infty}^{2d}F^3 3^{\frac{7d}{2}}\log(n)\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n} \\
&\quad + \frac{124F^2 2^d\|K\|_{\infty}^{d}}{n} + 22h_n^{2\beta}F^2d^{2\beta}\|K\|_{\infty}^{2d} \\
&\quad + 4^{d+7}19^2e^4\|K\|_{\infty}^{4d}F^6 3^{7d}\log^2(n)\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}.
\end{aligned}$$

We now have all ingredients to finish the proof of Theorem 3.3.1.

Proof of Theorem 3.3.1. If $\log^2(n)\log(\mathcal{N}_{\mathcal{F}}(\delta)) \geq n$, the statement follows with $C_1 = 4F^2$ by observing that $R(\widehat{f}, f_0) \leq 4F^2$.

It remains to consider the case $\log^2(n)\log(\mathcal{N}_{\mathcal{F}}(\delta)) \leq n$. The proof of Lemma 4, Part (I) in [127] states that for any $\epsilon \in (0,1]$,

$$\begin{aligned}&(1-\epsilon)\widehat{R}_n(\widehat{f},f_0) - \frac{F^2}{n\epsilon}\Big(15\log\big(\mathcal{N}_{\mathcal{F}}(\delta)\big)+75\Big) - 26\delta F\\ &\leq R(\widehat{f},f_0) \leq (1+\epsilon)\bigg(\widehat{R}_n(\widehat{f},f_0) + (1+\epsilon)\frac{F^2}{n\epsilon}\Big(12\log(\mathcal{N}_{\mathcal{F}}(\delta))+70\Big) + 26\delta F\bigg).\end{aligned} \tag{3.6.2}$$

This lemma for the standard nonparametric regression problem relates the risk to its empirical counterpart. The inequality and its proof only depend on the $\mathbf{X}_i$ and on the function class $\mathcal{F}$, not on the noise or the response variables Y_i. Since in our regression model (3.2.3) the variables $\mathbf{X}_i$ are i.i.d. (the dependence is induced by the response variables Y_i and ϵ_i), this inequality is still valid.

Substituting the bound on $\widehat{R}_n(\widehat{f},f_0)$ from Proposition 3.6.3 in (3.6.2), choosing $\varepsilon = 1$ and f as a minimizer over $\mathcal{F}$ of $\mathbb{E}_{\mathbf{X}}\left[(f(\mathbf{X})-f_0(\mathbf{X}))^2\right]$, the fact that $h_n \leq 2(\log(n)/n)^{1/d}$, and replacing the explicit constants by C_1, C_2, C_3 yields the result. □

3.6.2 Proof of Theorem 3.3.4

The following lemma provides a bound on the covering entropy.

Lemma 3.6.4 (Lemma 5 combined with Remark 1 of [127]). *For any $\delta > 0$*

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

The proof of Theorem 1 in [127] derives the following bound for the approximation error for function approximation in the function class $\mathcal{G}(q,\mathbf{d},\mathbf{t},\boldsymbol{\alpha},Q')$ by sparsely connected deep ReLU networks.

Theorem 3.6.5. *For every function $g \in \mathcal{G}(q,\mathbf{d},\mathbf{t},\boldsymbol{\alpha},Q')$ and whenever*

(i) $\sum_{i=1}^q \log_2(4t_i \vee 4\alpha_i)\log_2(n) \leq L \lesssim n\phi_n$,

(ii) $n\phi_n \lesssim \min_{i=1,\dots,L} p_i$,

(iii) $s \asymp n\phi_n \log(n)$,

(iv) $F \geq \max\{Q', 1\}$,

then there exists a neural network $H \in \mathcal{F}(L,\mathbf{p},s,F)$ and a constant C_8 only depending on $q,\mathbf{d},\mathbf{t},\boldsymbol{\alpha},F$ and the implicit constants in (i), (ii) and (iii), such that

$$\|g-H\|_\infty^2 \leq C_8\phi_n.$$

We now have all the necessary ingredients to prove Theorem 3.3.4

Proof of Theorem 3.3.4. Apply the general oracle inequality in Theorem 3.3.1 with the choice $\delta = n^{-1}$ to the neural network class $\mathcal{F}(L, \mathbf{p}, s, F)$ with parameter constraints as in the statement of the theorem. For the approximation error in the oracle inequality, we use Theorem 3.6.5.For the covering entropy, the bound from Lemma 3.6.4 gives $\log\left(\mathcal{N}_{\mathcal{F}(L,\mathbf{p},s,\infty)}(\delta)\right) \lesssim (s+1)L\log(n) \asymp nL\phi_n \log^2(n)$. Since $L \gtrsim \log(n)$, we have $(\log(n)/n)^{2\beta/d} \leq L(n^{-2\beta/d} \vee n^{-1})$. As $\phi_n \gg n^{-1}$, $L\phi_n \log^4(n) + (\log(n)/n)^{2\beta/d} \leq L\max(\phi_n \log^4(n), n^{-2\beta/d})$. Combined with the assumption that $\Delta_n(\widehat{f}_n, f_0) \leq C_4 L \max(\phi_n \log^4(n), n^{-2\beta/d})$, Theorem 3.3.1 yields

$$\begin{aligned} R(\widehat{f}_n, f_0) \leq & C_1 \frac{\log^2(n)\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n} + C_2\delta + C_3\left(\frac{\log(n)}{n}\right)^{\frac{2\beta}{d}} + \frac{16}{3}\Delta_n(\widehat{f}_n, f_0) \\ & + \frac{20}{3}\inf_{f\in\mathcal{F}} \mathbb{E}_{\mathbf{X}}\left[(f(\mathbf{X}) - f_0(\mathbf{X}))^2\right] \\ \lesssim & L\max\left(\phi_n \log^4(n), n^{-\frac{2\beta}{d}}\right). \end{aligned}$$

This yields the result. □

3.7 Proofs for Section 3.4

Lemma 3.7.1. *Let m be a positive integer and $Q > 0$. Then $f : [-Q,Q]^m \to \mathbb{R}$: $f(\mathbf{x}) := \prod_{i=1}^m x_i$ is in $C_m^\gamma([-Q,Q]^m, (Q+1)^m)$, for all $\gamma \geq m+1$.*

Proof. Observe that $|\partial^0 f(\mathbf{x})| = |f(\mathbf{x})| \leq Q^m$, $\partial_{x_j} f(\mathbf{x}) = \prod_{i=1, i\neq j}^m x_i$ and $\partial_{x_j}\partial_{x_j} f = 0$, for $i = 1, \ldots, m$. This means that for all $\boldsymbol{\alpha} \in \mathbb{Z}_{\geq 0}^m$ it holds that $\partial^{\boldsymbol{\alpha}} f = 0$ if $\alpha_j \geq 2$ for some $j \in \{1, \ldots, m\}$. Rephrased, $\partial^{\boldsymbol{\alpha}} f \neq 0$ if and only if $\boldsymbol{\alpha} \in \{0,1\}^m$. Furthermore for $\boldsymbol{\alpha} \in \{0,1\}^m$, $|\partial^{\boldsymbol{\alpha}} f(\mathbf{x})| = |\prod_{i:\alpha_i=0} x_i| \leq Q^{m-|\boldsymbol{\alpha}|_0}$, where $|\cdot|_0$ denotes the counting norm. There are $\binom{m}{m-|\boldsymbol{\alpha}|_0}$ ways to distribute $m - |\boldsymbol{\alpha}|_0$ zeros over a vector of length m. So for $\gamma \geq m+1$, we get by the binomial theorem

$$\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1<\gamma} \|\partial^{\boldsymbol{\alpha}} f\|_\infty \leq \sum_{k=0}^m \binom{m}{k} Q^k = (Q+1)^m.$$

If $|\boldsymbol{\alpha}|_1 > m$, then there exists at least one j such that $\alpha_j \geq 2$ and thus we have that $\partial^{\boldsymbol{\alpha}} f = 0$ in this case. In the case that $|\boldsymbol{\alpha}|_1 = m$, then either there exists a j such that $\alpha_j \geq 2$, so $\partial^{\boldsymbol{\alpha}} f = 0$, or $\boldsymbol{\alpha}$ is the vector with only ones, in which case $\partial^{\boldsymbol{\alpha}} f = 1$. Hence, $\gamma \geq m+1$ yields

$$\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1=\lfloor\gamma\rfloor} \sup_{\mathbf{x},\mathbf{y}\in\mathcal{D},\mathbf{x}\neq\mathbf{y}} \frac{|\partial^{\boldsymbol{\alpha}} f(\mathbf{x}) - \partial^{\boldsymbol{\alpha}} f(\mathbf{y})|}{|\mathbf{x}-\mathbf{y}|_\infty^{\gamma-\lfloor\gamma\rfloor}} = 0.$$

Together with the definition the Hölder ball, (3.3.1), the statement follows. □

Proof of Lemma 3.4.1. The function $g_0 = (g_{0,1}, \ldots, g_{0,|\mathcal{R}|})$ is given by $g_{0,I} = \psi_I$ for all $I \in \mathcal{R}$. From $\psi_I \in C_{r_I}^{\gamma}([0,1]^{r_I}, Q)$ and $|I| \leq r$ it follows that $t_0 = r$ and $\alpha_0 = \gamma$. The function $g_1(u_1, \ldots, u_{|\mathcal{R}|}) = \prod_{I \in \mathcal{R}} u_I$ is the product of $|\mathcal{R}|$ different terms in $[-Q, Q]$. Applying Lemma 3.7.1 yields $g_1 \in \mathcal{C}_{|\mathcal{R}|}^{\zeta}([-Q, Q]^{|\mathcal{R}|}, (Q+1)^{|\mathcal{R}|})$ for all $\zeta \geq |\mathcal{R}| + 1$. So $t_1 = |\mathcal{R}|$ and α_1 is arbitrarily large. □

Proof of Lemma 3.4.2. We argue by contradiction. Let j be the index of the marginal density that is at most α-Hölder smooth. Denote the Hölder smoothness of f by β and suppose that $\beta > \alpha$. Then there exists a constant Q such that $f \in C_d^{\beta}([0,1]^d, Q)$. Because f is β-Hölder smooth it holds that

$$\frac{\partial^k}{\partial x_j^k} f(\mathbf{x}) = \left(\prod_{i \neq j} f_i(x_i)\right) \frac{\partial^k}{\partial x_j^k} f_j(x_j) \tag{3.7.1}$$

for all $k = 0, 1, \ldots, \lfloor \beta \rfloor$. Since $\prod_{i \neq j} f_i(x_i)$ is a density on $[0,1]^{d-1}$, it is nonnegative and there exists a $\widetilde{\mathbf{x}}_{-j} = (\widetilde{x}_1, \ldots, \widetilde{x}_{j-1}, \widetilde{x}_{j+1}, \ldots, \widetilde{x}_d) \in [0,1]^{d-1}$ such that $\prod_{i \neq j} f_i(\widetilde{x}_i) = C > 0$. Since f_j only depends on x_j and f is β-Hölder smooth, for any $k = 0, 1, , \ldots, \lfloor \beta \rfloor$,

$$\begin{aligned}\frac{Q}{C} \geq \frac{1}{C} \left\| \left(\prod_{i \neq j} f_i(x_i)\right) \frac{\partial^k}{\partial x_j^k} f_j \right\|_{L^\infty([0,1]^d)} &\geq \frac{\prod_{i \neq j} f_i(\widetilde{x}_i)}{C} \left\| \frac{\partial^k}{\partial x_j^k} f_j \right\|_{L^\infty([0,1])} \\ &= \left\| \frac{\partial^k}{\partial x_j^k} f_j \right\|_{L^\infty([0,1])}.\end{aligned} \tag{3.7.2}$$

Similarly, by the β-Hölder smoothness of f and (3.7.1),

$$\begin{aligned}\frac{Q}{C} &\geq \frac{1}{C} \sup_{\mathbf{x}, \mathbf{y} \in [0,1]^d, \mathbf{x} \neq \mathbf{y}} \frac{\left| \frac{\partial^{\lfloor \beta \rfloor}}{\partial x_j^{\lfloor \beta \rfloor}} f(\mathbf{x}) - \frac{\partial^{\lfloor \beta \rfloor}}{\partial x_j^{\lfloor \beta \rfloor}} f(\mathbf{y}) \right|}{|\mathbf{x} - \mathbf{y}|_\infty^{\beta - \lfloor \beta \rfloor}} \\ &\geq \frac{\prod_{i \neq j} f_i(\widetilde{x}_i)}{C} \sup_{x, y \in [0,1], x \neq y} \frac{\left| \frac{\partial^{\lfloor \beta \rfloor}}{\partial x_j^{\lfloor \beta \rfloor}} f_j(x) - \frac{\partial^{\lfloor \beta \rfloor}}{\partial x_j^{\lfloor \beta \rfloor}} f_j(y) \right|}{|x - y|^{\beta - \lfloor \beta \rfloor}}.\end{aligned} \tag{3.7.3}$$

From (3.7.2) and (3.7.3) it follows that $f_j \in C_1^{\beta}([0,1], (\lfloor \beta \rfloor + 1)Q/C)$. Since $\beta > \alpha$, this contradicts the condition that f_j was at most α-Hölder smooth. Therefore, $\beta \leq \alpha$. □

Proof of Lemma 3.4.3. The function $g_0 = (g_{0,1}, \dots, g_{0,2d})$ is given by $g_{0,i}(x_i) = f_i(x_i)$ for $i = 1, \dots, d$ and $g_{0,i}(x_{i-d}) = F_{i-d}(x_{i-d})$ for $i = d+1, \dots, 2d$. Each of these functions is univariate, so $t_0 = 1$. Since F_{i-d} is the c.d.f. of f_{i-d}, it holds that $F_{i-d} \in C_1^{\gamma_i+1}([0,1], Q_i+1)$. Thus the function $g_{0,i}$ with the smallest Hölder smoothness has to correspond to one of the functions f_i and $\alpha_0 = \min_{i=1,\dots,d} \gamma_i$. The function $g_1 = (g_{1,1}, \dots, g_{1,d+1})$ satisfies $g_{1,i}(y_i) = y_i$ (the identity function) for $i = 1, \dots, d$ and $g_{1,d+1}(\mathbf{v}) = c(v_1, \dots, v_d)$, so $t_1 = d$. For $i = 1, \dots, d$ the domain of $g_{1,i}$ is $[0, \|f_i\|_\infty] \subseteq [0, Q_i]$, so $g_{1,i} \in C_1^\gamma([0, Q_i], Q_i + 1)$, for all $\gamma \geq 2$. This means the Hölder smoothness of $g_{1,i}$ can be taken arbitrarily large, that consequently $g_{1,d+1}$, corresponding to the copula c, has the smallest Hölder smoothness among the component functions of g_1, and thus $\alpha_1 = \gamma_c$. Set $Q := Q_c \vee (\max_{i=1,\dots,d} Q_i)$, then $g_2(u, y_1, \dots, y_d) = u \prod_{i=1}^d y_i$ is the product of $d+1$ different factors in $[-Q, Q]^{d+1}$. Applying Lemma 3.7.1 yields $g_2 \in C_{d+1}^\gamma([-Q, Q]^{d+1}, (Q+1)^{d+1})$ for all $\gamma \geq d+2$. So $t_2 = d+1$ and the smoothness index α_2 can be taken to be arbitrarily large. □

Proof of Lemma 3.4.4. The function $g_0 = (g_{0,1}, \dots, g_{0,2d})$ is given by $g_{0,i}(x_i) = f_i(x_i)$ for $i = 1, \dots, d$ and $g_{0,i}(x_{i-d}) = F_{i-d}(x_{i-d})$ for $i = d+1, \dots, 2d$. Since F_{i-d} is the c.d.f. of f_{i-d}, it holds that $F_{i-d} \in C_1^{\gamma_{m,i}+1}([0,1], Q_{m,i} + 1)$. So $t_0 = 1$ and $\alpha_0 = \min_{i=1,\dots,d} \gamma_{m,i}$. The function $g_1 = (g_{1,1}, \dots, g_{1,d+(d-1)})$ satisfies $g_{1,i}(u_i) = u_i$ (the identity function) for $1 = 1, \dots, d$. For f of the form (3.4.8) it holds that $g_{1,i}(v_1, v_2) = c_{1,i+1-d}(v_1, v_2)$ for $i = d+1, \dots, d+(d-1)$ and for f of the form (3.4.9) we have that $g_{1,i}(v_1, v_2) = c_{i-d,i+1-d}(v_1, v_2)$ for $i = d+1, \dots, d+(d-1)$. For $i = 1, \dots, d$ the domain of $g_{1,i}$ is $[0, \|f_i\|_\infty] \subseteq [0, Q_{m,i}]$, so for $i = 1, \dots, d$ it holds that $g_{1,i} \in C_1^\gamma([0, Q_{m,i}], Q_{m,i} + 1)$, for all $\gamma \geq 2$. This means the Hölder smoothness of $g_{1,i}$, for $i = 1, \dots, d$, can be taken to be arbitrarily large. So $t_1 = 2$ and $\alpha_2 = \gamma_c$. Set $Q = (\max_{i=1,\dots,d} Q_{m,i}) \vee Q_c$. The function $g_2(u_1, \dots, u_d, y_1, \dots, y_{d-1}) = \prod_{k=1}^d u_k \prod_{j=1}^{d-1} y_j$ is the product of $2d-1$ terms. Thus by Lemma 3.7.1 it holds that $g_2 \in C_{2d-1}^\gamma([-Q, Q]^{2d-1}, (Q+1)^{2d-1})$ for all $\gamma \geq 2d$, so α_2 is arbitrarily large and $t_2 = 2d-1$. □

3.7.1 Proof of Theorem 3.4.5

Recall that we work in the density estimation model as defined in Section 3.2 with mixture density $f_0 = \sum_{j=1}^r a_j f_j$, where $a_1, \dots a_r$ are non-negative mixture weights summing up to one, and $f_j \in \mathcal{C}_d^{\beta_j}([0,1]^d, Q) \cap \mathcal{G}(q_j, \mathbf{d}_j, \mathbf{t}_j, \boldsymbol{\alpha}_j, Q)$, for $j = 1, \dots, r$. Set $\phi_n^\star = \max_{j=1,\dots,r} \phi_{n,j}$, where $\phi_{n,j}$ is the rate (3.3.4) for estimation of f_j and set $\beta = \min_{j=1,\dots,r} \beta_j$.

Lemma 3.7.2 (Approximation of mixtures). *Whenever*

(i) $\max_{j=1,\dots,r} \sum_{i=1}^{q_j} \log_2(4t_{i,j} \vee 4\alpha_{i,j}) \log_2(n) \leq L \lesssim n\phi_n^\star$,

(ii) $n\phi_n^\star \lesssim \min_{i=1,\ldots,L} p_i$,
(iii) $s \asymp n\phi_n^\star \log(n)$,
(iv) $F \geq \max\{Q, 1\}$,
then, for n large enough, there exists a network $H \in \mathcal{F}(L, \mathbf{p}, s, F)$ and a constant C_9 only depending on $(q_j, \mathbf{d}_j, \mathbf{t}_j, \boldsymbol{\alpha}_j)_{j=1}^r$, r, F and the implicit constants in (i), (ii) and (iii) such that

$$\left\| \sum_{j=1}^r a_j f_j - H \right\|_\infty^2 \leq C_9 \phi_n^\star.$$

Proof. For positive constants $c_L, c_p, c_{s\ell}, c_{su}$, let $L^\star$, $\mathbf{p}^\star$ and $s^\star$ be such that
(i') $\max_{j=1,\ldots,r} \sum_{i=1}^{q_j} \log_2(4t_{i,j} \vee 4\alpha_{i,j}) \log_2(n) \leq L^\star \leq c_L n\phi_n^\star$
(ii') $n\phi_n^\star \leq c_p \min_{i=1,\ldots,L} p_i^\star$
(iii') $c_{s\ell} n\phi_n^\star \log(n) \leq s^\star \leq c_{su} n\phi_n^\star \log(n)$.
For n large enough, we have
(I) $c_L n\phi_n^\star \leq (c_{s\ell}/(2r)) n\phi_n^\star \log(n)$,
(II) $n\phi_n^\star > rc_p$,
(III) $\lfloor c_L n\phi_{n,j} \rfloor \geq \sum_{i=1}^{q_j} \log_2(4t_{i,j} \vee 4\alpha_{i,j}) \log_2(n)$, for all $j = 1, \ldots, r$,
(IV) $(c_{sl}/(4r)) n\phi_{n,j} \log(n) \geq 1$, for all $j = 1, \ldots, r$.

For $j = 1, \ldots, r$ define $L_j := \min\{L^\star, \lfloor c_L n\phi_{n,j} \rfloor\}$, $p_{i,j} = \lfloor p_i^\star / r \rfloor$ and $s_j = \lfloor s^\star \phi_{n,j} / (2r\phi_n^\star) \rfloor$. Recall that $\phi_n^\star = \max_{j=1,\ldots,r} \phi_{n,j}$. Using the definition of L_j and (III) yields

$$\sum_{i=1}^{q_j} \log_2(4t_{i,j} \vee 4\alpha_{i,j}) \log_2(n) \leq L_j \leq c_L n\phi_{n,j}.$$

Using (ii'), (II), and the definitions of ϕ_n^* and $\mathbf{p}_j$ we get that

$$n\phi_{n,j} \leq 2c_p r \min_{i=1,\ldots,L} \lfloor p_i^* / r \rfloor = 2c_p r \min_{i=1,\ldots,L} p_{i,j}.$$

From (IV), the definition $s_j = \lfloor s^\star \phi_{n,j} / (2r\phi_n^\star) \rfloor$, (iii'), and $\lfloor u \rfloor \geq u - 1$ for all $u \in \mathbb{R}$, it follows that

$$\frac{c_{sl}}{4r} n\phi_{n,j} \log(n) \leq \frac{c_{sl}}{2r} n\phi_{n,j} \log(n) - 1 \leq s_j \leq \frac{c_{su}}{2r} n\phi_{n,j} \log(n).$$

This means that for $j = 1, \ldots, r$ the class $\mathcal{F}(L_j, \mathbf{p}_j, s_j, F)$ and the function $f_j \in \mathcal{C}_d^{\beta_j}([0,1]^d, Q) \cap \mathcal{G}(q_j, \mathbf{d}_j, \mathbf{t}_j, \boldsymbol{\alpha}_j, Q)$ satisfy the conditions of Theorem 3.6.5. Applying Theorem 3.6.5 gives us that for each $j = 1 \ldots, r$ there exist a network $H_j \in \mathcal{F}(L_j, \mathbf{p}_j, s_j, F)$ such that $\|f_j - H_j\|_\infty^2 \leq C_{8,j} \phi_{n,j}$. Since a_j is in $[0,1]$, multiplying the last weight matrix of H_j with a_j yields a network $a_j H_j$ in the same network class as H_j such that $\|a_j f_j - a_j H_j\|_\infty^2 \leq C_{8,j} \phi_{n,j}$.

Whenever $L_j < L^\star$, we can synchronize the depth by adding additional layers with identity weight matrix such that

$$\mathcal{F}_j(L_j, \mathbf{p}_j, s_j, F) \subset \widetilde{\mathcal{F}}_j(L^*, (\mathbf{p}_j, \underbrace{1, \dots, 1}_{(L^\star - L_j)\text{ times}}), s_j + (L^\star - L_j), F).$$

For ease of notation define $\widetilde{\mathbf{p}}_j = (\mathbf{p}_j, 1, \dots, 1)$. Placing all these networks in parallel yields a network

$$H \in \mathcal{F}\Big(L^\star, \sum_{j=1}^r \widetilde{\mathbf{p}}_j, \sum_{j=1}^r \big(s_j + (L^\star - L_j)\big), F\Big),$$

such that

$$\begin{aligned}\left\| \sum_{j=1}^r a_j f_j - H \right\|_\infty^2 \leq \left(\sum_{j=1}^r \|a_j f_j - a_j H_j\|_\infty \right)^2 &\leq \left(\sum_{j=1}^r \sqrt{C_{8,j} \phi_{n,j}} \right)^2 \\ &\leq r^2 \max_{j=1,\dots,r} C_{8,j} \phi_{n,j}.\end{aligned}$$

A network with width $\mathbf{p}$ and sparsity s can always be embedded in a larger network of the same depth with width $\widetilde{\mathbf{p}} \geq \mathbf{p}$ and network sparsity $\widetilde{s} \geq s$. Thus it remains to show that $\sum_{j=1}^r \widetilde{\mathbf{p}}_j \leq \mathbf{p}^\star$ and $\sum_{j=1}^r \big(s_j + (L^\star - L_j)\big) \leq s^\star$. First consider the width. Using the definitions of $p_{i,j}$ and $\widetilde{\mathbf{p}}_j$ we get for $i = 1, \dots, L^\star$ that $\sum_{j=1}^r \widetilde{p}_{i,j} \leq r \max_{j=1,\dots,r} \widetilde{p}_{i,j} \leq r \max\{p_i^\star / r, 1\}$. From (II) and (ii') we get that $p_i^\star / r > 1$. Hence, $\sum_{j=1}^r \widetilde{\mathbf{p}}_j \leq \mathbf{p}^\star$. Now consider the sparsity. By the definition of s_j it holds that $s_j \leq s^\star/(2r)$. From (i') and (I) we get that $L^\star \leq s^\star/(2r)$. Hence, $\sum_{j=1}^r \big(s_j + (L^\star - L_j)\big) \leq \sum_{j=1}^r \big(s_j + L^\star\big) \leq s^\star$. □

Proof of Theorem 3.4.5. The derivative of a sum is the sum of the derivatives. Since $f_j \in \mathcal{C}_d^{\beta_j}([0,1]^d, Q)$ for $j = 1, \dots, r$, this means that f_0 has smoothness at least $\beta = \min_{j=1,\dots,r} \beta_j$. Furthermore $(a_1, \dots, a_r)$ are non-negative mixture weights that sum op to one and $f_0 \in \mathcal{C}_d^\beta([0,1]^d, Q)$. The statement of the theorem now follows from taking $\delta = 1/n$ and the network class $\mathcal{F}(L, \mathbf{p}, s, F)$ as the function class in Theorem 3.3.1. For the approximation error in the oracle inequality, we use Lemma 3.7.2 and for the covering entropy the bound from Lemma 3.6.4. This yields the result. □

3.8 Proofs for Section 3.5

Proof of Lemma 3.5.1. The function $g_0 = (g_{0,1}, \dots, g_{0,3})$ is given by $g_{0,1}(x_i) = x_i$, $g_{0,2}(x_k) = x_k$ and $g_{0,3}(x_k) = 1 - x_k$. Since $x_i, x_k \in [0,1]$, it holds that $g_{0,i} \in$

$C_1^\gamma([0,1],2)$, for all $\gamma \geq 2$. The function $g_1 = (g_{1,1}, \dots, g_{1,3})$ is given by $g_{1,1}(x_i) = x_i$, $g_{1,2}(u) = h_j(u)$ and $g_{1,3}(v) = h_j(v)$, so $t_1 = 1$. Since $g_{1,1} \in C_1^\gamma([0,1],2)$, for all $\gamma \geq 2$, we get that $\alpha_1 = \gamma_j$. The function g_2 is given by $g_2(x_i, y_1, y_2) = x_i y_1 + (1-x_i) y_2$, so $t_2 = 3$. The partial derivatives are $\partial_{x_i} g_2 = y_1 - y_2$, $\partial_{y_1} g_2 = x_i$, $\partial_{y_2} g_2 = 1 - x_i$, $\partial_{x_i}\partial_{y_1} = 1$ and $\partial_{x_i}\partial_{y_2} = -1$. All other partial derivatives of g_2 vanish. Thus $g_2 \in C_3^\gamma([0,1] \times [-Q,Q]^2, 4(Q+1))$, for all $\gamma \geq 3$, so α_2 is arbitrarily large. □

Proof of Lemma 3.5.2. The function g_0 is given by $g_0(x_j, x_i) = \max\{0, x_j - x_i/4\}$. The derivative of this function is discontinuous along the line $x_j - x_i/4 = 0$. Observe that $|\max(0,u) - \max(0,u+v)| \leq |v|$, for all real numbers u, v. Hence

$$\frac{|g_0(x_j,x_i) - g_0(x_j + h_j, x_i + h_i)|}{\max(|h_j|,|h_i|)} \leq \frac{|h_i/4 - h_j|}{\max(|h_j|,|h_i|)} \leq \frac{5}{4}.$$

Thus $g_0 \in C_2^1([0,1]^2, 9/4)$, so $\alpha_0 = 1$. The function g_1 is given by $g_1(y) = h_j(y)$, thus $t_1 = 1$ and $\alpha_1 = \gamma_j$. □

3.9 Proofs for Section 3.6

Proof of Lemma 3.6.2. The random variable $\epsilon_i = \widehat{f}_{\mathrm{KDE}}(\mathbf{X}_i) - f_0(\mathbf{X}_i)$ is not centered. The first step adds and subtracts $\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]$ to get the centered random variable $\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]$ instead. Together with the triangle inequality, this gives

$$\begin{aligned}
&\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n \epsilon_i(\widehat{f}(\mathbf{X}_i) - f(\mathbf{X}_i))\right]\right| \\
&\leq \left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n (\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i])(\widehat{f}(\mathbf{X}_i) - f_0(\mathbf{X}_i))\right]\right| \\
&+ \left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n (\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i])(f_0(\mathbf{X}_i) - f(\mathbf{X}_i))\right]\right| \qquad (3.9.1)\\
&+ \left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i](\widehat{f}(\mathbf{X}_i) - f(\mathbf{X}_i))\right]\right| \\
&=: (I) + (II) + (III).
\end{aligned}$$

By the tower rule, we can in (II) first condition the expectation on $\mathbf{X}_i$. Now $(II) = 0$ follows from

$$\begin{aligned}
&\mathbb{E}_{f_0}\Big[\left(\epsilon_i - \mathbb{E}_{f_0}\left[\epsilon_i|\mathbf{X}_i\right]\right)\left(f_0(\mathbf{X}_i) - f(\mathbf{X}_i)\right)\Big|\mathbf{X}_i\Big] \\
&\quad = \big(\mathbb{E}_{f_0}\left[\epsilon_i|\mathbf{X}_i\right] - \mathbb{E}_{f_0}\left[\epsilon_i|\mathbf{X}_i\right]\big)\big(f_0(\mathbf{X}_i) - f(\mathbf{X}_i)\big) = 0.
\end{aligned}$$

For real numbers a_i, b_i, we have $(|a_i| - |b_i|/2)^2 \geq 0$ and therefore $|a_i b_i| \leq a_i^2 + b_i^2/4$ as well as $\sum_i |a_i b_i| \leq \sum_i a_i^2 + \frac{1}{4}\sum_i b_i^2$. Bringing first the absolute value inside the expectation and applying this inequality twice, once to the sequences $(2\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]/\sqrt{n})_i$ and $((\widehat{f}(\mathbf{X}_i) - f_0(\mathbf{X}_i))/\sqrt{n})_i$ and once to the sequences $(2\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]/\sqrt{n})_i$ and $((f_0(\mathbf{X}_i) - f(\mathbf{X}_i))/\sqrt{n})_i$ yields

$$\begin{aligned}
&\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i](\widehat{f}(\mathbf{X}_i) - f(\mathbf{X}_i))\right]\right| \\
&\overset{(i)}{=} \left|\mathbb{E}_{f_0}\left[\sum_{i=1}^n \frac{2\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]}{\sqrt{n}} \frac{(\widehat{f}(\mathbf{X}_i) - f_0(\mathbf{X}_i))}{\sqrt{n}}\right]\right. \\
&\quad \left. + \mathbb{E}_{f_0}\left[\sum_{i=1}^n \frac{2\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]}{\sqrt{n}} \frac{(f_0(\mathbf{X}_i) - f(\mathbf{X}_i))}{\sqrt{n}}\right]\right| \\
&\leq 8\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n (\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i])^2\right] + \frac{1}{4}\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n (f_0(\mathbf{X}_i) - f(\mathbf{X}_i))^2\right] \\
&\quad + \frac{1}{4}\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n \Big(\widehat{f}(\mathbf{X}_i) - f_0(\mathbf{X}_i)\Big)^2\right] \\
&\overset{(ii)}{=} 8\mathbb{E}_{f_0}\Big[(\mathbb{E}_{f_0}[\epsilon_1|\mathbf{X}_1])^2\Big] + \frac{\mathbb{E}_{\mathbf{X}}[(f_0(\mathbf{X}) - f(\mathbf{X}))^2]}{4} + \frac{\widehat{R}_n(\widehat{f}, f_0)}{4},
\end{aligned}$$

where for (i) we added and subtracted the same term and (ii) follows from the definition of $\widehat{R}_n(\widehat{f}, f_0)$ and the fact that the $\mathbf{X}_i$ are i.i.d. Proposition 3.9.1 gives $\mathbb{E}_{f_0}[(\mathbb{E}_{f_0}[\epsilon_1|\mathbf{X}_1])^2] \leq h_n^{2\beta} F^2 d^{2\beta} \|K\|_\infty^{2d}$ and so

$$(III) \leq 8h_n^{2\beta} F^2 d^{2\beta} \|K\|_\infty^{2d} + \frac{\mathbb{E}_{\mathbf{X}}[(f_0(\mathbf{X}) - f(\mathbf{X}))^2]}{4} + \frac{\widehat{R}_n(\widehat{f}, f_0)}{4}. \qquad (3.9.2)$$

It remains to bound (I) in (3.9.1). Let $N := \mathcal{N}_{\mathcal{F}}(\delta)$ be the covering number. By assumption, the N centers $f_1, \ldots, f_N$ lie in $\mathcal{F}$. Choose $k^* \in \{1, \ldots, N\}$ such that

$$\|\widehat{f} - f_{k^*}\|_\infty = \min_{1 \leq \ell \leq N} \|\widehat{f} - f_\ell\|_\infty.$$

In particular, k^* is random. Define $(IV) := |\mathbb{E}_{f_0}[\frac{2}{n}\sum_{i=1}^n(\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i])(f_{k^*}(\mathbf{X}_i) - f_0(\mathbf{X}_i))]|$. This gives us that

$$\begin{aligned}
&\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n(\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i])(\widehat{f}(\mathbf{X}_i) - f_0(\mathbf{X}_i))\right]\right| \\
&\leq \left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n(\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i])(\widehat{f}(\mathbf{X}_i) - f_{k^*}(\mathbf{X}_i))\right]\right| + (IV) \\
&\overset{(i)}{\leq} \mathbb{E}_{f_0}\left[\frac{2\delta}{n}\sum_{i=1}^n\left|\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]\right|\right] + (IV) \\
&\overset{(ii)}{\leq} 4\delta\|K\|_\infty^d 2^d F + (IV)
\end{aligned} \tag{3.9.3}$$

where for (i) we used the property of the δ cover and the triangle inequality, and for (ii) we used Proposition 3.9.2.

In the next step we split the term (IV) into two parts. One case were the $\mathbf{X}_i$ used for the regression are distributed 'nicely' and a second case where we have an extreme concentration of data points $\mathbf{X}_i$. The bad second case can be shown to have small probability. For the derivation, we use the bins $\mathcal{B}_j$ as defined in Section 3.6.

Define the set A_j as $A_j := \{\sum_{i=1}^n \mathbb{1}_{\{\mathbf{X}_i\in\mathcal{B}_j\}} \leq 2^{d+3}F\log(n)\}$ and the set A as the intersection

$$A := \bigcap_{j\in\mathcal{J}} A_j. \tag{3.9.4}$$

By choice of h_n, $2^d\log(n) \geq nh_n^d$. Together with the union bound, it follows that

$$\begin{aligned}
\mathbb{P}_{f_0}(A^c) &\leq \sum_{j\in\mathcal{J}}\mathbb{P}_{f_0}(A_j^c) \\
&= \sum_{j\in\mathcal{J}}\mathbb{P}_{f_0}\left(\sum_{i=1}^n \mathbb{1}_{\{\mathbf{X}_i\in\mathcal{B}_j\}} > (7F+F)nh_n^d\right) \\
&\overset{(iii)}{\leq} \sum_{j\in\mathcal{J}}\mathbb{P}_{f_0}\left(\sum_{i=1}^n \mathbb{1}_{\{\mathbf{X}_i\in\mathcal{B}_j\}} > 7Fnh_n^d + np_j\right) \\
&= \sum_{j\in\mathcal{J}}\mathbb{P}_{f_0}\left(\sum_{i=1}^n \left(\mathbb{1}_{\{\mathbf{X}_i\in\mathcal{B}_j\}} - p_j\right) > 7Fnh_n^d\right) \\
&\leq \sum_{j\in\mathcal{J}}\mathbb{P}_{f_0}\left(\left|\sum_{i=1}^n \left(\mathbb{1}_{\{\mathbf{X}_i\in\mathcal{B}_j\}} - p_j\right)\right| > 7Fnh_n^d\right),
\end{aligned} \tag{3.9.5}$$

where for (iii) we used that $p_j = \int_{\mathcal{B}_j} f_0(\mathbf{x})\, d\mathbf{x} \leq F h_n^d$ is the probability that an observation falls into bin $\mathcal{B}_j$.

We now apply the moment-version of Bernstein's inequality stated in Proposition 3.9.4 (i). For any $m = 1, \ldots$

$$\mathbb{E}_{f_0}\left[|\mathbb{1}_{\{\mathbf{X}_i \in \mathcal{B}_j\}}|^m\right] = \mathbb{E}_{f_0}\left[\mathbb{1}_{\{\mathbf{X}_i \in \mathcal{B}_j\}}\right] = p_j.$$

Setting $U = 1$, $v_i = p_j$, and $v = nFh_n^d \geq np_j$, we get from Bernstein's inequality in Proposition 3.9.4 (i) that

$$\begin{aligned}\mathbb{P}_{f_0}\left(\left|\sum_{i=1}^n \left(\mathbb{1}_{\{\mathbf{X}_i \in \mathcal{B}_j\}} - p_j\right)\right| > 7Fnh_n^d\right) &\leq 2\exp\left(-\frac{7^2F^2n^2h_n^{2d}}{2n(Fh_n^d + 7Fh_n^d)}\right) \\ &= 2\exp\left(-\frac{49}{16}Fnh_n^d\right) \\ &\leq 2\exp(-3nh_n^d) \\ &\overset{(v)}{\leq} 2n^{-3},\end{aligned}$$

where for (v) we used that, by choice of h_n, $h_n^d \geq \log(n)/n$. Combined with (3.9.5), we find

$$\mathbb{P}_{f_0}(A^c) \leq 2\sum_{j \in \mathcal{J}} n^{-3} \leq 2n^{-2},$$

where the last inequality holds because $n > e$ implies $|\mathcal{J}| = h_n^{-d} \leq n/\log n \leq n$.

With

$$\xi_k := \sum_{i=1}^n \left(\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]\right)\left(f_k(\mathbf{X}_i) - f_0(\mathbf{X}_i)\right)\mathbb{1}_A$$

one can decompose (IV) as follows

$$\begin{aligned}&\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n \left(\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]\right)\left(f_{k^*}(\mathbf{X}_i) - f_0(\mathbf{X}_i)\right)\right]\right| \\ &\leq \left|\mathbb{E}_{f_0}\left[\frac{2}{n}\xi_{k^*}\right]\right| + \left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n \left(\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]\right)\left(f_{k^*}(\mathbf{X}_i) - f_0(\mathbf{X}_i)\right)\mathbb{1}_{A^c}\right]\right|.\end{aligned} \tag{3.9.6}$$

Moving the absolute value inside, using that f_{k^*} and f_0 are bounded by F and applying the Cauchy-Schwarz inequality yields

$$\begin{aligned}
&\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^{n}(\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i])(f_{k^*}(\mathbf{X}_i) - f_0(\mathbf{X}_i))\mathbb{1}_{A^c}\right]\right| \\
&\leq \frac{4F}{n}\sum_{i=1}^{n}\mathbb{E}_{f_0}\left[|\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]|\,\mathbb{1}_{A^c}\right] \\
&\leq \frac{4F}{n}\sum_{i=1}^{n}\sqrt{\mathbb{E}_{f_0}\left[|\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]|^2\right]}\sqrt{\mathbb{P}_{f_0}(A^c)} \qquad (3.9.7) \\
&\overset{(*)}{\leq} \frac{4F\sqrt{2(65F^2 2^{2d}\|K\|_\infty^{2d})}}{n} \\
&\leq \frac{46F^2 2^d\|K\|_\infty^d}{n}.
\end{aligned}$$

where for $(*)$ we used Proposition 3.9.3 and that $\mathbb{P}_{f_0}(A^c) \leq 2n^{-2}$ and for the last inequality we used that $4\sqrt{130} \leq 46$.

It remains to bound the term $|\mathbb{E}_{f_0}[\frac{2}{n}\xi_{k^*}]|$. Define

$$z_k := \sqrt{\log(\mathcal{N}_{\mathcal{F}}(\delta))} \vee \sqrt{n}\|f_k - f_0\|_n \qquad (3.9.8)$$

and define z_{k^*} as z_k for $k = k^*$. The empirical norm of a function g is

$$\|g\|_n := \left(\frac{1}{n}\sum_{i=1}^{n}(g(\mathbf{X}_i))^2\right)^{\frac{1}{2}}.$$

Using that k^* is the index of the center of the ball of the δ-cover closest to $\widehat{f}$, it holds that

$$\frac{z_{k^*}}{\sqrt{n}} = \sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}} \vee \|f_{k^*} - f_0\|_n \leq \|\widehat{f} - f_0\|_n + \delta + \sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}}.$$

Together with the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}
\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\xi_{k^*}\right]\right| &\leq \frac{2}{\sqrt{n}}\mathbb{E}_{f_0}\left[\left|\frac{\xi_{k^*}}{\sqrt{n}}\right|\right] \\
&\leq \frac{2}{\sqrt{n}}\mathbb{E}_{f_0}\left[\frac{\|\widehat{f} - f_0\|_n + \delta + \sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}}}{\frac{z_{k^*}}{\sqrt{n}}}\left|\frac{\xi_{k^*}}{\sqrt{n}}\right|\right] \qquad (3.9.9) \\
&\leq 2\frac{\sqrt{\widehat{R}_n(\widehat{f}, f_0)} + \delta + \sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}}}{\sqrt{n}}\sqrt{\mathbb{E}_{f_0}\left[\frac{\xi_{k^*}^2}{z_{k^*}^2}\right]}.
\end{aligned}$$

For notational ease, define

$$C_{i,k} := \frac{f_k(\mathbf{X}_i) - f_0(\mathbf{X}_i)}{nh_n^d z_k} \mathbb{1}_A. \tag{3.9.10}$$

Since probabilities are always upper bounded by one, we have for any $a > 0$ and any square integrable random variable T, $\mathbb{E}[T^2] = \int_0^\infty \mathbb{P}\left(T^2 \geq t\right)\, dt = \int_0^\infty \mathbb{P}\left(|T| \geq \sqrt{t}\right)\, dt \leq a + \int_a^\infty \mathbb{P}\left(|T| \geq \sqrt{t}\right)\, dt$. Therefore for any $a > 0$

$$\begin{aligned} \mathbb{E}_{f_0}\left[\xi_{k^*}^2/z_{k^*}^2 \,\middle|\, \mathbf{X}_1, \dots, \mathbf{X}_n\right] &\leq \mathbb{E}_{f_0}\left[\max_k \xi_k^2/z_k^2 \,\middle|\, \mathbf{X}_1, \dots, \mathbf{X}_n\right] \\ &\leq a + \int_a^\infty \mathbb{P}_{f_0}\left(\max_k |\xi_k/z_k| \geq \sqrt{t} \,\middle|\, \mathbf{X}_1, \dots, \mathbf{X}_n\right) dt. \end{aligned} \tag{3.9.11}$$

The ratio ξ_k/z_k can be rewritten as the sum $\sum_{\ell=1}^n h_k(\mathbf{X}'_\ell)$, where

$$h_k(u) = \sum_{i=1}^n \left(\prod_{r=1}^d K\left(\frac{u_r - X_{i,r}}{h_n}\right) - \int_{\mathbb{R}^d} \prod_{r=1}^d K\left(\frac{v_r - X_{i,r}}{h_n}\right) f_0(\mathbf{v})\, d\mathbf{v}\right) C_{i,k}$$

is $\sigma(\mathbf{X}_1, \dots, \mathbf{X}_n)$ measurable. Now let $\widetilde{\mathbf{X}}_1, \widetilde{\mathbf{X}}_2, \dots$ be i.i.d. random variables distributed as $\mathbf{X}$ and independent of the data. Let $\mathcal{M}$ be a Poisson(n) random variable independent of the data and of the $\widetilde{\mathbf{X}}_i$. By the union bound and Poissonization (Lemma 3.6.1),

$$\begin{aligned} &\mathbb{P}_{f_0}\left(\max_k |\xi_k/z_k| \geq \sqrt{t} \,\middle|\, \mathbf{X}_1, \dots, \mathbf{X}_n\right) \\ &\leq \mathcal{N}_{\mathcal{F}}(\delta) \max_k \mathbb{P}_{f_0}\left(|\xi_k/z_k| \geq \sqrt{t} \,\middle|\, \mathbf{X}_1, \dots, \mathbf{X}_n\right) \\ &= \mathcal{N}_{\mathcal{F}}(\delta) \max_k \mathbb{P}_{f_0}\left(\left|\sum_{\ell=1}^n h_k(\mathbf{X}'_\ell)\right| \geq \sqrt{t} \,\middle|\, \mathbf{X}_1, \dots, \mathbf{X}_n\right) \\ &\leq \sqrt{2e\pi n}\, \mathcal{N}_{\mathcal{F}}(\delta) \max_k \mathbb{P}_{f_0}\left(\left|\sum_{\ell=1}^{\mathcal{M}} h_k(\widetilde{\mathbf{X}}_\ell)\right| \geq \sqrt{t} \,\middle|\, \mathbf{X}_1, \dots, \mathbf{X}_n\right). \end{aligned} \tag{3.9.12}$$

With $W(\mathbf{X}_i) := \sum_{\ell=1}^{\mathcal{M}} \prod_{r=1}^d K(\frac{\widetilde{X}_{\ell,r} - X_{i,r}}{h_n})$, we can write

$$\sum_{\ell=1}^{\mathcal{M}} h_k(\widetilde{\mathbf{X}}_\ell) = \sum_{i=1}^n \left(W(\mathbf{X}_i) - \mathbb{E}_{f_0}\left[W(\mathbf{X}_i)|\mathbf{X}_i\right]\right) C_{i,k}. \tag{3.9.13}$$

Next we rewrite the sum over n. For this we use the bins $\mathcal{B}_j$ and the index sets of bins $\mathcal{J}_s$ as defined in Section 3.6.1. Using that the bins are disjoint and that each bin is in exactly one of the 3^d index classes $\mathcal{J}_s$, we have $\sum_{i=1}^n = \sum_{s=1}^{3^d} \sum_{j \in \mathcal{J}_s} \sum_{\mathbf{X}_i \in \mathcal{B}_j}$. Here we use $\sum_{\mathbf{X}_i \in \mathcal{B}_j}$ as shorthand notation for $\sum_{1 \le i \le n, \text{s.t.} \mathbf{X}_i \in \mathcal{B}_j}$. For non-negative random variables $U_1, \dots, U_m$, $\{U_1 + \ldots + U_m \ge \sqrt{t}\} \subseteq \cup_{j=1}^m \{U_j \ge \sqrt{t}/m\}$ and by the union bound $\mathbb{P}(U_1 + \ldots + U_m \ge \sqrt{t}) \le m \cdot \max_{j=1,\dots,m} \mathbb{P}(U_j \ge \sqrt{t}/m)$. Combined with (3.9.13),

$$\mathbb{P}_{f_0}\Big(\Big|\sum_{\ell=1}^{\mathcal{M}} h_k(\widetilde{\mathbf{X}}_\ell)\Big| \ge \sqrt{t} \,\Big|\, \mathbf{X}_1, \dots, \mathbf{X}_n\Big)$$
$$\le 3^d \max_{s=1,\dots,3^d} \mathbb{P}_{f_0}\Big(3^d \Big| \sum_{j \in \mathcal{J}_s} \sum_{\mathbf{X}_i \in \mathcal{B}_j} \big(W(\mathbf{X}_i) - \mathbb{E}_{f_0}\big[W(\mathbf{X}_i)|\mathbf{X}_i\big]\big) C_{i,k}\Big| \ge \sqrt{t} \,\Big|\, \mathbf{X}_1, \dots, \mathbf{X}_n\Big).$$

Thus, (3.9.11), (3.9.12) and the previous display give for any $a > 0$

$$\mathbb{E}_{f_0}\Big[\frac{\xi_{k^*}^2}{z_{k^*}^2} \,\Big|\, \mathbf{X}_1, \dots, \mathbf{X}_n\Big]$$
$$\le a + \int_a^\infty \mathcal{N}_{\mathcal{F}}(\delta) 3^d \sqrt{2e\pi n} \max_k \max_{s=1,\dots,3^d}$$
$$\mathbb{P}_{f_0}\Big(3^d \Big| \sum_{j \in \mathcal{J}_s} \sum_{\mathbf{X}_i \in \mathcal{B}_j} \big(W(\mathbf{X}_i) - \mathbb{E}_{f_0}[W(\mathbf{X}_i)|\mathbf{X}_i]\big) C_{i,k}\Big| \ge \sqrt{t} \,\Big|\, \mathbf{X}_1, \dots, \mathbf{X}_n\Big)\, dt \tag{3.9.14}$$

We will now apply Bernstein's inequality in the form of Proposition 3.9.4 (i) to the random variables $Z_j = \sum_{\mathbf{X}_i \in \mathcal{B}_j} W(\mathbf{X}_i) C_{i,k}$. For that we show first that, conditionally on $\mathbf{X}_1, \dots, \mathbf{X}_n$, the random variables $Z_j, j \in \mathcal{J}_s$ with fixed s are jointly independent. To see this, recall that $W(\mathbf{X}_i) := \sum_{\ell=1}^{\mathcal{M}} \prod_{r=1}^d K(\frac{\widetilde{X}_{\ell,r} - X_{i,r}}{h_n})$. The kernel K has support in $[-1, 1]$. By the definition of the neighborhood $NB(\mathcal{B}_j)$ in (3.6.1), Z_j only depends on the $\widetilde{\mathbf{X}}_1, \dots, \widetilde{\mathbf{X}}_n$ that fall into $NB(\mathcal{B}_j)$, that is, $Z_j = \sum_{\mathbf{X}_i \in \mathcal{B}_j} \sum_{\ell=1}^{\mathcal{M}} \prod_{r=1}^d K(\frac{\widetilde{X}_{\ell,r} - X_{i,r}}{h_n}) C_{i,k} \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}$. The variables $C_{i,k}$ defined in (3.9.10) depend on $\mathbf{X}_1, \dots, \mathbf{X}_n$ but not on $\widetilde{\mathbf{X}}_1, \dots, \widetilde{\mathbf{X}}_n$. Working conditionally on $\mathbf{X}_1, \dots, \mathbf{X}_n$ and interchanging the summations, we can write $Z_j = \sum_{\ell=1}^{\mathcal{M}} g_j(\widetilde{\mathbf{X}}_\ell) \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}$, for suitable real-valued functions $g_1, g_2, \dots$ Since the kernel K has support in $[-1, 1]$, it follows from the definition of $\mathcal{J}_s$ that if two different indices j and $\widetilde{j}$ are both in $\mathcal{J}_s$, then $\{\mathbf{x} : g_j(\mathbf{x}) \neq 0\} \cap \{\mathbf{x} : g_{\widetilde{j}}(\mathbf{x}) \neq 0\} = \varnothing$. Let $\mathcal{B}(\mathbb{R})$ be the Borel σ-algebra on $\mathbb{R}$ and define $g(\mathbf{x}) = \sum_{j \in \mathcal{J}_s} g_j(\mathbf{x}) \mathbb{1}_{\{\mathbf{x} \in NB(\mathcal{B}_j)\}}$. The map $T : [0, 1]^d \times \mathcal{B}(\mathbb{R}) \to [0, 1]$, given by

$$T(\mathbf{x}, B) = \begin{cases} 1, & \text{if } g(\mathbf{x}) \in B, \\ 0, & \text{otherwise,} \end{cases}$$

defines a transition kernel. Since $\widetilde{\mathbf{X}}_\ell$ are i.i.d. and $\mathcal{M} \sim \text{Poisson}(n)$, $\sum_{\ell=1}^{\mathcal{M}} \delta_{\widetilde{\mathbf{X}}_\ell}$, with δ_u the point measure at u, is a Poisson point process on $[0,1]^d$. The marking theorem states that a Poisson process on a space A and a transition kernel to the Borel algebra of another space B induces a Poisson point process on the product space $A \times B$, see Proposition 4.10.1(b) of [118] and Chapter 5 of [69]. Hence together with the transition kernel T, we get from the marking theorem, that $\sum_{\ell=1}^{\mathcal{M}} \delta_{\widetilde{\mathbf{X}}_\ell, g(\widetilde{\mathbf{X}}_\ell)}$ is a Poisson point process on the product space $[0,1]^d \times \mathbb{R}$. Since the neighborhoods $NB(\mathcal{B}_j)$ are by construction disjoint sets for different $j \in \mathcal{J}_s$, the processes

$$\sum_{\ell=1}^{\mathcal{M}} \delta_{\widetilde{\mathbf{X}_\ell}, g_j(\widetilde{\mathbf{X}_\ell})} \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}},$$

are independent Poisson point processes for different $j \in \mathcal{J}_s$. Hence, conditionally on the $\mathbf{X}_i$, the random variables $Z_j = \sum_{\ell=1}^{\mathcal{M}} g_j(\widetilde{\mathbf{X}}_\ell) \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}$, $j \in \mathcal{J}_s$ are jointly independent.

To apply Bernstein's inequality, it remains to check that there exists U and v such that $\sum_{j \in \mathcal{J}_s} \mathbb{E}_{f_0}[|Z_j|^m] \leq \frac{1}{2} m! U^{m-2} v$, for $m = 2, 3, \ldots$.

We have conditionally on $\mathbf{X}_i$ that

$$\begin{aligned}
&\mathbb{E}_{f_0}\Big[\Big|\sum_{\mathbf{X}_i \in \mathcal{B}_j} W(\mathbf{X}_i) C_{i,k}\Big|^m \Big| \mathbf{X}_1, \ldots, \mathbf{X}_n\Big] \qquad (3.9.15)\\
&= \mathbb{E}_{f_0}\Big[\Big|\sum_{\mathbf{X}_i \in \mathcal{B}_j} \Big(\sum_{\ell=1}^{\mathcal{M}} \prod_{r=1}^{d} K\Big(\frac{\widetilde{X}_{\ell,r} - X_{i,r}}{h_n}\Big)\Big) C_{i,k}\Big|^m \Big| \mathbf{X}_1, \ldots, \mathbf{X}_n\Big]\\
&\overset{(i)}{\leq} \mathbb{E}_{f_0}\Big[\Big(\sum_{\mathbf{X}_i \in \mathcal{B}_j} \Big(\sum_{\ell=1}^{\mathcal{M}} \Big|\prod_{r=1}^{d} K\Big(\frac{\widetilde{X}_{\ell,r} - X_{i,r}}{h_n}\Big)\Big|\Big) |C_{i,k}|\Big)^m \Big| \mathbf{X}_1, \ldots, \mathbf{X}_n\Big]\\
&\overset{(ii)}{=} \mathbb{E}_{f_0}\Big[\Big(\sum_{\mathbf{X}_i \in \mathcal{B}_j} \Big(\sum_{\ell=1}^{\mathcal{M}} \Big|\prod_{r=1}^{d} K\Big(\frac{\widetilde{X}_{\ell,r} - X_{i,r}}{h_n}\Big)\Big| \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}\Big) |C_{i,k}|\Big)^m \Big| \mathbf{X}_1, \ldots, \mathbf{X}_n\Big]\\
&\overset{(iii)}{\leq} \mathbb{E}_{f_0}\Big[\Big(\sum_{\mathbf{X}_i \in \mathcal{B}_j} \Big(\sum_{\ell=1}^{\mathcal{M}} \|K\|_\infty^d \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}\Big) |C_{i,k}|\Big)^m \Big| \mathbf{X}_1, \ldots, \mathbf{X}_n\Big]\\
&\overset{(iv)}{=} \mathbb{E}_{f_0}\Big[\Big(\sum_{\ell=1}^{\mathcal{M}} \|K\|_\infty^d \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}\Big)^m \Big(\sum_{\mathbf{X}_i \in \mathcal{B}_j} |C_{i,k}|\Big)^m \Big| \mathbf{X}_1, \ldots, \mathbf{X}_n\Big]\\
&\overset{(v)}{=} \|K\|_\infty^{dm} \Big(\sum_{\mathbf{X}_i \in \mathcal{B}_j} |C_{i,k}|\Big)^m \mathbb{E}_{f_0}\Big[\Big(\sum_{\ell=1}^{\mathcal{M}} \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}\Big)^m\Big].
\end{aligned}$$

Where (i) follows from the triangle inequality. For (ii) we used that $\mathbf{X}_i \in \mathcal{B}_j$ and that K has support in $[-1,1]$, so if $\widetilde{\mathbf{X}}_\ell$ is outside $NB(\mathcal{B}_j)$ then $\prod_{r=1}^d K\left(\frac{\widetilde{X}_{\ell,r}-X_{i,r}}{h_n}\right) = 0$. For (iii) we use that $\|K\|_\infty < \infty$ and that all terms are non-negative. The equality (iv) follows from observing that $\sum_{\ell=1}^{\mathcal{M}} \|K\|_\infty^d \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}$ does not depend on i and can be taken out of the sum. Finally (v) follows by taking all the constants out of the expectation, recalling that $C_{i,k}$ is $\mathbf{X}_1, \dots, \mathbf{X}_n$ measurable.

Since $\widetilde{\mathbf{X}}_\ell$ are i.i.d. and $\mathcal{M} \sim \text{Poisson}(n)$, we have $\sum_{\ell=1}^{\mathcal{M}} \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}} \sim \text{Poisson}(n\widetilde{p}_j)$, where $\widetilde{p}_j$ denotes the probability that $\mathbf{X} \in NB(\mathcal{B}_j)$. Expressing the moments of the Poisson distribution as Bell polynomials [2] gives

$$\mathbb{E}_{f_0}\left[\left(\sum_{\ell=1}^{\mathcal{M}} \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}\right)^m\right] = \sum_{t=0}^m (n\widetilde{p}_j)^t \left\{ {m \atop t} \right\} \le (n\widetilde{p}_j \vee 1)^m \sum_{t=0}^m \left\{ {m \atop t} \right\},$$

where $\left\{ {m \atop t} \right\}$ denote the Stirling numbers of the second kind. The m-th Bell number equals the sum $\sum_{t=0}^m \left\{ {m \atop t} \right\}$. Applying now the bound on Bell numbers derived in Theorem 2.1 of [18] gives

$$\sum_{t=0}^m \left\{ {m \atop t} \right\} \le \left(\frac{m}{\log(m+1)}\right)^m.$$

Due to $m \ge 2$, $\log(m+1) \ge \log(3) > 1$ and the right hand side of the previous display can be upper bounded by m^m. Using Stirling's formula ([120]) again, we get that $\sqrt{2\pi m} m^m e^{-m} \le m!$. Since $m \ge 2$, $\sqrt{2\pi m} \ge e$ and $m^m \le m! e^{m-1}$. Thus

$$\mathbb{E}_{f_0}\left[\left(\sum_{\ell=1}^{\mathcal{M}} \mathbb{1}_{\{\widetilde{\mathbf{X}}_\ell \in NB(\mathcal{B}_j)\}}\right)^m\right] \le m! e^{m-1} (n\widetilde{p}_j \vee 1)^m \le m! e^{m-1} (F 3^d n h_n^d)^m.$$

The last inequality follows from observing that $\widetilde{p}_j \le F 3^d h_n^d$ (the upper bound on f_0 times the Lebesgue measure of $NB(\mathcal{B}_j)$) and that $3^d F n h_n^d \ge 3^d F \log(n) \ge 1$. Combined with (3.9.15), this leads to

$$\mathbb{E}_{f_0}\left[\left|\sum_{\mathbf{X}_i \in \mathcal{B}_j} W(\mathbf{X}_i) C_{i,k}\right|^m \middle| \mathbf{X}_1, \dots, \mathbf{X}_n\right] \le m! e^{m-1} (F 3^d n h_n^d)^m \|K\|_\infty^{dm} \left(\sum_{\mathbf{X}_i \in \mathcal{B}_j} |C_{i,k}|\right)^m.$$

The previous inequality suggests to take the parameters v and U in Bernstein's inequality as upper bounds of $\sum_{j \in \mathcal{J}_s} (e\|K\|_\infty^d)^2 (F 3^d n h_n^d)^2 (\sum_{\mathbf{X}_i \in \mathcal{B}_j} |C_{i,k}|)^2$ and $e\|K\|_\infty^d 3^d F n h_n^d \sum_{\mathbf{X}_i \in \mathcal{B}_j} |C_{i,k}|$, respectively. To find a convenient expression for v, observe that

$$\begin{aligned}
&\sum_{j\in\mathcal{J}_s}(e\|K\|_\infty^d)^2(F3^dnh_n^d)^2\bigg(\sum_{\mathbf{X}_i\in\mathcal{B}_j}|C_{i,k}|\bigg)^2\\
&=\sum_{j\in\mathcal{J}_s}(e3^d\|K\|_\infty^dF)^2n^2h_n^{2d}\bigg(\sum_{\mathbf{X}_i\in\mathcal{B}_j}\bigg|\frac{(f_k(\mathbf{X}_i)-f_0(\mathbf{X}_i))}{nh_n^dz_k}\mathbb{1}_A\bigg|\bigg)^2\\
&=\sum_{j\in\mathcal{J}_s}\frac{(e3^d\|K\|_\infty^dF)^2}{z_k^2}\bigg(\sum_{\mathbf{X}_i\in\mathcal{B}_j}|f_k(\mathbf{X}_i)-f_0(\mathbf{X}_i)|\,\mathbb{1}_A\bigg)^2.
\end{aligned}$$

By Cauchy-Schwarz,

$$\begin{aligned}
\bigg(\sum_{\mathbf{X}_i\in\mathcal{B}_j}|f_k(\mathbf{X}_i)-f_0(\mathbf{X}_i)|\mathbb{1}_A\bigg)^2&\leq\bigg(\mathbb{1}_A\sum_{\mathbf{X}_i\in\mathcal{B}_j}1^2\bigg)\bigg(\sum_{\mathbf{X}_i\in\mathcal{B}_j}\big(f_k(\mathbf{X}_i)-f_0(\mathbf{X}_i)\big)^2\bigg)\\
&\leq 2^{d+3}F\log(n)\sum_{\mathbf{X}_i\in\mathcal{B}_j}\big(f_k(\mathbf{X}_i)-f_0(\mathbf{X}_i)\big)^2,
\end{aligned}$$

where for the last inequality we used that the definition of the event A in (3.9.4) implies $\sum_{\mathbf{X}_i\in\mathcal{B}_j}1\leq 2^{d+3}F\log(n)$. By (3.9.8), $z_k\geq\sqrt{n}\|f_k-f_0\|_n$. Moreover, $\sum_{i=1}^n=\sum_{j\in\mathcal{J}_s}\sum_{\mathbf{X}_i\in\mathcal{B}_j}$ and thus

$$\begin{aligned}
&\sum_{j\in\mathcal{J}_s}(e\|K\|_\infty^d)^2(F3^dnh_n^d)^2\bigg(\sum_{\mathbf{X}_i\in\mathcal{B}_j}|C_{i,k}|\bigg)^2\\
&\leq\sum_{j\in\mathcal{J}_s}\frac{(e3^d\|K\|_\infty^dF)^2}{n\|f_k-f_0\|_n^2}2^{d+3}F\log(n)\bigg(\sum_{\mathbf{X}_i\in\mathcal{B}_j}(f_k(\mathbf{X}_i)-f_0(\mathbf{X}_i))^2\bigg)\\
&=2^{d+3}\frac{(e3^d\|K\|_\infty^d)^2}{n\|f_k-f_0\|_n^2}F^3\log(n)n\|f_k-f_0\|_n^2\\
&=2^{d+3}(e3^d\|K\|_\infty^d)^2F^3\log(n).
\end{aligned}$$

Hence we can take $v=2^{d+3}(e3^d\|K\|_\infty^d)^2F^3\log(n)$ in Bernstein's inequality.

To obtain a convenient expression for the U in Bernstein's inequality, we now bound $\sum_{\mathbf{X}_i\in\mathcal{B}_j}|C_{i,k}|$. Using that by $z_k\geq\sqrt{\log(\mathcal{N}_\mathcal{F}(\delta))}$, that f_k and f_0 are bounded by F, and that on the event A, $\sum_{\mathbf{X}_i \in \mathcal{B}_j} \leq 2^{d+3} F \log(n)$ gives

$$\sum_{\mathbf{X}_i \in \mathcal{B}_j} |C_{i,k}| = \sum_{\mathbf{X}_i \in \mathcal{B}_j} \frac{|f_k(\mathbf{X}_i) - f_0(\mathbf{X}_i)|}{nh_n^d z_k} \mathbb{1}_A \leq \frac{2F}{nh_n^d \sqrt{\log(\mathcal{N}_{\mathcal{F}}(\delta))}} \sum_{\mathbf{X}_i \in \mathcal{B}_j} \mathbb{1}_A$$
$$\leq \frac{2^{d+3} F^2 \log(n)}{nh_n^d \sqrt{\log(\mathcal{N}_{\mathcal{F}}(\delta))}}.$$

Hence it holds that

$$e\|K\|_\infty^d 3^d F n h_n^d \Bigg(\sum_{\mathbf{X}_i \in \mathcal{B}_j} |C_{i,k}| \Bigg) \leq \frac{2^{d+3} e \|K\|_\infty^d 3^d F^3 \log(n)}{\sqrt{\log(\mathcal{N}_{\mathcal{F}}(\delta))}}.$$

The support of the kernel is contained in $[-1,1]$. This means that $1 \leq 2\|K\|_\infty$ and consequently, $e3^d\|K\|_\infty^d \geq 1$. Thus, setting $U = v/\sqrt{\log(\mathcal{N}_{\mathcal{F}}(\delta))}$ with $v = 2^{d+3}(e3^d\|K\|_\infty^d)^2 F^3 \log(n)$, as above, we find $U \geq 2^{d+3} e \|K\|_\infty^d 3^d F^3 \log(n)/\sqrt{\log(\mathcal{N}_{\mathcal{F}}(\delta))}$ and obtain

$$\sum_{j \in \mathcal{J}_s} \mathbb{E}_{f_0} \Bigg[\Bigg| \sum_{\mathbf{X}_i \in \mathcal{B}_j} W(\mathbf{X}_i) C_{i,k} \Bigg|^m \Bigg] \leq \frac{m!}{2} v U^{m-2},$$

for all $m = 2, 3, \ldots$. Consequently we can apply Bernstein's inequality with those choices for U and v.

Using Bernstein's inequality on the sum over the variables Z_j with the bound U and v as defined above we get that

$$\mathbb{P}_{f_0}\Bigg(3^d \Bigg| \sum_{j \in \mathcal{J}_s} \sum_{\mathbf{X}_i \in \mathcal{B}_j} \big(W(\mathbf{X}_i) - \mathbb{E}_{f_0}[W(\mathbf{X}_i)|\mathbf{X}_i] \big) C_{i,k} \Bigg| \geq \sqrt{t} \,\Bigg|\, \mathbf{X}_1, \ldots, \mathbf{X}_n \Bigg)$$
$$= \mathbb{P}_{f_0}\Bigg(\Bigg| \sum_{j \in \mathcal{J}_s} (Z_j - \mathbb{E}_{f_0}[Z_j|\mathbf{X}_i]) \Bigg| \geq 3^{-d}\sqrt{t} \,\Bigg|\, \mathbf{X}_1, \ldots, \mathbf{X}_n \Bigg)$$
$$\leq 2 \exp\left(- \frac{t 3^{-2d}}{2\left(v + U 3^{-d}\sqrt{t}\right)} \right)$$
$$= 2 \exp\left(- \frac{t 3^{-2d}}{2v\left(1 + 3^{-d}\sqrt{t/\log(\mathcal{N}_{\mathcal{F}}(\delta))}\right)} \right).$$

If $t \geq 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta))$, the previous expression can be further bounded by

$$\leq 2 \exp\left(- \frac{\sqrt{t \log(\mathcal{N}_{\mathcal{F}}(\delta))} 3^{-d}}{4v} \right). \tag{3.9.16}$$

Observe that this gives us an upper bound that is the same for all collections of bins $\mathcal{J}_s$ and all cover centers k. Choosing $a = 64v^2 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta))$ in (3.9.14) gives

$$\begin{aligned}
&\mathbb{E}_{f_0}\left[\frac{\xi_{k^*}^2}{z_{k^*}^2}\middle|\mathbf{X}_1, \ldots, \mathbf{X}_n\right] \\
&\overset{(i)}{\leq} 64v^2 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta)) \\
&+ 2\mathcal{N}_{\mathcal{F}}(\delta) 3^d \sqrt{2e\pi n} \int_{64v^2 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta))}^{\infty} \exp\left(-\sqrt{t}\frac{\sqrt{\log(\mathcal{N}_{\mathcal{F}}(\delta))}3^{-d}}{4v}\right) dt \\
&\overset{(ii)}{=} 64v^2 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta)) \\
&+ 4\mathcal{N}_{\mathcal{F}}(\delta) 3^d \sqrt{2e\pi n}(16v^2 3^{2d}) \frac{(2\log(\mathcal{N}_{\mathcal{F}}(\delta)) + 1)}{\log(\mathcal{N}_{\mathcal{F}}(\delta))} \exp\left(-2\log(\mathcal{N}_{\mathcal{F}}(\delta))\right) \\
&\overset{(iii)}{\leq} 64v^2 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta)) + 1280 v^2 3^{3d} \\
&\overset{(iv)}{\leq} (2^{d+5} 10 (e\|K\|_\infty^d)^2 F^3 \log(n))^2 3^{7d} \log(\mathcal{N}_{\mathcal{F}}(\delta)),
\end{aligned}$$

where for (i) we used (3.9.16) combined with the observation that if $t \geq 64v^2 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta))$, then $t \geq 3^{2d} \log(\mathcal{N}_{\mathcal{F}}(\delta))$, since $v \geq 1$. For (ii) we used that $\int_{b^2}^{\infty} e^{-\sqrt{uc}} du = 2\int_b^{\infty} s e^{-sc} ds = 2(bc+1)e^{-bc}/c^2$. For (iii) we used that $\log(\mathcal{N}_{\mathcal{F}}(\delta)) \geq 1$ so $(2\log(\mathcal{N}_{\mathcal{F}}(\delta))+1)/\log(\mathcal{N}_{\mathcal{F}}(\delta)) \leq 4$ and $\mathcal{N}_{\mathcal{F}}(\delta) \geq n$, $\sqrt{2e\pi} \leq 5$, $\log(\mathcal{N}_{\mathcal{F}}(\delta)) \geq 1$. For (iv) we substituted $v = 2^{d+3}(e3^d\|K\|_\infty^d)^2 F^3 \log(n)$ and used that $\sqrt{1344} = 4\sqrt{84}$ and $\sqrt{84} \leq 10$.

Together with (3.9.9), this yields

$$\begin{aligned}
&\left|\mathbb{E}_{f_0}\left[\frac{2}{n}\xi_{k^*}\right]\right| \\
\leq& 2\frac{\sqrt{\widehat{R}_n(\widehat{f}, f_0) + \delta + \sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}}}}{\sqrt{n}} \sqrt{\mathbb{E}_{f_0}\left[\frac{\xi_{k^*}^2}{z_{k^*}^2}\right]} \\
\leq& 2\frac{\sqrt{\widehat{R}_n(\widehat{f}, f_0) + \delta + \sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}}}}{\sqrt{n}} \sqrt{\left(2^{d+5} 10 e^2 \|K\|_\infty^{2d} F^3 \log(n)\right)^2 3^{7d} \log(\mathcal{N}_{\mathcal{F}}(\delta))} \\
=& \left(\sqrt{\widehat{R}_n(\widehat{f}, f_0) + \delta + \sqrt{\frac{\log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}}}\right) 2^{d+6} 10 e^2 \|K\|_\infty^{2d} F^3 \log(n) \sqrt{\frac{3^{7d} \log(\mathcal{N}_{\mathcal{F}}(\delta))}{n}}.
\end{aligned}$$

Inserting this bound in (3.9.6) together with (3.9.7) gives a bound for (IV). Together with (3.9.3) and (3.9.2) and combining the terms with δ, using that $\log^2(n)\log(\mathcal{N}_{\mathcal{F}}(\delta)) \leq n$ finishes the proof. □

Proof of Proposition 3.6.3. Expanding the square yields

$$\begin{aligned}\big(\widehat{f}(\mathbf{X}_i)-f_0(\mathbf{X}_i)\big)^2 &= \big(\widehat{f}(\mathbf{X}_i)-Y_i+Y_i-f_0(\mathbf{X}_i)\big)^2\\ &= (\widehat{f}(\mathbf{X}_i)-Y_i)^2+2(\widehat{f}(\mathbf{X}_i)-Y_i)(Y_i-f_0(\mathbf{X}_i))+(Y_i-f_0(\mathbf{X}_i))^2.\end{aligned}$$

We use this identity to rewrite the definition $\widehat{R}_n(\widehat{f},f_0)=\mathbb{E}_{f_0}[\frac{1}{n}\sum_{i=1}^n(\widehat{f}(\mathbf{X}_i)-f_0(\mathbf{X}_i))^2]$. Applying moreover that for any fixed $f\in\mathcal{F}$, we have by definition of $\Delta_n(\widehat{f},f_0)$ that

$$\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n(Y_i-\widehat{f}(\mathbf{X}_i))^2\right]\le\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n(Y_i-f(\mathbf{X}_i))^2\right]+\Delta_n(\widehat{f},f_0)$$

yields

$$\begin{aligned}&\widehat{R}_n(\widehat{f},f_0)\\ &=\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n\Big((\widehat{f}(\mathbf{X}_i)-Y_i)^2+2(\widehat{f}(\mathbf{X}_i)-Y_i)(Y_i-f_0(\mathbf{X}_i))+(Y_i-f_0(\mathbf{X}_i))^2\Big)\right]\\ &\le\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n\Big((f(\mathbf{X}_i)-Y_i)^2+2(\widehat{f}(\mathbf{X}_i)-Y_i)(Y_i-f_0(\mathbf{X}_i))+(Y_i-f_0(\mathbf{X}_i))^2\Big)\right]\\ &+\Delta_n(\widehat{f},f_0)\\ &=\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n\big((f(\mathbf{X}_i)-Y_i)^2+2(f(\mathbf{X}_i)-Y_i)(Y_i-f_0(\mathbf{X}_i))+(Y_i-f_0(\mathbf{X}_i))^2\big)\right]\\ &+\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n(Y_i-f_0(\mathbf{X}_i))(\widehat{f}(\mathbf{X}_i)-f(\mathbf{X}_i))\right]+\Delta_n(\widehat{f},f_0)\\ &=\mathbb{E}_{f_0}\left[\frac{1}{n}\sum_{i=1}^n(f(\mathbf{X}_i)-f_0(\mathbf{X}_i))^2\right]+\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n(Y_i-f_0(\mathbf{X}_i))(\widehat{f}(\mathbf{X}_i)-f(\mathbf{X}_i))\right]\\ &+\Delta_n(\widehat{f},f_0)\\ &=\mathbb{E}_{\mathbf{X}}\left[(f(\mathbf{X})-f_0(\mathbf{X}))^2\right]+\mathbb{E}_{f_0}\left[\frac{2}{n}\sum_{i=1}^n\epsilon_i(\widehat{f}(\mathbf{X}_i)-f(\mathbf{X}_i))\right]+\Delta_n(\widehat{f},f_0),\end{aligned}$$

where for the last equality we used that the $\mathbf{X}_i$ are independent and have the same distribution as $\mathbf{X}$.

Combined with Lemma 3.6.2, this yields

$$\widehat{R}_n(\widehat{f},f_0)\le\mathbb{E}_{\mathbf{X}}\left[(f(\mathbf{X})-f_0(\mathbf{X}))^2\right]$$

$$
\begin{aligned}
&+ \sqrt{\widehat{R}_n(\widehat{f}, f_0)}2^{d+6}14e^2\|K\|_\infty^{2d}F^3\log(n)\sqrt{\frac{3^{7d}\log(\mathcal{N}_\mathcal{F}(\delta))}{n}} \\
&+ 2^{d+6}14e^2\|K\|_\infty^{2d}F^3 3^{\frac{7d}{2}}\log(n)\frac{\log(\mathcal{N}_\mathcal{F}(\delta))}{n} \\
&+ \delta 2^{d+6}14e^2\|K\|_\infty^{2d}F^3 3^{\frac{7d}{2}} + \frac{46F^2 2^d\|K\|_\infty^d}{n} \\
&+ 8h_n^{2\beta}F^2d^{2\beta}\|K\|_\infty^{2d} + \frac{\mathbb{E}_\mathbf{X}[(f_0(\mathbf{X}) - f(\mathbf{X}))^2]}{4} + \frac{\widehat{R}_n(\widehat{f}, f_0)}{4} + \Delta_n(\widehat{f}, f_0).
\end{aligned}
$$

Rewriting this and upper bounding constants, yields

$$
\begin{aligned}
\widehat{R}_n(\widehat{f}, f_0) \le{}& \frac{5}{3}\mathbb{E}_\mathbf{X}\left[(f(\mathbf{X}) - f_0(\mathbf{X}))^2\right] \\
&+ \sqrt{\widehat{R}_n(\widehat{f}, f_0)}2^{d+6}19e^2\|K\|_\infty^{2d}F^3\log(n)\sqrt{\frac{3^{7d}\log(\mathcal{N}_\mathcal{F}(\delta))}{n}} \\
&+ 2^{d+6}19e^2\|K\|_\infty^{2d}F^3 3^{\frac{7d}{2}}\log(n)\frac{\log(\mathcal{N}_\mathcal{F}(\delta))}{n} + \delta 2^{d+6}19e^2\|K\|_\infty^{2d}F^3 3^{\frac{7d}{2}} \\
&+ \frac{62F^2 2^d\|K\|_\infty^d}{n} + 11h_n^{2\beta}F^2d^{2\beta}\|K\|_\infty^{2d} + \frac{4}{3}\Delta_n(\widehat{f}, f_0).
\end{aligned}
$$

For real numbers a, c, d, satisfying $|a| \le 2\sqrt{a}c+d$, we have $|a| \le 2\sqrt{a}c+d \le \frac{1}{2}|a|+2c^2+d$ and thus $|a| \le 2d + 4c^2$. Applying this inequality with $a = \widehat{R}_n(\widehat{f}, f_0)$,

$$
c = 2^{d+6}19e^2\|K\|_\infty^{2d}F^3\log(n)\sqrt{\frac{3^{7d}\log(\mathcal{N}_\mathcal{F}(\delta))}{n}}
$$

and

$$
\begin{aligned}
d ={}& \delta 2^{d+6}19e^2\|K\|_\infty^{2d}F^3 3^{\frac{7d}{2}} + \frac{62F^2 2^d\|K\|_\infty^d}{n} \\
&+ 2^{d+6}19e^2\|K\|_\infty^{2d}F^3 3^{\frac{7d}{2}}\log(n)\frac{\log(\mathcal{N}_\mathcal{F}(\delta))}{n} \\
&+ 11h_n^{2\beta}F^2d^{2\beta}\|K\|_\infty^{2d} + \frac{4}{3}\Delta_n(\widehat{f}, f_0) + \frac{5}{3}\mathbb{E}_\mathbf{X}\left[(f(\mathbf{X}) - f_0(\mathbf{X}))^2\right]
\end{aligned}
$$

yields the result. □

Proposition 3.9.1. *$|\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]| \le h_n^\beta d^\beta\|K\|_\infty^d F$.*

Proof. By the construction of the ϵ_i in (3.2.2) and (3.2.3), $\epsilon_i = \widehat{f}_{\mathrm{KDE}}(\mathbf{X}_i) - f_0(\mathbf{X}_i)$. Using moreover the definition of the multivariate kernel density estimator in (3.2.1) and writing $|\mathbf{v}|^{\boldsymbol{\alpha}}$ for $|v_1|^{\alpha_1}\cdot\ldots\cdot|v_d|^{\alpha_d}$, we obtain

$$
\begin{aligned}
&\big|\mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]\big| = \left|\mathbb{E}_{f_0}\left[\frac{1}{nh_n^d}\sum_{\ell=1}^{n}\prod_{r=1}^{d} K\left(\frac{X'_{\ell,r}-X_{i,r}}{h_n}\right) - f_0(\mathbf{X}_i)\middle|\mathbf{X}_i\right]\right| \\
&\overset{(i)}{=} \left|\frac{1}{h_n^d}\int_{[0,1]^d} f_0(\mathbf{u})\prod_{r=1}^{d} K\left(\frac{u_r-X_{i,r}}{h_n}\right)d\mathbf{u} - f_0(\mathbf{X}_i)\right| \\
&\overset{(ii)}{=} \left|\int_{\mathbb{R}^d}\left(\prod_{r=1}^{d} K(v_r)\right) f_0(X_{i,1}+v_1h_n,\ldots,X_{i,d}+v_dh_n)\,d\mathbf{v} - f_0(\mathbf{X}_i)\right| \\
&\overset{(iii)}{=} \left|\int_{\mathbb{R}^d}\left(\prod_{r=1}^{d} K(v_r)\right)\Big(f_0(X_{i,1}+v_1h_n,\ldots,X_{i,d}+v_dh_n) - f_0(\mathbf{X}_i)\Big)\,d\mathbf{v}\right| \\
&\overset{(iv)}{=} \Bigg|\int_{\mathbb{R}^d}\left(\prod_{r=1}^{d} K(v_r)\right) \\
&\qquad\times\Bigg(\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1\le\lfloor\beta\rfloor-1,\boldsymbol{\alpha}\neq 0}\frac{(h_n\mathbf{v})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}(\partial^{\boldsymbol{\alpha}}f_0)(\mathbf{X}_i) \\
&\qquad\qquad\qquad + \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1=\lfloor\beta\rfloor}\frac{(h_n\mathbf{v})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}(\partial^{\boldsymbol{\alpha}}f_0)(\mathbf{X}_i+h_n\tau\mathbf{v})\Bigg)\,d\mathbf{v}\Bigg| \\
&\overset{(v)}{=} \left|\int_{\mathbb{R}^d}\left(\prod_{r=1}^{d} K(v_r)\right)\left(\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1=\lfloor\beta\rfloor}\frac{(h_n\mathbf{v})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}\big((\partial^{\boldsymbol{\alpha}}f_0)(\mathbf{X}_i+h_n\tau\mathbf{v}) - (\partial^{\boldsymbol{\alpha}}f_0)(\mathbf{X}_i)\big)\right)d\mathbf{v}\right| \\
&\overset{(vi)}{\le} h_n^{\lfloor\beta\rfloor}\int_{\mathbb{R}^d}\left|\prod_{r=1}^{d} K(v_r)\right|\left(\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1=\lfloor\beta\rfloor}\frac{|\mathbf{v}|^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}\big|(\partial^{\boldsymbol{\alpha}}f_0)(\mathbf{X}_i+h_n\tau\mathbf{v}) - (\partial^{\boldsymbol{\alpha}}f_0)(\mathbf{X}_i)\big|\right)d\mathbf{v} \\
&\overset{(vii)}{\le} h_n^{\lfloor\beta\rfloor}\|K\|_\infty^d\int_{[0,1]^d}\left(\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1=\lfloor\beta\rfloor}\frac{|\mathbf{v}|^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}|h_n\tau\mathbf{v}|_\infty^{\beta-\lfloor\beta\rfloor}F\right)d\mathbf{v} \\
&\overset{(viii)}{\le} h_n^{\beta}\|K\|_\infty^d F\int_{[0,1]^d}\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1=\lfloor\beta\rfloor}\frac{1}{\boldsymbol{\alpha}!}\,d\mathbf{v} \\
&\overset{(ix)}{\le} h_n^{\beta}\|K\|_\infty^d d^{\beta}F.
\end{aligned}
$$

Here we used for (i) that the $\mathbf{X}'_\ell$ are i.i.d. and independent of $\mathbf{X}_i$. For (ii) we substituted the transformed variables $v_r = (u_r - X_{i,r})/h_n$ and used that f_0 vanishes outside $[0,1]^d$, since f_0 has support in $[0,1]^d$ and is continuous on $\mathbb{R}^d$. For *(iii)* we used that a kernel integrates to 1 and that $f_0(\mathbf{X}_i)$ is a constant with respect to the integration variables. Step *(iv)* applies $\lfloor\beta\rfloor$-order Taylor expansion, that is, for a suitable $\tau \in (0,1)$,

$$\begin{aligned} f_0(\mathbf{X}_i + h_n\mathbf{v}) = f_0(\mathbf{X}_i) + &\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1 \le \lfloor\beta\rfloor-1, \boldsymbol{\alpha}\neq 0} \frac{(h_n\mathbf{v})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}(\partial^{\boldsymbol{\alpha}} f_0)(\mathbf{X}_i) \\ &+ \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1 = \lfloor\beta\rfloor} \frac{(h_n\mathbf{v})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}(\partial^{\boldsymbol{\alpha}} f_0)(\mathbf{X}_i + h_n\tau\mathbf{v}) \end{aligned}$$

see Theorem 2.2.5 in [66]. For *(v)* we used that K is a kernel of order $\lfloor\beta\rfloor$ and therefore $\int v^m K(v)\,dv = 0$ for all $m = 1,\ldots,\lfloor\beta\rfloor$. For *(vi)* we used that $h_n^{\lfloor\beta\rfloor}$ appears in every term of the sum. Jensen's inequality and triangle inequality are moreover applied to move the absolute value inside the integral and the sum. For *(vii)* we used that f_0 is in the β-Hölder ball with radius F and that K has support contained in $[-1,1]$. For *(viii)* we used that $|\tau| \le 1$. To see *(ix)*, observe that for the multinomial distribution with number of trials $\lfloor\beta\rfloor$ and d event probabilities $(1/d,\ldots,1/d)$, we have

$$1 = \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1 = \lfloor\beta\rfloor} \frac{\lfloor\beta\rfloor!}{\boldsymbol{\alpha}!}\Big(\frac{1}{d}\Big)^{\alpha_1} \cdot\ldots\cdot \Big(\frac{1}{d}\Big)^{\alpha_d} = \lfloor\beta\rfloor! d^{-\lfloor\beta\rfloor} \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1 = \lfloor\beta\rfloor} \frac{1}{\boldsymbol{\alpha}!} \ge d^{-\beta} \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1 = \lfloor\beta\rfloor} \frac{1}{\boldsymbol{\alpha}!}.$$

□

Proposition 3.9.2. $\mathbb{E}_{f_0}[|\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]|] \le F\|K\|_\infty^d 2^{d+1}$.

Proof. By definition, $\epsilon_i = Y_i - f_0(\mathbf{X}_i)$. Together with conditioning on $\mathbf{X}_i$, triangle inequality and Jensen's inequality this yields

$$\begin{aligned} \mathbb{E}_{f_0}[|\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]|] &= \mathbb{E}_{f_0}[|Y_i - \mathbb{E}_{f_0}[Y_i|\mathbf{X}_i]|] \le 2\mathbb{E}_{f_0}\left[\mathbb{E}_{f_0}[|Y_i||\mathbf{X}_i]\right] \\ &\le 2\sum_{\ell=1}^n \frac{1}{nh_n^d}\mathbb{E}_{f_0}\left[\mathbb{E}_{f_0}\left[\prod_{r=1}^d \left|K\left(\frac{X'_{\ell,r} - X_{i,r}}{h_n}\right)\right| \,\Big|\, \mathbf{X}_i\right]\right]. \end{aligned} \tag{3.9.17}$$

Using that $\|f_0\|_\infty \le F$ and the kernel K is supported on $[-1,1]$, we get by substitution

$$\begin{aligned} \mathbb{E}_{f_0}\left[\prod_{r=1}^d \left|K\left(\frac{X'_{\ell,r} - X_{i,r}}{h_n}\right)\right| \,\Big|\, \mathbf{X}_i\right] &\le F\int_{\mathbb{R}^d} \prod_{r=1}^d \left|K\left(\frac{u_r - X_{i,r}}{h_n}\right)\right| d\mathbf{u} \\ &= Fh_n^d \int_{\mathbb{R}^d} \prod_{r=1}^d |K(v_r)|\, d\mathbf{v} \\ &\le F\|K\|_\infty^d 2^d h_n^d. \end{aligned}$$

□

Proposition 3.9.3. $\mathbb{E}_{f_0}\left[|\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]|^2\right] \leq 65F^2 2^{2d}\|K\|_\infty^{2d}$.

Proof. By definition, $\epsilon_i = Y_i - f_0(\mathbf{X}_i)$. For a non-negative random-variable T, it holds that $\mathbb{E}[T^2] = \int_0^\infty \mathbb{P}(T^2 \geq t)\,dt = \int_0^\infty \mathbb{P}(T \geq \sqrt{t})\,dt$.

$$\begin{aligned}\mathbb{E}_{f_0}\left[|\epsilon_i - \mathbb{E}_{f_0}[\epsilon_i|\mathbf{X}_i]|^2\right] &= \mathbb{E}_{f_0}\left[\left|Y_i - \mathbb{E}_{f_0}[Y_i|\mathbf{X}_i]\right|^2\right] \\ &= \mathbb{E}_{f_0}\left[\mathbb{E}_{f_0}\left[\left|Y_i - \mathbb{E}_{f_0}[Y_i|\mathbf{X}_i]\right|^2 \,\middle|\, \mathbf{X}_i\right]\right] \\ &= \mathbb{E}_{f_0}\left[\int_0^\infty \mathbb{P}_{f_0}\left(\left|Y_i - \mathbb{E}_{f_0}[Y_i|\mathbf{X}_i]\right| \geq \sqrt{t} \,\middle|\, \mathbf{X}_i\right) dt\right].\end{aligned}$$

The probability can also be written as

$$\begin{aligned}&\int_0^\infty \mathbb{P}_{f_0}\left(\left|Y_i - \mathbb{E}_{f_0}[Y_i|\mathbf{X}_i]\right| \geq \sqrt{t} \,\middle|\, \mathbf{X}_i\right) dt \\ &= \int_0^\infty \mathbb{P}_{f_0}\Bigg(\Bigg|\sum_{\ell=1}^n \Bigg(\prod_{r=1}^d K\left(\frac{X'_{\ell,r} - X_{i,r}}{h_n}\right) \\ &\qquad\qquad - \int_{[0,1]^d} f_0(\mathbf{u}) \prod_{r=1}^d K\left(\frac{u_r - X_{i,r}}{h_n}\right) d\mathbf{u}\Bigg)\Bigg| \geq nh_n^d\sqrt{t}\,\Bigg|\, \mathbf{X}_i\Bigg)\, dt.\end{aligned}$$

This is a sum of i.i.d. random variables minus their expectation (conditionally on $\mathbf{X}_i$). Using that $\|f_0\|_\infty \leq F$ and the kernel K is supported on $[-1,1]$, we get by substitution

$$\begin{aligned}\mathbb{E}_{f_0}\left[\prod_{r=1}^d K^2\left(\frac{X'_{\ell,r} - X_{i,r}}{h_n}\right) \,\middle|\, \mathbf{X}_i\right] &\leq F\int_{\mathbb{R}^d} \prod_{r=1}^d K^2\left(\frac{u_r - X_{i,r}}{h_n}\right) d\mathbf{u} \\ &= Fh_n^d \int_{\mathbb{R}^d} \prod_{r=1}^d K^2(v_r)\, d\mathbf{v} \\ &\leq F\|K\|_\infty^{2d} 2^d h_n^d.\end{aligned}$$

Applying the bounded variable version of Bernstein's inequality in Proposition 3.9.4 (ii) with $v = F\|K\|_\infty^{2d}2^d h_n^d$ and $b = 3\|K\|_\infty^d$ (that is, $b/3 = \|K\|_\infty$), we get that

$$\begin{aligned}&\int_0^\infty \mathbb{P}_{f_0}\Bigg(\Bigg|\sum_{\ell=1}^n \Bigg(\prod_{r=1}^d K\left(\frac{X'_{\ell,r} - X_{i,r}}{h_n}\right) \\ &\qquad\qquad - \int_{[0,1]^d} f_0(\mathbf{u}) \prod_{r=1}^d K\left(\frac{u_r - X_{i,r}}{h_n}\right) d\mathbf{u}\Bigg)\Bigg| \geq nh_n^d\sqrt{t}\Bigg)\, dt\end{aligned}$$

$$
\begin{aligned}
&\le \int_0^\infty 1 \wedge 2\exp\left(-\frac{n^2 h_n^{2d} t}{2(n\|K\|_\infty^{2d} F2^d h_n^d + \|K\|_\infty^d n h_n^d \sqrt{t})}\right) dt \\
&= \int_0^\infty 1 \wedge 2\exp\left(-\frac{n h_n^d t}{2(\|K\|_\infty^{2d} F2^d + \|K\|_\infty^d \sqrt{t})}\right) dt \\
&\overset{(*)}{\le} F^2 2^{2d}\|K\|_\infty^{2d} + 2\int_{F^2 2^{2d}\|K\|_\infty^{2d}}^\infty \exp\left(-\frac{n h_n^d \sqrt{t}}{4\|K\|_\infty^d}\right) dt \\
&\overset{(**)}{=} F^2 2^{2d}\|K\|_\infty^{2d} + \frac{64\|K\|_\infty^{2d}\left(\frac{F2^d\|K\|_\infty^d n h_n^d}{4\|K\|_\infty^d} + 1\right)\exp\left(-\frac{F2^d\|K\|_\infty^d n h_n^d}{4\|K\|_\infty^d}\right)}{n^2 h_n^{2d}} \\
&\overset{(***)}{\le} F^2 2^{2d}\|K\|_\infty^{2d} + 64\|K\|_\infty^{2d}(F2^d + 1) \\
&\overset{(****)}{\le} 65F^2 2^{2d}\|K\|_\infty^{2d},
\end{aligned}
$$

where we used for $(*)$ that $2(\|K\|_\infty^{2d} F2^d + \|K\|_\infty^d \sqrt{t}) \le 4\|K\|_\infty^d \sqrt{t}$ when $t \ge F^2 2^{2d}\|K\|_\infty^{2d}$. For $(**)$ we used that $\int_{b^2}^\infty e^{-a\sqrt{u}}\, du = 2\int_b^\infty s e^{-sa} ds = 2(ba+1)e^{-ba}/a^2$, with $a = nh_n^d/(4\|K\|_\infty^d)$ and $b = F2^d\|K\|_\infty^d$. For $(***)$ we used that $nh_n^d \ge \log(n) \ge 1$ and that $0 < \exp(-x) \le 1$ for $x \ge 0$. For $(****)$ we used that $F2^d + 1 \le 2F2^d \le F2^{2d} \le F^2 2^{2d}$. The result follows from observing that $\mathbb{E}[c] = c$ for $c \in \mathbb{R}$. □

Proposition 3.9.4. *Given independent random variables $Z_1, \dots, Z_n$.*

(i) (moment version) If for some constants U and v_i the moment bounds $\mathbb{E}_{Z_i}[|Z_i|^m] \le \frac{1}{2} m! U^{m-2} v_i$ hold for all $m = 2, 3, \dots,$ and all $i = 1, \dots, n$, then

$$
\mathbb{P}_{Z_i}\left(\left|\sum_{i=1}^n (Z_i - \mathbb{E}_{Z_i}[Z_i])\right| > t\right) \le 2e^{-\frac{t^2}{2v + 2Ut}}, \qquad \text{for all } v \ge \sum_{i=1}^n v_i.
$$

(ii) (bounded version) If for some constants b and v_i, the bounds $|Z_i| \le b$ and $\mathbb{E}_{Z_i}[|Z_i|^2] \le v_i$ hold for all $i = 1, \dots, n$, then,

$$
\mathbb{P}_{Z_i}\left(\left|\sum_{i=1}^n (Z_i - \mathbb{E}_{Z_i}[Z_i])\right| > t\right) \le 2e^{-\frac{t^2}{2v + 2bt/3}}, \qquad \text{for all } v \ge \sum_{i=1}^n v_i.
$$

These formulations of Bernstein's inequality are based on Corollary 2.11 and Equation (2.10) in [25].

Acknowledgments

We want to thank Claire Donnat for pointing us to Lindsey's method.

Chapter 4

Convergence guarantees for forward gradient descent in the linear regression model

Abstract

Renewed interest in the relationship between artificial and biological neural networks motivates the study of gradient-free methods. Considering the linear regression model with random design, we theoretically analyze in this chapter the biologically motivated (weight-perturbed) forward gradient scheme that is based on random linear combination of the gradient. If d denotes the number of parameters and k the number of samples, we prove that the mean squared error of this method converges for $k \gtrsim d^2 \log(d)$ with rate $d^2 \log(d)/k$. Compared to the dimension dependence d for stochastic gradient descent, an additional factor $d \log(d)$ occurs.

4.1 Introduction

Looking at the past developments, it is apparent that artificial neural networks (ANNs) became more powerful the more they resembled the brain. It is therefore anticipated that the future of AI is even more biologically inspired. As in the past, the bottlenecks towards more biologically inspired learning are computational barriers. For instance, shallow networks only became computationally feasible after the backpropagation algorithm was proposed. Deep neural networks were proposed for a longer time but deep learning became implementable after the development of large scale GPU

*This chapter is based on: Thijs Bos and Johannes Schmidt-Hieber. Convergence guarantees for forward gradient descent in the linear regression model. To appear in Journal of Statistical Planning and Inference, Volume **233**. The research has been supported by the NWO/STAR grant 613.009.034b and the NWO Vidi grant VI.Vidi.192.021.*

computing. Neuromorphic computing aims to imitate the brain on computer chips, but is currently not fully scalable.

The mathematics of AI has focused on explaining the state-of-the-art performance of modern machine learning methods and empirically observed phenomena such as the good generalization properties of extreme overparametrization. To shape the future of AI, statistical theory needs more emphasis on anticipating future developments and proposing biologically motivated methods already at a stage before scalable implementations exist.

This chapter aims to analyze a biologically motivated learning rule building on the renewed interest of the differences and similarities between ANNs and biological neural networks (BNNs) [89, 128, 155] which are rooted in the foundational literature from the 1980s [53, 33]. A key difference between ANNs and BNNs is that ANNs are usually trained based on a version of (stochastic) gradient descent, while this seems prohibitive for BNNs. Indeed, to compute the gradient, knowledge of all parameters in the network is required, but biological networks do not posses the capacity to transport this information to each neuron. This suggests that biological networks cannot directly use the gradient to update their parameters [33, 89, 142].

The brain still performs well without gradient descent and can learn tasks with much fewer examples than ANNs. This sparks interest in biologically plausible learning methods that do not require (full) access of the gradient. Such methods are called derivative-free. A simple example of a derivative-free method is to randomly sample in each step a new parameter. If this decreases the loss one keeps the parameter and otherwise discards it. There is a wide variety of derivative-free strategies [32, 83, 135]. Among those, so-called zero-order methods use evaluations of the loss function to build a noisy estimate of the gradient. This substitute is then used to replace the gradient in the gradient descent routine [92, 41]. [128] establishes a connection between the Hebbian learning underlying the local learning of the brain (see e.g. Chapter 6 of [142]) and a specific zero-order method. A statistical analysis of this zero-order scheme is provided in the companion article [129].

In this chapter, we study (weight-perturbed) forward gradient descent. This method is motivated by biological neural networks [13, 117] and lies between full gradient descent methods and derivative-free methods, as only random linear combination of the gradient are required. The form of the random linear combination is related to zero-order estimators, see Section 4.2. Settings with partial access to the gradient have been studied before. For example, [105] proposes a learning method based on directional derivatives for convex functions. In this chapter we specifically derive theoretical guarantees for forward gradient descent in the linear regression model with random design. Theorem 4.3.1 establishes an expression for the expectation. A bound on the mean squared error is provided in Theorem 4.3.3.

The structure of this chapter is as follows. In Section 4.2 we describe the forward gradient descent update rule in the linear regression model. Results are in Section 4.3 and the corresponding proofs can be found in Section 4.4.

Notation

Vectors are denoted by bold letters and we write $\|\cdot\|_2$ for the Euclidean norm. We denote the largest and smallest eigenvalue of a matrix A by the respective expressions $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$. The spectral norm is $\|A\|_S := \sqrt{\lambda_{\max}(A^\top A)}$. The condition number of a positive semi-definite matrix B is $\kappa(B) := \lambda_{\max}(B)/\lambda_{\min}(B)$.

For a random variable U we denote the expectation with respect to U by $\mathbb{E}_U$. The symbol $\mathbb{E}$ stands for an expectation taken with respect to all random variables that are inside that expectation. The (multivariate) normal distribution with mean vector μ and covariance matrix Σ is denoted by $\mathcal{N}(\mu, \Sigma)$.

4.2 Weight-perturbed forward gradient descent

Suppose we want to learn a parameter vector $\boldsymbol{\theta}$ from training data $(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots \in \mathbb{R}^d \times \mathbb{R}$. Stochastic gradient descent (SGD) is based on the iterative update rule

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_{k+1} \nabla L(\boldsymbol{\theta}_k), \quad k = 0, 1, \ldots \tag{4.2.1}$$

with $\boldsymbol{\theta}_0$ some initial value and $L(\boldsymbol{\theta}_k) := L(\boldsymbol{\theta}_k, \mathbf{X}_k, Y_k)$ a loss that depends on the data only through the k-th sample $(\mathbf{X}_k, Y_k)$.

For a standard normal random vector $\boldsymbol{\xi}_{k+1} \sim \mathcal{N}(0, \mathbf{I}_d)$ that is independent of all the other randomness, the quantity $(\nabla L(\boldsymbol{\theta}_k))^\top \boldsymbol{\xi}_{k+1} \boldsymbol{\xi}_{k+1}$ is called the (weight-perturbed) forward gradient [13, 117]. *(Weight-perturbed) forward gradient descent* is then given by the update rule

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_{k+1} \big(\nabla L(\boldsymbol{\theta}_k)\big)^\top \boldsymbol{\xi}_{k+1} \boldsymbol{\xi}_{k+1}, \quad k = 0, 1, \ldots \tag{4.2.2}$$

Assuming that the exogenous noise has unit variance is sufficient. Indeed, generalizing to $\boldsymbol{\xi}_{k+1} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ with variance parameter σ^2 has the same effect as rescaling the learning rate $\alpha_{k+1} \to \sigma^{-2} \alpha_{k+1}$.

Since for a deterministic d-dimensional vector $\mathbf{v}$, one has $\mathbb{E}[\mathbf{v}^t \boldsymbol{\xi}_{k+1} \boldsymbol{\xi}_{k+1}] = \mathbf{v}$, taking the expectation of the weight-perturbed forward gradient descent scheme with respect to the exogenous randomness induced by $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots$ gives

$$\mathbb{E}_{(\boldsymbol{\xi}_i)_{i \geq 1}}[\boldsymbol{\theta}_{k+1}] = \mathbb{E}_{(\boldsymbol{\xi}_i)_{i \geq 1}}[\boldsymbol{\theta}_k] - \alpha_{k+1} \mathbb{E}_{(\boldsymbol{\xi}_i)_{i \geq 1}}[\nabla L(\boldsymbol{\theta}_k)], \tag{4.2.3}$$

resembling the SGD dynamic (4.2.1). If $\nabla L(\boldsymbol{\theta}_k)$ depends on $\boldsymbol{\theta}_k$ linearly then also $\mathbb{E}_{(\boldsymbol{\xi}_i)_{i \geq 1}}[\nabla L(\boldsymbol{\theta}_k)] = \nabla L(\mathbb{E}_{(\boldsymbol{\xi}_i)_{i \geq 1}}[\boldsymbol{\theta}_k])$.

While in expectation, forward gradient descent is related to SGD, the induced randomness of the d-dimensional random vectors $\mathbf{x}_{k+1}$ induces a large amount of noise. To control the high noise level in the dynamic is the main obstacle in the mathematical analysis. One of the implications is that one has to make small steps by choosing a small learning rate to avoid completely erratic behavior. This particularly effects the first phase of the learning.

First order multivariate Taylor expansion shows that $L(\boldsymbol{\theta}_k + \boldsymbol{\xi}_k) - L(\boldsymbol{\theta}_k)$ and $(\nabla L(\boldsymbol{\theta}_k))^\top \boldsymbol{\xi}_{k+1}$ are close. Therefore, forward gradient descent is related to the zero-order method

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_{k+1}\big(L(\boldsymbol{\theta}_k + \boldsymbol{\xi}_k) - L(\boldsymbol{\theta}_k)\big)\boldsymbol{\xi}_k, \tag{4.2.4}$$

[92]. Consequently, forward gradient descent can be viewed as an intermediate step between gradient descent, with full access to the gradient, and zero-order methods that are solely based on (randomly) perturbed function evaluations.

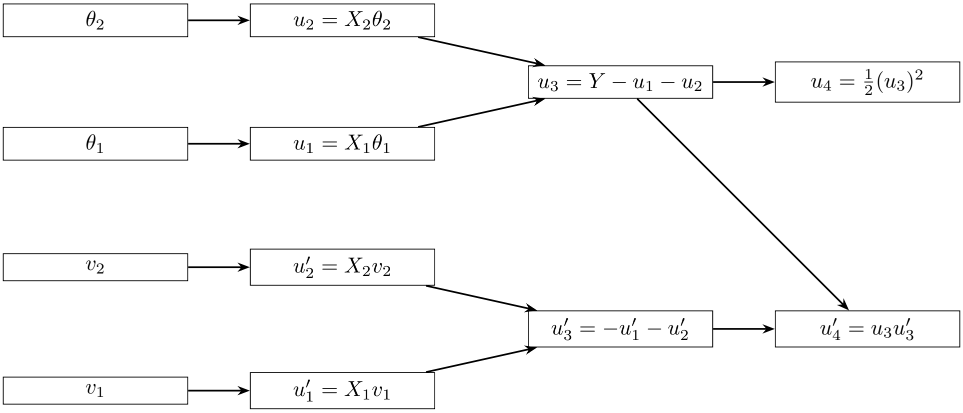


Figure 4.2.1: Computional graphs for computing in a forward pass $L(\boldsymbol{\theta}) = \frac{1}{2}(Y - X_1\theta_1 - X_2\theta_2)^2$ (upper half) and $(\nabla L(\boldsymbol{\theta}))^\top \mathbf{v}$ (lower half).

We now comment on the biological plausibility of forward gradient descent. As mentioned in the introduction, it is widely accepted that the brain cannot perform (full) gradient descent. The backpropagation algorithm decomposes the computation of the gradient in a forward pass and a backward pass. The forward pass evaluates the loss for a training sample by sending signal through the network. This is biologically plausible. For a given vector $\mathbf{v}$, it is even possible to compute both $L(\boldsymbol{\theta}_k)$ and $\big(\nabla L(\boldsymbol{\theta}_k)\big)^\top \mathbf{v}$ in one forward pass, [13, 117, 12]. The construction can be conveniently explained for two variables $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top$, see Figure 4.2.1. The loss function $L(\boldsymbol{\theta}) = \frac{1}{2}(Y - X_1\theta_1 - X_2\theta_2)^2$

is implemented by first computing $u_1 = X_1\theta_1$ and $u_2 = X_2\theta_2$ in parallel. Subsequently, one can infer $u_3 = Y - u_1 - u_2 = Y - X_1\theta_1 - X_2\theta_2$ and $u_4 = \frac{1}{2}(u_3)^2 = L(\boldsymbol{\theta})$. For a given vector $\mathbf{v} = (v_1, v_2)^\top$, the update value $(\nabla L(\boldsymbol{\theta}))^\top \mathbf{v}$ in the forward gradient descent routine can be computed from v_1, v_2, and $u_3 = Y - X_1\theta_1 - X_2\theta_2$. Indeed, after computing X_1v_1 and X_2v_2 in a first step, one can compute $u_3' = -X_1v_1 - X_2v_2$ and finally $u_4' = u_3u_3' = (Y - X_1\theta_1 - X_2\theta_2)(-X_1v_1 - X_2v_2) = -(Y - \mathbf{X}^\top\boldsymbol{\theta})\mathbf{X}^\top\mathbf{v} = (\nabla L(\boldsymbol{\theta}))^\top\mathbf{v}$. For more background on the implementation, see for instance [12].

In [128], it has been shown that under appropriate conditions, Hebbian learning of excitatory neurons in biological neural networks leads to a zeroth-order learning rule that has the same structure as (4.2.4).

To complete this section, we briefly compare forward gradient descent with feedback alignment as both methods are motivated by biological learning and are based on additional randomness. Inspired by biological learning, feedback alignment proposes to replace the learned weights in the backward pass by random weights chosen at the start of the training procedure [88, 89]. The so-called direct feedback alignment method goes even further: instead of back-propagating the gradient through all the layers of the network by the chain-rule, layers are updated with the gradient of the output layer multiplied with a fixed random weight matrix [106, 84]. (Direct) feedback alignment causes the forward weights to change in such a way that the true gradient of the network weights and the substitutes used in the update rule become more aligned [88, 106, 89]. The linear model can be viewed as neural network without hidden layers. The absence of layers means that in the backward step, no weight information is transported between different layers. As a consequence, both feedback alignment and direct feedback alignment collapse in the linear model into standard gradient descent. The conclusion is that feedback alignment and forward gradient descent are not comparable. The argument also shows that to unveil nontrivial statistical properties of feedback alignment, one has to go beyond the linear model. We leave the statistical analysis as an open problem.

4.3 Convergence rates in the linear regression model

We analyze weight-perturbed forward gradient descent for data generated from the d-dimensional linear regression with Gaussian random design. In this framework, we observe i.i.d. pairs $(\mathbf{X}_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, 2, \ldots$ satisfying

$$\mathbf{X}_i \sim \mathcal{N}(0, \Sigma), \quad Y_i = \mathbf{X}_i^\top \boldsymbol{\theta}_\star + \epsilon_i, \quad i = 1, 2, \ldots \tag{4.3.1}$$

with $\boldsymbol{\theta}_\star$ the unknown d-dimensional regression vector, Σ an unknown covariance matrix, and independent noise variables ϵ_i with mean zero and variance one.

For the analysis, we consider the squared loss $L(\boldsymbol{\theta}_k, \mathbf{X}_k, Y_k) = \frac{1}{2}(Y_k - \mathbf{X}_k^\top \boldsymbol{\theta}_k)^2$. The gradient is given by

$$\nabla L(\boldsymbol{\theta}_k) = -\big(Y_k - \mathbf{X}_k^\top \boldsymbol{\theta}_k\big)\mathbf{X}_k. \tag{4.3.2}$$

We now analyze the forward gradient estimator assuming that the initial value $\boldsymbol{\theta}_0$ can be random or deterministic but should be independent of the data. We employ a similar proving strategy as in the recent analysis of dropout in the linear model in [31]. In particular, we will derive a recursive formula for $\mathbb{E}\left[(\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star)(\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star)^\top\right]$. In contrast to this work, we consider a different form of noise and non-constant learning rates.

The first result shows that forward gradient descent does gradient descent in expectation.

Theorem 4.3.1. *We have* $\mathbb{E}[\boldsymbol{\theta}_k] - \boldsymbol{\theta}_\star = \big(\mathbf{I}_d - \alpha_k \Sigma\big)\big(\mathbb{E}[\boldsymbol{\theta}_{k-1}] - \boldsymbol{\theta}_\star\big)$ *and thus*

$$\mathbb{E}[\boldsymbol{\theta}_k] = \boldsymbol{\theta}_\star + \left(\prod_{\ell=1}^{k}(\mathbf{I}_d - \alpha_\ell \Sigma)\right)\big(\mathbb{E}[\boldsymbol{\theta}_0] - \boldsymbol{\theta}_\star\big). \tag{4.3.3}$$

The proof does not exploit the Gaussian design and only requires that $\mathbf{X}_i$ is centered and has covariance matrix Σ. The exogenous randomness induced by $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots$ disappears in the expected values but heavily influences the recursive expressions for the squared expectations.

Theorem 4.3.2. *Consider forward gradient descent* (4.2.2). *If* $A_k := \mathbb{E}\left[(\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star)(\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star)^\top\right]$, *then*

$$\begin{aligned} A_k =& (\mathbf{I}_d - \alpha_k \Sigma) A_{k-1} (\mathbf{I}_d - \alpha_k \Sigma) \\ &+ 3\alpha_k^2 \Sigma A_{k-1} \Sigma + 2\alpha_k^2 \mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top \Sigma (\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)\big]\Sigma + 2\alpha_k^2 \Sigma \\ &+ 2\alpha_k^2 \operatorname{tr}\big(\Sigma A_{k-1} \Sigma\big)\mathbf{I}_d + \alpha_k^2 \mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top \Sigma (\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)\big] \operatorname{tr}\big(\Sigma\big)\mathbf{I}_d \\ &+ \alpha_k^2 \operatorname{tr}(\Sigma)\mathbf{I}_d. \end{aligned}$$

Since A_k depends on $\boldsymbol{\theta}_k^2$, the fourth moments of the design vectors $\mathbf{X}_i$ and the exogenous random vectors $\boldsymbol{\xi}_k$ play a role in this equation.

The risk $\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big]$ is the trace of the matrix A_k. Setting

$$\kappa(\Sigma) := \frac{\|\Sigma\|_S}{\lambda_{\min}(\Sigma)}$$

for the condition number and building on Theorem 4.3.2, we can establish the following risk bound for forward gradient descent.

Theorem 4.3.3 (Mean squared error)**.** *Consider forward gradient descent* (4.2.2) *and assume that* Σ *is positive definite. For constant* $a > 2$, *choosing the learning rate*

$$\alpha_k = \frac{a\lambda_{\min}(\Sigma)}{k\lambda_{\min}^2(\Sigma) + a\|\Sigma\|_S^2(d+2)^2}, \quad k = 1, 2, \ldots, \tag{4.3.4}$$

yields

$$\begin{aligned}\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] \le \left(\frac{1 + a\kappa^2(\Sigma)(d+2)^2}{k + a\kappa^2(\Sigma)(d+2)^2}\right)^a \mathbb{E}\big[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_2^2\big] \\ + \frac{2ea\kappa(\Sigma)(d+2)^2}{\lambda_{\min}(\Sigma)(k + a\kappa^2(\Sigma)(d+2)^2)}.\end{aligned}$$

Alternatively, the upper bound of Theorem 4.3.3 can be written as

$$\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] \le \Big(1 - a^{-1}\lambda_{\min}(\Sigma)(k-1)\alpha_k\Big)^a \mathbb{E}\big[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_2^2\big] + 2e\kappa(\Sigma)(d+2)^2\alpha_k.$$

In the upper bound, the risk $\mathbb{E}\big[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_2^2\big]$ of the initial estimate $\boldsymbol{\theta}_0$ appears. A realistic scenario is that the entries of $\boldsymbol{\theta}_\star$ and $\boldsymbol{\theta}_0$ are all of order one. In this case, the inequality $\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_2^2 \le d\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_\infty^2$ shows that the risk of the initial estimate will scale with the number of parameters d. Taking $a = \log(d)$ (for $d \ge 8 > e^2$ such that $a > \log(e^2) = 2$), Theorem 4.3.3 implies that

$$\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] \lesssim d\left(\frac{d^2\log(d)}{k}\right)^{\log(d)} \mathbb{E}\big[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_\infty^2\big] + \frac{d^2\log(d)}{k}.$$

For $k_\star = e^2 d^2\log(d)$, $d^2\log(d)/k_\star = e^{-2}$ and $d(d^2\log(d)/k_\star)^{\log(d)} = 1/d$. Since $d > e^2$, this means that $d\big(d^2\log(d)/k_\star\big)^{\log(d)} < d^2\log(d)/k_\star$. Moreover, $k^{-\log(d)}$ tends faster to zero than k^{-1} as $k \to \infty$. So, for $k \ge k_\star = e^2d^2\log(d)$,

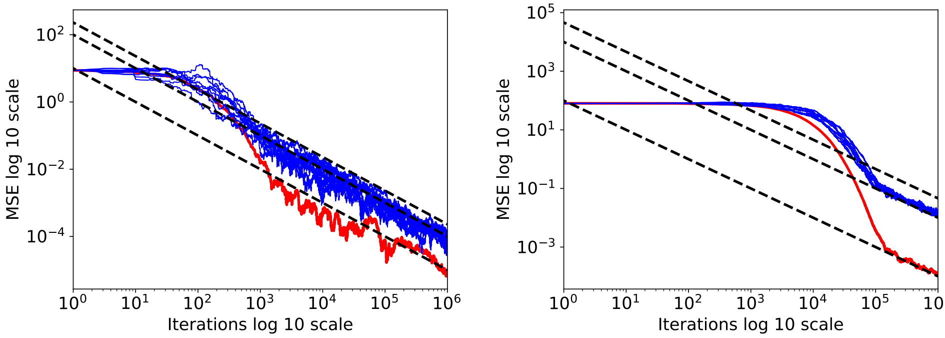
$$d\left(\frac{d^2\log(d)}{k}\right)^{\log(d)} \mathbb{E}\big[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_\infty^2\big] + \frac{d^2\log(d)}{k} \le \frac{d^2\log(d)}{k}\Big(1 + \mathbb{E}\big[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_\infty^2\big]\Big). \tag{4.3.5}$$

The rate for $k \ge e^2d^2\log(d)$ is thus $d^2\log(d)/k$. This means that forward gradient descent has dimension dependence $d^2\log(d)$. This is by a factor $d\log(d)$ worse than the minimax rate for the linear regression problem, [144, 63, 98]. In contrast, methods that have access to the gradient can achieve optimal dimension dependence in the rate, [114, 82]. The obtained convergence rate is in line with results for zero-order methods, which show that for convex optimization problems these methods have a higher dimension dependence, [41, 92, 105].

We believe that faster convergence rates are obtainable if the same datapoint is assessed several times. This means that each data point is used for several updates of the forward gradient $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_{k+1}\big(\nabla L(\boldsymbol{\theta}_k)\big)^\top \boldsymbol{\xi}_{k+1}\boldsymbol{\xi}_{k+1}$, for instance by running multiple epochs. However, in every iteration a new random direction $\boldsymbol{\xi}_{k+1}$ is sampled. We expect that if every data point is used $m \leq d$ times, one should be able to achieve the convergence rate $d^2/(km)$, up to some logarithmic terms. If this is true and if m is of the order of d, one could even recover the minimax rate d/k. Using the same datapoints multiple times induces additional dependence among the parameter updates. To deal with this dependence is the key challenge to establish the convergence rate $d^2/(km)$.

Assuming that the covariance matrix Σ is positive definite is standard for linear regression with random design [63, 98, 132].

For $k \gtrsim d^2$, the decrease of the learning rate α_k is of the order $1/k$, which is the standard choice [81, 55, 17]. A constant learning rate is used for Ruppert-Polyak averaging in [114, 55]. For least squares linear regression, it is possible to achieve (near) optimal convergence with a constant (universal) stepsize [6]. Conditions under which a constant (universal) stepsize in more general settings than linear least squares works or fails are investigated in [82].



(a) $d = 10$ (b) $d = 100$

Figure 4.3.1: Comparison of the MSE of forward gradient descent (blue) and SGD (red) for dimensions $d = 10$ and $d = 100$. The upper dashed line is $k \mapsto d^2 \log(d)/k$, the middle dashed line is $k \mapsto d^2/k$, and the lower dashed line is $k \mapsto d/k$.

In a small simulation study, we investigated whether there is a discrepancy between the derived convergence rates and the empirical decay of the risk. For dimensions $d = 10$ and $d = 100$, data according to (4.3.1) with $\Sigma = \mathbf{I}_d$ are generated. On these data, we run ten times weight perturbed forward gradient descent (4.2.2), and compare the mean squared errors (MSEs) to one realization of SGD (4.2.1). For all simulations of forward gradient descent and SGD, we use the same initialization $\boldsymbol{\theta}_0$, drawn from a $\mathcal{N}(0, \mathbf{I}_d)$ distribution, and the learning rate α_k specified in (4.3.4) with $a = \log(d)$. Thus, only the random perturbation vectors $\boldsymbol{\xi}_k$ in the forward gradient descent schemes differ across different runs. The outcomes are reported in Figure 4.3.1. For each of the 10+1 simulations, we report on a log-log scale the MSE for the first one million iterations. The upper dashed line gives the derived convergence rate $k \mapsto d^2 \log(d)/k$, the middle dashed line is d^2/k, and the lower dashed line is d/k. The ten paths from the ten forward gradient descent runs are shown in blue. The path from the SGD is displayed in red. We see three regimes. In the first regime, the risk remains nearly constant. For dimension $d = 100$, this is true up to the first ten thousand of iterations. Afterwards there is a sudden decrease of the risk. Eventually, for large number of iterations k, the MSE of forward gradient descent concentrates near the line $k \mapsto d^2/k$, while the MSE of SGD concentrates around $k \mapsto d/k$. This suggest that up to the $\log(d)$-factor, the derived theory does in fact describe the rate of the MSE. Equation (4.3.5) predicts that the rate $d^2 \log(d)/k$ will occur for $k \geq k_\star = e^2 d^2 \log(d)$. For $d = 10$, $k_\star \approx 1.7 \times 10^3$ and for $d = 100$, $k_\star \approx 3.4 \times 10^5$. Thus, in terms of orders of magnitude, there is a close agreement between theory and simulations.

Starting with a good initializer that lies already in the neighborhood of the true parameter, one can avoid the long burn-in time in the beginning. Otherwise, it remains an open problem, whether one can modify the procedure such that also for smaller values of k, the risk behaves more like $d^2 \log(d)/k$.

Python code is available on Github [24].

4.4 Proofs

Proof of Theorem 4.3.1. By (4.3.2) and the linear regression model $Y_{k-1} = \mathbf{X}_{k-1}^\top \boldsymbol{\theta}_\star + \epsilon_{k-1}$, we have

$$\begin{aligned} \nabla L(\boldsymbol{\theta}_{k-1}) &= -(Y_{k-1} - \mathbf{X}_{k-1}^\top \boldsymbol{\theta}_{k-1})\mathbf{X}_{k-1} \\ &= -(\mathbf{X}_{k-1}^\top (\boldsymbol{\theta}_\star - \boldsymbol{\theta}_{k-1}) + \epsilon_{k-1})\mathbf{X}_{k-1} \\ &= -\epsilon_{k-1}\mathbf{X}_{k-1} - \mathbf{X}_{k-1}\mathbf{X}_{k-1}^\top (\boldsymbol{\theta}_\star - \boldsymbol{\theta}_{k-1}). \end{aligned} \tag{4.4.1}$$

Since $\mathbb{E}[\mathbf{X}_{k-1}\mathbf{X}_{k-1}^\top] = \Sigma$, $\mathbb{E}[\epsilon_{k-1}] = 0$, and $\mathbf{X}_{k-1}, \epsilon_{k-1}, \boldsymbol{\theta}_{k-1}$ are jointly independent, we obtain

$$\begin{aligned} \mathbb{E}\big[\nabla L(\boldsymbol{\theta}_{k-1}) \,\big|\, \boldsymbol{\theta}_{k-1}\big] &= \mathbb{E}\big[-\epsilon_{k-1}\mathbf{X}_{k-1} - \mathbf{X}_{k-1}\mathbf{X}_{k-1}^\top (\boldsymbol{\theta}_\star - \boldsymbol{\theta}_{k-1}) \,\big|\, \boldsymbol{\theta}_{k-1}\big] \\ &= -\Sigma(\boldsymbol{\theta}_\star - \boldsymbol{\theta}_{k-1}). \end{aligned} \tag{4.4.2}$$

Combined with (4.2.3), we find

$$\mathbb{E}\big[\boldsymbol{\theta}_k\big] = \mathbb{E}\big[\boldsymbol{\theta}_{k-1}\big] - \alpha_k \mathbb{E}\big[\nabla L(\boldsymbol{\theta}_{k-1})\big] = \mathbb{E}\big[\boldsymbol{\theta}_{k-1}\big] + \alpha_k \Sigma \mathbb{E}\big[\boldsymbol{\theta}_\star - \boldsymbol{\theta}_{k-1}\big].$$

The true parameter $\boldsymbol{\theta}_\star$ is deterministic. Subtracting $\boldsymbol{\theta}_\star$ on both sides, yields the claimed identity $\mathbb{E}[\boldsymbol{\theta}_k] - \boldsymbol{\theta}_\star = \big(\mathbf{I}_d - \alpha_k \Sigma\big)\big(\mathbb{E}[\boldsymbol{\theta}_{k-1}] - \boldsymbol{\theta}_\star\big)$.

□

4.4.1 Proof of Theorem 4.3.2

Lemma 4.4.1. *If* $\mathbf{Z} \sim \mathcal{N}(0, \Gamma)$ *is a d-dimensional random vector and* $\mathbf{U}$ *is a d-dimensional random vector that is independent of* $\mathbf{Z}$*, then*

$$\mathbb{E}\big[(\mathbf{U}^\top \mathbf{Z})^2 \mathbf{Z}\mathbf{Z}^\top\big] = 2\Gamma \mathbb{E}\big[\mathbf{U}\mathbf{U}^\top\big]\Gamma + \mathbb{E}\big[\mathbf{U}^\top \Gamma \mathbf{U}\big]\Gamma.$$

Proof. Because $\mathbf{U}$ and $\mathbf{Z}$ are independent, the (i,j)-th entry of the $d \times d$ matrix $\mathbb{E}\big[(\mathbf{U}^\top \mathbf{Z})^2 \mathbf{Z}\mathbf{Z}^\top\big]$ is

$$\sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell U_m\big]\mathbb{E}\big[Z_\ell Z_m Z_i Z_j\big].$$

Since $\mathbf{Z} \sim \mathcal{N}(0, \Gamma)$,

$$\mathbb{E}\big[Z_\ell Z_m Z_i Z_j\big] = \Gamma_{\ell,m}\Gamma_{i,j} + \Gamma_{\ell,i}\Gamma_{m,j} + \Gamma_{\ell,j}\Gamma_{m,i},$$

see for instance the example at the end of Section 2 in [143]. Thus

$$\sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell U_m\big]\mathbb{E}\big[Z_\ell Z_m Z_i Z_j\big] = \sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell U_m\big]\big(\Gamma_{\ell,m}\Gamma_{i,j} + \Gamma_{\ell,i}\Gamma_{m,j} + \Gamma_{\ell,j}\Gamma_{m,i}\big).$$

Because of

$$\sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell U_m\big]\Gamma_{\ell,m}\Gamma_{i,j} = \sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell \Gamma_{\ell,m} U_m\big]\Gamma_{i,j} = \mathbb{E}\left[\mathbf{U}^\top \Gamma \mathbf{U} \Gamma_{i,j}\right],$$

$$\sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell U_m\big]\Gamma_{\ell,i}\Gamma_{m,j} = \sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell \Gamma_{\ell,i} U_m \Gamma_{m,j}\big] = \mathbb{E}\left[\big(\mathbf{U}^\top \Gamma\big)_i \big(\mathbf{U}^\top \Gamma\big)_j\right],$$

and

$$\sum_{\ell,m=1}^{d} \mathbb{E}\big[U_\ell U_m\big]\Gamma_{\ell,j}\Gamma_{m,i} = \sum_{\ell,m=1}^{d} \mathbb{E}\big[U_m \Gamma_{m,i} U_\ell \Gamma_{\ell,j}\big] = \mathbb{E}\left[\big(\mathbf{U}^\top \Gamma\big)_i \big(\mathbf{U}^\top \Gamma\big)_j\right],$$

the (i,j)-th entry of the matrix $\mathbb{E}\left[(\mathbf{U}^\top \mathbf{Z})^2\mathbf{Z}\mathbf{Z}^\top\right]$ is

$$2\mathbb{E}\left[\big(\mathbf{U}^\top\Gamma\big)_i\big(\mathbf{U}^\top\Gamma\big)_j\right]+\mathbb{E}\left[\mathbf{U}^\top\Gamma\mathbf{U}\Gamma_{i,j}\right].$$

For a vector $\mathbf{a}=(a_1,\dots,a_d)^\top$, the scalar a_ia_j is the (i,j)-th entry of the matrix $\mathbf{a}\mathbf{a}^\top$. Combined with the previous display, the result follows. □

Proof of Theorem 4.3.2. As Theorem 4.3.2 only involves one update step, we can simplify the notation by dropping the index k and analyzing $\boldsymbol{\theta}''=\boldsymbol{\theta}'-\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}$ for one data point $(\mathbf{X},Y)$ and independent $\boldsymbol{\xi}\sim\mathcal{N}(0,I_d)$. With $A':=\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\big]$ and $A'':=\mathbb{E}\big[(\boldsymbol{\theta}''-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}''-\boldsymbol{\theta}_\star)^\top\big]$, we then have to prove that

$$\begin{aligned}A''=&(\mathbf{I}_d-\alpha\Sigma)A'(\mathbf{I}_d-\alpha\Sigma)+3\alpha^2\Sigma A'\Sigma+2\alpha^2\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)\big]\Sigma+2\alpha^2\Sigma\\&+2\alpha^2\operatorname{tr}\big(\Sigma A'\Sigma\big)\mathbf{I}_d+\alpha^2\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)\big]\operatorname{tr}\big(\Sigma\big)\mathbf{I}_d+\alpha^2\operatorname{tr}(\Sigma)\mathbf{I}_d.\end{aligned}$$

Substituting the update rule (4.2.2) in A_k gives by the linearity of the transpose that

$$\begin{aligned}A''&=\mathbb{E}\big[(\boldsymbol{\theta}''-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}''-\boldsymbol{\theta}_\star)^\top\big]\\&=\mathbb{E}\left[\Big(\boldsymbol{\theta}'-\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}-\boldsymbol{\theta}_\star\Big)\Big(\boldsymbol{\theta}'-\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}-\boldsymbol{\theta}_\star\Big)^\top\right]\\&=A'-\alpha\mathbb{E}\left[\Big(\boldsymbol{\theta}-\boldsymbol{\theta}_\star\Big)\Big(\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)^\top\right]-\alpha\mathbb{E}\left[\Big(\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)\Big(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star\Big)^\top\right]\\&\quad+\mathbb{E}\left[\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)^\top\right].\end{aligned}\tag{4.4.3}$$

First, consider the terms with the minus sign in the above expression. The random vector $\boldsymbol{\xi}$ is independent of all other randomness and hence $\mathbb{E}_{\boldsymbol{\xi}}\Big[\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big]=\nabla L(\boldsymbol{\theta}')$. Moreover, together with (4.4.2),

$$\begin{aligned}\mathbb{E}\left[\Big(\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)\Big(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star\Big)^\top\,\middle|\,\boldsymbol{\theta}'\right]&=\mathbb{E}\big[\nabla L(\boldsymbol{\theta}')\,|\,\boldsymbol{\theta}'\big](\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\\&=\Sigma(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top.\end{aligned}$$

Taking the transpose and tower rule, we find

$$\begin{aligned}&-\alpha\mathbb{E}\left[\Big(\boldsymbol{\theta}-\boldsymbol{\theta}_\star\Big)\Big(\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)^\top\right]-\alpha\mathbb{E}\left[\Big(\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)\Big(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star\Big)^\top\right]\\&=-\alpha\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\big]\Sigma-\alpha\Sigma\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\big].\end{aligned}\tag{4.4.4}$$

In a next step, we derive an expression for $\mathbb{E}\Big[\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)^\top\Big]$. Since $\boldsymbol{\xi}\sim\mathcal{N}(0,\mathbf{I}_d)$ is independent of $\nabla L(\boldsymbol{\theta}')$ we can apply Lemma 4.4.1 to derive

$$\begin{aligned}&\mathbb{E}\Big[\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)^\top\Big]\\ &=\alpha^2\mathbb{E}\Big[\Big(\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\Big)^2\boldsymbol{\xi}\boldsymbol{\xi}^\top\Big]\\ &=2\alpha^2\mathbb{E}\Big[\big(\nabla L(\boldsymbol{\theta}')\big)\big(\nabla L(\boldsymbol{\theta}')\big)^\top\Big]+\alpha^2\mathbb{E}\Big[\big(\nabla L(\boldsymbol{\theta}')\big)^\top\big(\nabla L(\boldsymbol{\theta}')\big)\Big]\mathbf{I}_d\\ &=2\alpha^2\mathbb{E}\Big[\big(\nabla L(\boldsymbol{\theta}')\big)\big(\nabla L(\boldsymbol{\theta}')\big)^\top\Big]+\alpha^2\operatorname{tr}\Big(\mathbb{E}\Big[\big(\nabla L(\boldsymbol{\theta}')\big)\big(\nabla L(\boldsymbol{\theta}')\big)^\top\Big]\Big)\mathbf{I}_d.\end{aligned}\tag{4.4.5}$$

Arguing as for (4.4.1) gives $\nabla L(\boldsymbol{\theta}')=-\epsilon\mathbf{X}-\mathbf{X}\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')$ and this yields

$$\mathbb{E}\Big[\big(\nabla L(\boldsymbol{\theta}')\big)\big(\nabla L(\boldsymbol{\theta}')\big)^\top\Big]=\mathbb{E}\Big[\mathbb{E}_\epsilon\Big[\big(\epsilon\mathbf{X}+\mathbf{X}\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')\big)\big(\epsilon\mathbf{X}+\mathbf{X}\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')\big)^\top\Big]\Big].$$

Because ϵ has mean zero and variance one and is independent of $(\mathbf{X},\boldsymbol{\theta}')$, we conclude that

$$\begin{aligned}\mathbb{E}\Big[\big(\nabla L(\boldsymbol{\theta}')\big)\big(\nabla L(\boldsymbol{\theta}')\big)^\top\Big]&=\mathbb{E}\Big[\big(\mathbf{X}\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')\big)\big(\mathbf{X}\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')\big)^\top+\mathbf{X}\mathbf{X}^\top\Big]\\ &=\mathbb{E}\Big[\big(\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')\big)^2\mathbf{X}\mathbf{X}^\top\Big]+\Sigma,\end{aligned}\tag{4.4.6}$$

where for the last equality we used that $\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')$ is a scalar and that $\mathbf{X}\sim\mathcal{N}(0,\Sigma)$. Since $\mathbf{X}\sim\mathcal{N}(0,\Sigma)$ is independent of $\boldsymbol{\theta}'$ we get by Lemma 4.4.1 that

$$\mathbb{E}\Big[\big(\mathbf{X}^\top(\boldsymbol{\theta}_\star-\boldsymbol{\theta}')\big)^2\mathbf{X}\mathbf{X}^\top\Big]=2\Sigma\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\big]\Sigma+\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)\big]\Sigma.$$

Substituting this in (4.4.6) and (4.4.5) yields

$$\begin{aligned}&\mathbb{E}\Big[\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)\Big(\alpha\big(\nabla L(\boldsymbol{\theta}')\big)^\top\boldsymbol{\xi}\boldsymbol{\xi}\Big)^\top\Big]\\ &=4\alpha^2\Sigma\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\big]\Sigma+2\alpha^2\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)\big]\Sigma+2\alpha^2\Sigma\\ &+2\alpha^2\operatorname{tr}\Big(\Sigma\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\big]\Sigma\Big)\mathbf{I}_d+\alpha^2\operatorname{tr}\Big(\mathbb{E}\big[(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}'-\boldsymbol{\theta}_\star)\big]\Sigma\Big)\mathbf{I}_d\\ &+\alpha^2\operatorname{tr}(\Sigma)\mathbf{I}_d.\end{aligned}\tag{4.4.7}$$

Combining (4.4.3) with (4.4.4) and (4.4.7) yields the statement of the theorem. □

4.4.2 Proof of Theorem 4.3.3

For two vectors $\mathbf{u}, \mathbf{v}$ of the same length, $\operatorname{tr}(\mathbf{u}\mathbf{v}^\top) = \mathbf{u}^\top\mathbf{v}$. Thus, $\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] = \operatorname{tr}\left(\mathbb{E}\big[(\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star)(\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star)^\top\big]\right)$. Together with Theorem 4.3.2, $\operatorname{tr}(\mathbf{I}_d) = d$ and $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ for square matrices A and B of the same size, this yields

$$\begin{aligned}
\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] &= \operatorname{tr}\Big((\mathbf{I}_d - \alpha_k\Sigma)\mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top\big](\mathbf{I}_d - \alpha_k\Sigma)\Big) \\
&\quad + 3\alpha_k^2 \operatorname{tr}\Big(\Sigma\mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top\big]\Sigma\Big) \\
&\quad + 2\alpha_k^2 \operatorname{tr}\Big(\mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)\big]\Sigma\Big) + 2\alpha_k^2 \operatorname{tr}\big(\Sigma\big) \\
&\quad + 2\alpha_k^2 \operatorname{tr}\Big(\Sigma\mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top\big]\Sigma\Big) \operatorname{tr}\big(\mathbf{I}_d\big) \\
&\quad + \alpha_k^2\mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)\big] \operatorname{tr}\big(\Sigma\big) \operatorname{tr}\big(\mathbf{I}_d\big) \\
&\quad + \alpha_k^2 \operatorname{tr}(\Sigma) \operatorname{tr}\big(\mathbf{I}_d\big) \\
&= \mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top(\mathbf{I}_d - 2\alpha_k\Sigma)^\top(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)\big] \\
&\quad + 2(d+2)\alpha_k^2 \operatorname{tr}\Big(\Sigma\mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top\big]\Sigma\Big) \\
&\quad + (d+2)\alpha_k^2\Big(\mathbb{E}\big[(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)^\top\Sigma(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)\big] \operatorname{tr}\big(\Sigma\big) + \operatorname{tr}\big(\Sigma\big)\Big).
\end{aligned} \tag{4.4.8}$$

If λ is an eigenvalue of Σ then $(1 - 2\alpha_k\lambda)$ is an eigenvalue of $\mathbf{I}_d - 2\alpha_k\Sigma$. By assumption, $0 < \alpha_k \leq \lambda_{\min}(\Sigma)/\big(2\|\Sigma\|_S^2\big) \leq 1/\big(2\lambda_{\max}(\Sigma)\big)$ and therefore the matrix $\mathbf{I}_d - 2\alpha_k\Sigma$ is positive semi-definite and $(1 - 2\alpha_k\lambda_{\min}(\Sigma))$ is the largest eigenvalue.

For a positive semi-definite matrix A and a vector $\mathbf{v}$, the min-max theorem states that $\mathbf{v}^\top A\mathbf{v} \leq \lambda_{\max}(A)\|\mathbf{v}\|_2^2 = \|A\|_S\|\mathbf{v}\|_2^2$. Using that for a vector $\mathbf{x}$ it holds that $\operatorname{tr}(\mathbf{x}\mathbf{x}^\top) = \mathbf{x}^\top\mathbf{x}$, with $\mathbf{x} = \Sigma(\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star)$ in (4.4.8) and applying $\mathbf{v}^\top A\mathbf{v} \leq \|A\|_S\|\mathbf{v}\|_2^2$ with $\mathbf{v} = \boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star$ and $A \in \{\Sigma, \mathbf{I}_d - 2\alpha_k\Sigma, \Sigma^2\}$, yields

$$\begin{aligned}
&\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] \leq \big(1 - 2\alpha_k\lambda_{\min}(\Sigma)\big)\mathbb{E}\big[\|\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star\|_2^2\big] \\
&\quad + (d+2)\alpha_k^2\bigg(\operatorname{tr}(\Sigma)\|\Sigma\|_S\mathbb{E}\big[\|\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star\|_2^2\big] + 2\|\Sigma\|_S^2\mathbb{E}\big[\|\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star\|_2^2\big] + \operatorname{tr}(\Sigma)\bigg).
\end{aligned}$$

The spectral norm of a positive semi-definite matrix is equal to the largest eigenvalue and so $\operatorname{tr}(\Sigma) = \sum_{i=1}^d \lambda_i \leq d\lambda_{\max} = d\|\Sigma\|_S$. Therefore,

$$\begin{aligned}
\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] &\leq \Big(1 - 2\alpha_k\lambda_{\min}(\Sigma) + \|\Sigma\|_S^2(d+2)^2\alpha_k^2\Big)\mathbb{E}\big[\|\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star\|_2^2\big] \\
&\quad + \|\Sigma\|_S(d+2)^2\alpha_k^2.
\end{aligned}$$

Using that $\alpha_k \leq \lambda_{\min}(\Sigma)/\big(\|\Sigma\|_S^2(d+2)^2\big)$ yields

$$\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] \leq \big(1 - \alpha_k \lambda_{\min}(\Sigma)\big)\mathbb{E}\big[\|\boldsymbol{\theta}_{k-1} - \boldsymbol{\theta}_\star\|_2^2\big] + \|\Sigma\|_S(d+2)^2\alpha_k^2.$$

Rewritten in non-recursive, we obtain

$$\begin{aligned}\mathbb{E}\big[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2\big] \leq &\mathbb{E}\big[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_2^2\big] \prod_{\ell=1}^{k} \big(1 - \alpha_\ell \lambda_{\min}(\Sigma)\big) \\ &+ \|\Sigma\|_S(d+2)^2 \sum_{m=0}^{k-1} \alpha_{k-m}^2 \prod_{\ell=k-m+1}^{k} \big(1 - \alpha_\ell \lambda_{\min}(\Sigma)\big),\end{aligned} \tag{4.4.9}$$

where we use the convention that the (empty) product over zero terms is assigned the value 1. For ease of notation define $c_d := a\kappa^2(\Sigma)(d+2)^2$, with condition number $\kappa(\Sigma) = \|\Sigma\|_S/\lambda_{\min}(\Sigma)$. From the definition of α_k, (4.3.4), it follows that $\alpha_k = \frac{a}{\lambda_{\min}(\Sigma)} \cdot \frac{1}{k+c_d}$. Using that for all real numbers x it holds that $1 + x \leq e^x$, we get that for all integers $k^* < k$,

$$\prod_{\ell=k^*}^{k} \big(1 - \alpha_\ell \lambda_{\min}(\Sigma)\big) \leq \exp\Bigg(- \lambda_{\min}(\Sigma) \sum_{\ell=k^*}^{k} \alpha_\ell \Bigg) = \exp\Bigg(- a \sum_{\ell=k^*}^{k} \frac{1}{\ell + c_d} \Bigg). \tag{4.4.10}$$

The function $x \mapsto 1/(x + c)$ is monotone decreasing for $x > 0$ and $c \geq 0$ and thus,

$$\begin{aligned}\sum_{\ell=k^*}^{k} \frac{1}{\ell + c_d} &\geq \sum_{\ell=k^*}^{k} \int_{\ell}^{\ell+1} \frac{1}{x + c_d} dx \\ &= \int_{k^*}^{k+1} \frac{1}{x + c_d} dx \\ &= \log(k + 1 + c_d) - \log(k^* + c_d) \\ &= \log\Big(\frac{k + 1 + c_d}{k^* + c_d}\Big).\end{aligned} \tag{4.4.11}$$

Using (4.4.10) and (4.4.11) with $k^* = 1$ gives

$$\prod_{\ell=1}^{k} \big(1 - \alpha_\ell \lambda_{\min}(\Sigma)\big) \leq \exp\Bigg(- a \log\Big(\frac{k + 1 + c_d}{1 + c_d}\Big)\Bigg) = \Bigg(\frac{1 + c_d}{k + 1 + c_d}\Bigg)^a. \tag{4.4.12}$$

Using (4.4.10) and (4.4.11) with $k^* = k - m + 1$ gives

$$\begin{aligned}
&\sum_{m=0}^{k-1} \alpha_{k-m}^2 \prod_{\ell=k-m+1}^{k} \big(1 - \alpha_\ell \lambda_{\min}(\Sigma)\big) \\
&\le \frac{a^2}{\lambda_{\min}^2(\Sigma)} \sum_{m=0}^{k-1} \frac{1}{\big((k-m)+c_d\big)^2} \left(\frac{k-m+1+c_d}{k+1+c_d}\right)^a \\
&= \frac{a^2}{\lambda_{\min}^2(\Sigma)(k+1+c_d)^a} \sum_{m=0}^{k-1} \frac{\big(k-m+1+c_d\big)^a}{\big((k-m)+c_d\big)^2} \\
&= \frac{a^2}{\lambda_{\min}^2(\Sigma)(k+1+c_d)^a} \sum_{m=1}^{k} \frac{\big(m+1+c_d\big)^a}{\big(m+c_d\big)^2}.
\end{aligned} \tag{4.4.13}$$

Observe that $c_d = a\kappa^2(\Sigma)(d+2)^2 \ge a$. This gives us that $c_d + 1 \le (1 + 1/a)c_d$ and thus $m + 1 + c_d \le (1 + 1/a)(m + c_d)$. For all real numbers x, $(1 + x) \le e^x$ and thus $(1 + 1/a)^a \le e$. Therefore,

$$\sum_{m=1}^{k} \frac{\big(m+1+c_d\big)^a}{\big(m+c_d\big)^2} \le e \sum_{m=1}^{k} \big(m+c_d\big)^{a-2}. \tag{4.4.14}$$

For $p > 0$, the function $x \mapsto (x + c)^p$ is monotone increasing for $x, c > 0$, Hence,

$$\begin{aligned}
\sum_{\ell=1}^{k} (\ell + c)^p &\le \sum_{\ell=1}^{k} \int_{\ell}^{\ell+1} (x+c)^p dx \\
&= \int_{1}^{k+1} (x+c)^p dx \\
&= \frac{(k+1+c)^{p+1}}{p+1} - \frac{(1+c)^{p+1}}{p+1} \\
&\le \frac{(k+1+c)^{p+1}}{p+1}.
\end{aligned}$$

Since $a > 2$, we can apply this with $p = a - 2 > 0$ to find

$$e \sum_{m=1}^{k} \big(m+c_d\big)^{a-2} \le e \frac{(k+1+c_d)^{a-1}}{a-1}.$$

Combining (4.4.9), (4.4.12), (4.4.13) and (4.4.14) finally gives

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_\star\|_2^2] \leq & \left(\frac{1 + a\kappa^2(\Sigma)(d+2)^2}{k+1+a\kappa^2(\Sigma)(d+2)^2}\right)^a \mathbb{E}[\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_\star\|_2^2] \\ & + \frac{ea^2\kappa(\Sigma)(d+2)^2}{\lambda_{\min}(\Sigma)\big(a-1\big)\big(k+1+a\kappa^2(\Sigma)(d+2)^2\big)}.\end{aligned}$$

Using that $0 < a/(a-1) < 2$ for $a > 2$, now yields the result. □

Chapter 5

General discussion

In this thesis, risk bounds for deep learning have been established in various settings. The central aim was to use statistical theory to obtain new insights into the performance of deep neural networks. Chapter 2 showed that deep neural networks can achieve optimal convergence rates under the (truncated) cross-entropy risk for the conditional class probabilities in the classification model. Furthermore, this chapter includes approaches to deal with the unboundedness of the cross-entropy loss for conditional class probabilities near zero. The used approaches are truncation and the small-value bound assumption. This last bound controls the probability that the conditional probabilities are close to zero. In Chapter 3 a method was studied that transforms the unsupervised density estimation problem into a supervised regression problem. In this way, convergence rates were obtained using existing results for regression. These rates show that deep neural networks can exploit a compositional structure to partly circumvent the curse of dimensionality. Furthermore, it was demonstrated that different existing density models indeed satisfy the compositional structure assumption. Chapter 4 considered an optimization method motivated by biological networks: forward gradient descent. It was shown that the extra randomness in forward gradient descent leads to a convergence rate in the linear regression model that is a dimension-dependent factor $d\log(d)$ slower than the optimal rate that can be achieved by gradient descent.

These findings rely on certain assumptions. This chapter discusses some of these underlying assumptions in more detail, relates them to existing literature on neural networks and discusses whether and how these assumptions can be adapted to extend the results in this thesis.

5.1 Statistical theory and training of neural networks

In Chapters 2 and 3 the risk bounds depend on the assumption that the estimator has empirical risk close to the risk of an empirical risk minimizer. The analysis of empirical risk minimizers without specifying how to obtain them is standard in statistical literature on risk bounds for deep neural networks. Examples of this approach include [67, 74, 76, 134]. In practice it is non-trivial to compute such an estimator. One additional issue is that the constraints on the deep neural network classes in Chapters 2 and 3 do not necessarily match the network structures considered in the deep learning literature. Most importantly, overparametrized neural networks are studied in practice because they can be trained relatively easily and successfully by simple gradient methods [10, 15]. But such overparametrized neural networks do not match the neural network classes studied in this thesis.

On the other hand, Chapter 4 considers forward gradient descent in the linear regression model. In this case the training method is the focus of the analysis, including an explicit (theoretical) learning rate. When the relevant properties of the covariance matrix Σ are known, the theory provides all the information required to run the method. This in contrast to the training of neural networks, as done in the simulation study in Chapter 3, where various (training) parameters must be chosen before the neural networks can be trained properly. The limitation here is that the results of Chapter 4 are for the linear regression model, a setting that is much easier to deal with than deep neural networks. There exists (optimization) literature on the complexity of stochastic gradient descent and zero-order methods that expands results for those methods to more general strongly convex-optimization problems, [115, 133]. This suggests the possibility for further research extending the results in Chapter 4 to general convex problems. The key challenge is to deal simultaneously with the randomness from the data and the additional randomness introduced by forward gradient descent. As training deep neural networks is a non-convex optimization problem it remains unclear if it is feasible to extend the analysis to the deep neural networks considered in Chapters 2 and 3.

5.2 Model assumptions

In this thesis various assumptions on the target function are imposed. In Chapter 2 it is assumed that the conditional class probabilities are β-Hölder smooth. In Chapter 3 it is assumed that the densities have a compositional structure, where each function in the composition is in some Hölder-smoothness class. The main motivation behind the choice for these smoothness assumptions is that this makes comparison with existing risk bounds in the literature possible, as convergence of the risk under these assumptions has been widely studied. The compositional structure in Chapter 3, as well as the possible inclusion of a compositional structure as discussed in Chapter 2, are motivated by existing results for regression [62, 72, 11, 127, 75]. In these works, it is shown that deep neural networks can circumvent the curse of dimensionality under compositional structure assumptions. This provides a possible explanation for the observed good performance on high-dimensional input problems of deep neural networks in practice.

For image classification there exists a related assumption, the hierarchical max-pooling model considered in [74, 76]. This compositional model is tailored to the image classification task in combination with convolutional neural networks. The principal idea behind this model is that the question: "contains the image a prespecified object?", can be answered by estimating the probability that this is true for subparts of the image and then taking the maximum of the probabilities over the subparts.

A different kind of model assumption is based on the observation that in many practical datasets the data seem to lie around a low dimensional manifold. In [103] it is shown for the regression problem that if the data are scattered around a lower dimensional manifold, then deep neural networks can exploit this to obtain convergence rates that depend on the intrinsic manifold dimension instead of the full dimension of the input space. For this result it is also assumed that the regression function is Hölder-smooth. This paper includes a numerical estimation of the intrinsic dimension of the MNIST and CIFAR-10 benchmark datasets, showing that these datasets indeed have an intrinsic dimension that is much smaller than their full dimension. An assumption that is closer related to the composition structure assumption in this thesis is the assumption of local low dimensionality studied in [73]. The idea of the local low dimensionality assumption is that the function locally only depends on very few of its components. Under this assumption it is shown that in the regression problem the bounds depend on the local dimensionality instead of the full input dimension. These works [103, 73] suggest that it should be possible to combine the idea of the data lying around a lower dimensional manifold with the results for the classification and density estimation models studied in this thesis. How to combine the composition and manifold assumptions in a manner that is realistic for practical datasets and the exact effects of such a combination on the risk bounds is an avenue for further research.

Bibliography

[1] Aas, K., Czado, C., Frigessi, A., and Bakken, H. Pair-copula constructions of multiple dependence. *Insurance Math. Econom. 44*, 2 (2009), 182–198.

[2] Ahle, T. D. Sharp and simple bounds for the raw moments of the binomial and Poisson distributions. *Statist. Probab. Lett. 182* (2022), Paper No. 109306, 5.

[3] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. Deep speech 2 : End-to-end speech recognition in English and Mandarin. In *Proceedings of The 33rd International Conference on Machine Learning* (New York, New York, USA, 20–22 Jun 2016), M. F. Balcan and K. Q. Weinberger, Eds., vol. 48 of *Proceedings of Machine Learning Research*, PMLR, pp. 173–182.

[4] Anderson, G. D., Vamanamurthy, M. K., and Vuorinen, M. Inequalities for quasiconformal mappings in space. *Pacific J. Math. 160*, 1 (1993), 1–18.

[5] Audibert, J.-Y., and Tsybakov, A. B. Fast learning rates for plug-in classifiers. *Ann. Statist. 35*, 2 (2007), 608–633.

[6] Bach, F., and Moulines, E. Non-strongly-convex smooth stochastic approximation with convergence rate O (1/n). In *Advances in Neural Information*

Processing Systems (2013), C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26, Curran Associates, Inc.

[7] Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inform. Theory 39*, 3 (1993), 930–945.

[8] Barron, A. R. Approximation and estimation bounds for artificial neural networks. *Machine learning 14*, 1 (1994), 115–133.

[9] Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *J. Amer. Statist. Assoc. 101*, 473 (2006), 138–156.

[10] Bartlett, P. L., Montanari, A., and Rakhlin, A. Deep learning: a statistical viewpoint. *Acta Numerica 30* (2021), 87–201.

[11] Bauer, B., and Kohler, M. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Ann. Statist. 47*, 4 (2019), 2261–2285.

[12] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research 18*, 153 (2018), 1–43.

[13] Baydin, A. G., Pearlmutter, B. A., Syme, D., Wood, F., and Torr, P. Gradients without backpropagation. *arXiv preprint arXiv:2202.08587* (2022).

[14] Bedford, T., and Cooke, R. M. Probability density decomposition for conditionally dependent random variables modeled by vines. *Ann. Math. Artif. Intell. 32*, 1-4 (2001), 245–268.

[15] Belkin, M., Rakhlin, A., and Tsybakov, A. B. Does data interpolation contradict statistical optimality? In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (16–18 Apr 2019), K. Chaudhuri and M. Sugiyama, Eds., vol. 89 of *Proceedings of Machine Learning Research*, PMLR, pp. 1611–1619.

[16] Bennett, G. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association 57*, 297 (1962), 33–45.

[17] Benveniste, A., Métivier, M., and Priouret, P. *Adaptive algorithms and stochastic approximations*, vol. 22 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1990. Translated from the French by Stephen S. Wilson.

[18] Berend, D., and Tassa, T. Improved bounds on Bell numbers and on moments of sums of random variables. *Probab. Math. Statist. 30*, 2 (2010), 185–205.

[19] Besag, J. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Statist. Soc. Ser. B 36* (1974), 192–236.

[20] Birgé, L., and Massart, P. Minimum contrast estimators on sieves: exponential bounds and rates of convergence. *Bernoulli 4*, 3 (1998), 329–375.

[21] Bishop, C. M. *Pattern recognition and machine learning.* Information Science and Statistics. Springer, New York, 2006.

[22] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. End to end learning for self-driving cars. *arXiv e-prints* (2016), arXiv:1604.07316.

[23] Bos, T., and Schmidt-Hieber, J. Simulation-code: A supervised deep learning method for nonparametric density estimation. `https://github.com/Bostjm/Simulation-code`, Apr. 2023.

[24] Bos, T., and Schmidt-Hieber, J. Simulation code: Convergence guarantees for forward gradient descent in the linear regression model. `https://github.com/Bostjm/SimulationCodeForwardGradient`, Jan. 2024.

[25] Boucheron, S., Lugosi, G., and Massart, P. *Concentration inequalities.* Oxford University Press, Oxford, 2013.

[26] Brechmann, E. C., Czado, C., and Aas, K. Truncated regular vines in high dimensions with application to financial data. *Canad. J. Statist. 40*, 1 (2012), 68–85.

[27] Breiman, L. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Trans. Inform. Theory 39*, 3 (1993), 999–1013.

[28] Cherubini, U., Luciano, E., and Vecchiato, W. *Copula methods in finance.* Wiley Finance Series. John Wiley & Sons, Ltd., Chichester, 2004.

[29] Choromanska, A., Henaff, M., Mathieu, M., Ben Arous, G., and LeCun, Y. The loss durfaces of multilayer networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* (San Diego, California, USA, 09–12 May 2015), G. Lebanon and S. V. N. Vishwanathan, Eds., vol. 38 of *Proceedings of Machine Learning Research*, PMLR, pp. 192–204.

[30] Cireşan, D., Meier, U., and Schmidhuber, J. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 3642–3649.

[31] Clara, G., Langer, S., and Schmidt-Hieber, J. Dropout regularization versus ℓ_2-penalization in the linear model. *arXiv e-prints* (2023), arXiv:2306.10529.

[32] Conn, A. R., Scheinberg, K., and Vicente, L. N. *Introduction to derivative-free optimization*, vol. 8 of *MPS/SIAM Series on Optimization.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Programming Society (MPS), Philadelphia, PA, 2009.

[33] Crick, F. The recent excitement about neural networks. *Nature 337* (1989), 129–132.

[34] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems 2*, 4 (1989), 303–314.

[35] Czado, C. *Analyzing dependent data with vine copulas*, vol. 222 of *Lecture Notes in Statistics.* Springer, Cham, 2019. A practical guide with R.

[36] Czado, C., and Nagler, T. Vine copula based modeling. *Annu. Rev. Stat. Appl. 9* (2022), 453–477.

[37] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems* (2014), Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc.

[38] Devroye, L., Györfi, L., and Lugosi, G. *A probabilistic theory of pattern recognition*, vol. 31 of *Applications of Mathematics (New York).* Springer-Verlag, New York, 1996.

[39] Drouet Mari, D., and Kotz, S. *Correlation and dependence.* Imperial College Press, London; distributed by World Scientific Publishing Co., Inc., River Edge, NJ, 2001.

[40] Dua, D., and Graff, C. UCI machine learning repository, 2017.

[41] Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Wibisono, A. Optimal rates for zero-order convex optimization: the power of two function evaluations. *IEEE Trans. Inform. Theory 61*, 5 (2015), 2788–2806.

[42] DUDLEY, R. M. A course on empirical processes. In *École d'été de probabilités de Saint-Flour, XII—1982*, vol. 1097 of *Lecture Notes in Math.* Springer, Berlin, 1984, pp. 1–142.

[43] DURANTE, F., AND SEMPI, C. Copula theory: an introduction. In *Copula theory and its applications*, vol. 198 of *Lect. Notes Stat. Proc.* Springer, Heidelberg, 2010, pp. 3–31.

[44] EFROMOVICH, S. *Nonparametric curve estimation.* Springer Series in Statistics. Springer-Verlag, New York, 1999.

[45] EFRON, B., AND TIBSHIRANI, R. Using specially designed exponential families for density estimation. *Ann. Statist. 24*, 6 (1996), 2431–2461.

[46] FUNAHASHI, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural Networks 2*, 3 (1989), 183–192.

[47] GÄNSSLER, P. *Empirical processes*, vol. 3 of *Institute of Mathematical Statistics Lecture Notes—Monograph Series.* Institute of Mathematical Statistics, Hayward, CA, 1983.

[48] GAO, Z., AND HASTIE, T. LinCDE: conditional density estimation via Lindsey's method. *J. Mach. Learn. Res. 23* (2022), Paper No. [52], 55.

[49] GLOROT, X., AND BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010), JMLR Workshop and Conference Proceedings, pp. 249–256.

[50] GLOROT, X., BORDES, A., AND BENGIO, Y. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (Fort Lauderdale, FL, USA, 11–13 Apr 2011), G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15 of *Proceedings of Machine Learning Research*, PMLR, pp. 315–323.

[51] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[52] GREENSPAN, H., VAN GINNEKEN, B., AND SUMMERS, R. M. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging 35*, 5 (2016), 1153–1159.

[53] GROSSBERG, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science 11*, 1 (1987), 23–63.

[54] Györfi, L., Kohler, M., Krzyżak, A., and Walk, H. *A distribution-free theory of nonparametric regression.* Springer Series in Statistics. Springer-Verlag, New York, 2002.

[55] Györfi, L., and Walk, H. On the averaged stochastic approximation for linear regression. *SIAM J. Control Optim. 34*, 1 (1996), 31–61.

[56] Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning*, second ed. Springer Series in Statistics. Springer, New York, 2009. Data mining, inference, and prediction.

[57] Haussler, D., and Opper, M. Mutual information, metric entropy and cumulative relative entropy risk. *Ann. Statist. 25*, 6 (1997), 2451–2492.

[58] He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 1026–1034.

[59] Heckerman, E., and Nathwani, N. Toward normative expert systems: Part II. Probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in medicine 31*, 02 (1992), 106–116.

[60] Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput. 18*, 7 (2006), 1527–1554.

[61] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks 2*, 5 (1989), 359–366.

[62] Horowitz, J. L., and Mammen, E. Rate-optimal estimation for a general class of nonparametric regression models with unknown link functions. *Ann. Statist. 35*, 6 (2007), 2589–2619.

[63] Hsu, D., Kakade, S. M., and Zhang, T. Random design analysis of ridge regression. *Found. Comput. Math. 14*, 3 (2014), 569–600.

[64] Johnson, N. L., and Kotz, S. On some generalized Farlie-Gumbel-Morgenstern distributions. II. Regression, correlation and further generalizations. *Comm. Statist.—Theory Methods A6*, 6 (1977), 485–496.

[65] Jones, L. K. A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Ann. Statist. 20*, 1 (1992), 608–613.

[66] Kantorovitz, S. *Several real variables.* Springer Undergraduate Mathematics Series. Springer, [Cham], 2016.

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

[68] Kingma, D. P., and Welling, M. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning 12*, 4 (2019), 307–392.

[69] Kingman, J. F. C. *Poisson processes*, vol. 3 of *Oxford Studies in Probability.* The Clarendon Press, Oxford University Press, New York, 1993. Oxford Science Publications.

[70] Kirshner, S. Learning with tree-averaged densities and distributions. In *Advances in Neural Information Processing Systems* (2007), J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, Curran Associates, Inc.

[71] Kohler, M., and Krzyżak, A. Adaptive regression estimation with multilayer feedforward neural networks. *J. Nonparametr. Stat. 17*, 8 (2005), 891–913.

[72] Kohler, M., and Krzyżak, A. Nonparametric regression based on hierarchical interaction models. *IEEE Trans. Inform. Theory 63*, 3 (2017), 1620–1630.

[73] Kohler, M., Krzyżak, A., and Langer, S. Estimation of a function of low local dimensionality by deep neural networks. *IEEE Trans. Inform. Theory 68*, 6 (2022), 4032–4042.

[74] Kohler, M., Krzyzak, A., and Walter, B. On the rate of convergence of image classifiers based on convolutional neural networks. *arXiv preprint arXiv:2003.01526* (2020).

[75] Kohler, M., and Langer, S. On the rate of convergence of fully connected very deep neural network regression estimates. *arXiv e-prints* (2019), arXiv:1908.11133.

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

[77] Kohler, M., and Langer, S. On the rate of convergence of fully connected deep neural network regression estimates. *Ann. Statist. 49*, 4 (2021), 2231–2249.

[78] Koller, D., and Friedman, N. *Probabilistic graphical models.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2009.

[79] Korb, K. B., and Nicholson, A. E. *Bayesian artificial intelligence.* Chapman & Hall/CRC Computer Science and Data Analysis Series. Chapman & Hall/CRC, Boca Raton, FL, 2004.

[80] Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25.* Curran Associates, Inc., 2012, pp. 1097–1105.

[81] Kushner, H. J., and Yin, G. G. *Stochastic approximation and recursive algorithms and applications*, second ed., vol. 35 of *Applications of Mathematics (New York).* Springer-Verlag, New York, 2003.

[82] Lakshminarayanan, C., and Szepesvari, C. Linear stochastic approximation: How far does constant step-size and iterate averaging go? In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* (09–11 Apr 2018), A. Storkey and F. Perez-Cruz, Eds., vol. 84 of *Proceedings of Machine Learning Research*, PMLR, pp. 1347–1355.

[83] Larson, J., Menickelly, M., and Wild, S. M. Derivative-free optimization methods. *Acta Numer. 28* (2019), 287–404.

[84] Launay, J., Poli, I., Boniface, F., and Krzakala, F. Direct feedback alignment scales to modern deep learning tasks and architectures. In *Advances in Neural Information Processing Systems* (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., pp. 9346–9360.

[85] Lauritzen, S. L. *Graphical models*, vol. 17 of *Oxford Statistical Science Series.* The Clarendon Press, Oxford University Press, New York, 1996. Oxford Science Publications.

[86] Le Cun, Y. Learning process in an asymmetric threshold network. In *Disordered systems and biological organization*, vol. 20. Springer, 1986, pp. 233–240.

[87] Leibig, C., Allken, V., Ayhan, M. S., Berens, P., and Wahl, S. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports 7*, 1 (2017), 1–14.

[88] Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications 7*, 1 (2016), 13276.

[89] Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. Backpropagation and the brain. *Nature Reviews Neuroscience 21* (2020), 335–346.

[90] Lindsey, J. K. Comparison of probability distributions. *J. Roy. Statist. Soc. Ser. B 36* (1974), 38–47.

[91] Lindsey, J. K. Construction and comparison of statistical models. *J. Roy. Statist. Soc. Ser. B 36* (1974), 418–425.

[92] Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero III, A. O., and Varshney, P. K. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine 37*, 5 (2020), 43–54.

[93] Loader, C. *Local regression and likelihood.* Statistics and Computing. Springer-Verlag, New York, 1999.

[94] Mammen, E., and Tsybakov, A. B. Smooth discrimination analysis. *Ann. Statist. 27*, 6 (1999), 1808–1829.

[95] McCulloch, W. S., and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics 5* (1943), 115–133.

[96] Mörters, P., and Peres, Y. *Brownian motion*, vol. 30 of *Cambridge Series in Statistical and Probabilistic Mathematics.* Cambridge University Press, Cambridge, 2010.

[97] Moschopoulos, P., and Staniswalis, J. G. Estimation given conditionals from an exponential family. *Amer. Statist. 48*, 4 (1994), 271–275.

[98] Mourtada, J. Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices. *Ann. Statist. 50*, 4 (2022), 2157–2178.

[99] Murphy, K. P. *Machine Learning: a Probabilistic Perspective.* MIT press, 2012.

[100] Nadaraya, E. A. On estimating regression. *Theory of Probability & Its Applications 9*, 1 (1964), 141–142.

[101] Nagler, T., and Czado, C. Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *J. Multivariate Anal. 151* (2016), 69–89.

[102] Nair, V., and Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (2010), pp. 807–814.

[103] Nakada, R., and Imaizumi, M. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *J. Mach. Learn. Res. 21* (2020), Paper No. 174, 38.

[104] Nelsen, R. B. *An introduction to copulas*, second ed. Springer Series in Statistics. Springer, New York, 2006.

[105] Nesterov, Y., and Spokoiny, V. Random gradient-free minimization of convex functions. *Found. Comput. Math. 17*, 2 (2017), 527–566.

[106] Nøkland, A. Direct feedback alignment provides learning in deep neural networks. In *Advances in Neural Information Processing Systems* (2016), D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc.

[107] Nussbaum, M. Asymptotic equivalence of density estimation and Gaussian white noise. *Ann. Statist. 24*, 6 (1996), 2399–2430.

[108] Parzen, E. On estimation of a probability density function and mode. *Ann. Math. Statist. 33* (1962), 1065–1076.

[109] Pearl, J. *Causality*, second ed. Cambridge University Press, Cambridge, 2009.

[110] Petersen, P., and Voigtlaender, F. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks 108* (2018), 296–330.

[111] Petersen, P., and Voigtlaender, F. Optimal learning of high-dimensional classification problems using deep neural networks. *arXiv preprint arXiv:2112.12555* (2021).

[112] Pinelis, I. L'Hospital type results for monotonicity, with applications. *JIPAM. J. Inequal. Pure Appl. Math. 3*, 1 (2002), Article 5, 5.

[113] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing 14*, 5 (2017), 503–519.

[114] Polyak, B. T., and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim. 30*, 4 (1992), 838–855.

[115] Rakhlin, A., Shamir, O., and Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647* (2011).

[116] Ray, K., and Schmidt-Hieber, J. The Le Cam distance between density estimation, Poisson processes and Gaussian white noise. *Math. Stat. Learn. 1*, 2 (2018), 101–170.

[117] Ren, M., Kornblith, S., Liao, R., and Hinton, G. Scaling forward gradient with local losses. *arXiv preprint arXiv:2210.03310* (2022).

[118] Resnick, S. *Adventures in stochastic processes.* Birkhäuser Boston, Inc., Boston, MA, 1992.

[119] Rice, J. A. *Mathematical Statistics and Data Analysis (Third Edition).* Brooks/Cole,Cengage Learning, 2007.

[120] Robbins, H. A remark on Stirling's formula. *Amer. Math. Monthly 62* (1955), 26–29.

[121] Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review 65*, 6 (1958), 386–408.

[122] Rosenblatt, F. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms.* Spartan books Washington, DC, 1962.

[123] Rosenblatt, M. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist. 27* (1956), 832–837.

[124] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature 323* (1986), 533–536.

[125] Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *ICLR* (2014).

[126] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks 61* (2015), 85–117.

[127] Schmidt-Hieber, J. Nonparametric regression using deep neural networks with ReLU activation function. *Ann. Statist. 48*, 4 (2020), 1875–1897.

[128] Schmidt-Hieber, J. Interpreting learning in biological neural networks as zero-order optimization method. *arXiv preprint arXiv:2301.11777* (2023).

[129] Schmidt-Hieber, J., and Koolen, W. Hebbian learning inspired estimation of the linear regression parameters from queries. *arXiv preprint* (2023).

[130] Schmidt-Hieber, J., and Zamolodtchikov, P. Local convergence rates of the least squares estimator with applications to transfer learning. *arXiv e-prints* (2022), arXiv:2204.05003.

[131] Scott, D. W. *Multivariate density estimation*, second ed. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, 2015. Theory, practice, and visualization.

[132] Shaffer, J. P. The Gauss-Markov theorem and random regressors. *Amer. Statist. 45*, 4 (1991), 269–273.

[133] Shamir, O. On the complexity of bandit and derivative-free stochastic convex optimization. In *Proceedings of the 26th Annual Conference on Learning Theory* (Princeton, NJ, USA, 12–14 Jun 2013), S. Shalev-Shwartz and I. Steinwart, Eds., vol. 30 of *Proceedings of Machine Learning Research*, PMLR, pp. 3–24.

[134] Shen, G., Jiao, Y., Lin, Y., and Huang, J. Non-asymptotic excess risk bounds for classification with deep convolutional neural networks. *arXiv preprint arXiv:2105.00292* (2021).

[135] Spall, J. C. *Introduction to stochastic search and optimization.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2003. Estimation, simulation, and control.

[136] Stöber, J., Joe, H., and Czado, C. Simplified pair copula constructions—limitations and extensions. *J. Multivariate Anal. 119* (2013), 101–118.

[137] Stone, C. J. Optimal rates of convergence for nonparametric estimators. *Ann. Statist. 8*, 6 (1980), 1348–1360.

[138] Stone, C. J. Optimal global rates of convergence for nonparametric regression. *Ann. Statist. 10*, 4 (1982), 1040–1053.

[139] Stone, C. J. Additive regression and other nonparametric models. *Ann. Statist. 13*, 2 (1985), 689–705.

[140] Tarigan, B., and van de Geer, S. A. A moment bound for multi-hinge classifiers. *J. Mach. Learn. Res. 9* (2008), 2171–2185.

[141] TELGARSKY, M. Benefits of depth in neural networks. In *29th Annual Conference on Learning Theory* (Columbia University, New York, New York, USA, 23–26 Jun 2016), V. Feldman, A. Rakhlin, and O. Shamir, Eds., vol. 49 of *Proceedings of Machine Learning Research*, PMLR, pp. 1517–1539.

[142] TRAPPENBERG, T. P. *Fundamentals of Computational Neuroscience: Third Edition.* Oxford University Press, 12 2022.

[143] TRIANTAFYLLOPOULOS, K. On the central moments of the multidimensional Gaussian distribution. *Math. Sci. 28*, 2 (2003), 125–128.

[144] TSYBAKOV, A. B. Optimal rates of aggregation. In *Learning Theory and Kernel Machines* (Berlin, Heidelberg, 2003), B. Schölkopf and M. K. Warmuth, Eds., Springer Berlin Heidelberg, pp. 303–313.

[145] TSYBAKOV, A. B. Optimal aggregation of classifiers in statistical learning. *Ann. Statist. 32*, 1 (2004), 135–166.

[146] TSYBAKOV, A. B. *Introduction to Nonparametric Estimation.* Springer Series in Statistics. Springer New York, 2008.

[147] VAN DE GEER, S. A. *Empirical Processes in M-Estimation.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2000.

[148] VAN DER VAART, A. W., AND WELLNER, J. A. *Weak convergence and empirical processes.* Springer Series in Statistics. Springer-Verlag, New York, 1996. With applications to statistics.

[149] VAN ERVEN, T., AND HARREMOËS, P. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inform. Theory 60*, 7 (2014), 3797–3820.

[150] VAPNIK, V. N. *The nature of statistical learning theory*, second ed. Statistics for Engineering and Information Science. Springer-Verlag, New York, 2000.

[151] WAND, M. P., AND JONES, M. C. *Kernel smoothing*, vol. 60 of *Monographs on Statistics and Applied Probability.* Chapman and Hall, Ltd., London, 1995.

[152] WASSERMAN, L. *All of statistics.* Springer Texts in Statistics. Springer-Verlag, New York, 2004.

[153] WASSERMAN, L. *All of nonparametric statistics.* Springer Texts in Statistics. Springer, New York, 2006.

[154] WATSON, G. S. Smooth regression analysis. *Sankhyā Ser. A 26* (1964), 359–372.

[155] Whittington, J. C. R., and Bogacz, R. An approximation of the error backpropagation algorithm in a predictive coding network with local Hebbian synaptic plasticity. *Neural Comput. 29*, 5 (2017), 1229–1262.

[156] Widrow, B., and Hoff, M. E. Adaptive switching circuits. In *IRE WESCON convention record* (1960), vol. 4, New York, pp. 96–104.

[157] Wong, W. H., and Severini, T. A. On maximum likelihood estimation in infinite-dimensional parameter spaces. *Ann. Statist. 19*, 2 (1991), 603–632.

[158] Wong, W. H., and Shen, X. Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *Ann. Statist. 23*, 2 (1995), 339–362.

[159] Yang, Y., and Barron, A. Information-theoretic determination of minimax rates of convergence. *Ann. Statist. 27*, 5 (1999), 1564–1599.

[160] Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Networks 94* (2017), 103–114.

Summary

In this thesis, deep learning is studied from a statistical perspective. Bounds are obtained for the worst-case risk of neural network estimators in the classification, density estimation and linear regression model. Special attention is given to the role of the input dimension since in practice, neural networks have shown promising results for high dimensional input settings.

In Chapter 1 an introduction to nonparametric statistics and deep learning is provided. Chapter 2 considers the problem of estimating the conditional class probabilities in the classification model. This is done using the cross-entropy loss. This loss can be derived from the likelihood of the conditional class probabilities. One challenge with this loss is that it becomes unbounded near zero. To deal with this a truncated version of the risk is introduced. Convergence rates are obtained for a neural network estimator under this truncated risk. These rates depend on a new criterion, the small value bound, controlling the probability that the conditional class probabilities are near zero. The truncated risk provides an upper bound on the risk related to the Hellinger loss. This connection implies that the obtained convergence rates are optimal for conditional class probabilities that are bounded away from zero.

Chapter 3 considers density estimation. A two-step procedure is proposed for this problem. Since density estimation is an unsupervised learning problem, the first step is to transform it into a supervised regression problem: An undersmoothed kernel density estimator is constructed using half of the data. This estimator is then used to generate fake response variables for the other half of the data. In the second step, a deep neural network is fitted to the supervised learning problem obtained in the first step. As the obtained supervised data-pairs are dependent on each other, standard theory for i.i.d. data cannot be applied directly. Using a Poissonization argument, an oracle inequality for this setting is derived. Based on existing approximation results, convergence rates for the two-step procedure for the squared risk are obtained. These rates show that if the underlying density has a compositional structure, then the proposed procedure achieves faster convergence rates. A simulation study explores the two-step method for finite sample sizes. This simulation study uses structured

multivariate densities from the Bayesian network and copula models.

Forward gradient descent is studied in Chapter 4. This is a biologically motivated alternative for gradient descent. Forward gradient descent has additional randomness compared to gradient descent. Furthermore, it can be seen as an intermediate step between gradient descent and derivative free zero-order methods. We prove convergence rates for this method in the linear regression model with random design. The obtained rates are a dimension dependence factor $d \log(d)$ slower than the rates achieved by gradient descent. However, the obtained rates are the same as the rates achieved by zero-order methods.

Samenvatting

In dit proefschrift wordt diep leren (in het Engels: deep learning) bestudeerd vanuit een statistisch perspectief. Bovengrenzen worden bewezen voor het risico in het slechtste geval van neurale netwerk schatters in de classificatie, kansdichtheidsschatting en lineaire regressie modellen. Neurale netwerken hebben in de praktijk veel belovende resultaten behaald voor hoog-dimensionale input problemen. Speciale aandacht wordt daarom gegeven aan de rol van de dimensie.

In hoofdstuk 1 worden non-parametrische statistiek en diep leren geïntroduceerd. Hoofdstuk 2 behandelt het probleem van het schatten van de voorwaardelijke categorielidmaatschapskansen in het classificatie model. Hiervoor wordt de kruisentropie verliesfunctie (in het Engels: Cross-Entropy loss) gebruikt. Deze verliesfunctie kan worden afgeleid van de aannemelijkheidsfunctie van de conditionele categorielidmaatschapskansen. Deze verliesfunctie kan oneindig groot worden dichtbij nul. Extra voorzichtigheid is daarom geboden voor conditionele categorielidmaatschapskansen dichtbij nul. We introduceren een afgeknotte versie van de verliesfunctie om de onbegrensdheid in de buurt van nul te verhelpen. Convergentiesnelheden voor het risico gebaseerd op deze afgeknotte verliesfunctie van een neuraal netwerk schatter worden bewezen. Deze snelheden hangen af van een nieuw criterium, de kleine waarde grens (in het Engels: small value bound). Dit criterium bepaalt hoe snel de conditionele categorielidmaatschapskansen naar nul mogen gaan. Het door ons gebruikte afgeknotte risico begrenst het risico gebaseerd op de Hellinger verliesfunctie van boven. Uit deze verwantschap volgt dat de bewezen convergentiesnelheden optimaal zijn als de conditionele categorielidmaatschapskansen wegbegrensd zijn van nul.

Hoofdstuk 3 gaat over kansdichtheidsschatting, een leertaak zonder voorbeeld/leraar (in het Engels: unsupervised learning). Een tweestapsmethode voor dit probleem wordt geïntroduceerd. Eerst wordt het probleem getransformeerd in een regressie probleem: een leertaak met voorbeeld/leraar (in het Engels: supervised learning). Een kernel kansdichtheidsschatter (in het Engels: kernel density estimator) met een te kleine brandbreedte wordt geconstrueerd op basis van de helft van de data. Deze

schatter wordt vervolgens gebruikt om uitkomstvariabelen te genereren voor de andere helft van de data. In de tweede stap van de methode wordt een diep neuraal netwerk getraind op de data die gegenereerd is in de eerste stap. Standaard theorie voor onafhankelijke identiek verdeelde (in het Engels: independent identically distributed, i.i.d.) variabelen is niet meteen toepasbaar. De uitkomstvariabelen zijn namelijk onderling afhankelijk van elkaar. Met behulp van een argumentatie gebaseerd op de Poisson-verdeling wordt een orakelongelijkheid bewezen. Convergentiesnelheden voor deze tweestapsmethode voor het kwadratische risico worden bewezen met behulp van bestaande benaderingsresultaten voor diepe neurale netwerken. Deze snelheden tonen aan dat als de kansdichtheid een samengestelde structuur heeft, dan is de tweestapsmethode in staat om hiervan gebruik te maken om een hogere convergentiesnelheid te bereiken. Een verkennende simulatiestudie bestudeerd de tweestapsmethode voor eindige data hoeveelheden. De simulatiestudie gebruikt hiervoor gestructureerde multivariate kansdichtheden uit de Bayesiaanse netwerk en copula modellen.

Voorwaartse gradiënt afdaling (in het Engels: Forward gradient descent) wordt bestudeerd in hoofdstuk 4. Dit is een biologisch gemotiveerd alternatief voor gradiënt afdaling. Deze methode bevat extra ruis ten opzichte van gradiënt afdaling. Het kan gezien worden als een tussenstap tussen gradiënt afdaling en afgeleide vrije nulde-order methoden (in het Engels: zero-order methods). Convergentiesnelheden voor het kwadratenrisico van deze methode worden bewezen in het lineaire regressie model met gerandomiseerd ontwerp. Deze snelheden zijn een dimensie afhankelijke factor $d\log(d)$ trager dan de convergentiesnelheden die bereikt worden door gradiënt afdaling. Echter, de bewezen snelheden zijn gelijk aan de convergentiesnelheden behaalt door nulde-order methoden.

Acknowledgements

In 2018 begon ik met mijn onderzoek voor mijn PhD. Onderzoek dat al een aanvang genomen had met mijn masterscriptie. Op deze plaats wil ik de mensen bedanken die hebben bijgedragen aan dit traject. Johannes, bedankt voor de mogelijkheid die je mij gegeven hebt om dit onderzoek te doen. Mijn interesse in statiek werd gewekt door het door jouw gegeven en ontwikkelde vak Mathematical Statistics. Bedankt voor al de jaren van begeleiding. Bedankt voor jouw bereidheid om rekening te houden met mijn beperkingen en mee te denken over praktische oplossingen.

Peter, bedankt dat je bereid was om de rol van begeleider op je te nemen nadat bleek dat er nog een begeleider met aanstelling in Leiden nodig was. Bedankt ook voor het delen van je kennis en ervaringen. Iets dat je al deed toen we af en toe het kantoor deelden, lang voordat je mijn begeleider werd.

Valentina, thank you for our very nice cooperation on the course Mathematical Statistics. Together we managed to deal with the practical challenges related to classes and exams.

Emiel en Marieke, bedankt voor het meedenken tijdens onze periodieke overleggen en voor het uitzoeken van hoe al de regels nu precies in elkaar zitten.

Lisanne, Angela, Kees en Inge mijn coaches vanuit Stichting studeren en werken op maat, bedankt voor jullie begeleiding bij de praktische zaken, communicatieve uitdagingen en stress die komen kijken bij het hebben van werk.

Wies, bedankt dat je mij met jouw taalgevoel geholpen hebt om de leesbaarheid van mijn teksten te vergroten.

Inge en Meike, bedankt voor jullie steun en het delen van jullie ervaringen, ook op die momenten dat een PhD niet alleen maar leuk was. Bedankt ook voor al de gezelligheid en ontspanning die er altijd is als we samen zijn.

Pap en mam, bedankt voor jullie warmte en steun elke dag weer. Bedankt ook voor al jullie inspanningen om de uitdagende puzzel te helpen leggen die onderwijs of werk soms voor mij is.

Curriculum Vitae

Thijs Bos was born in Bergschenhoek, the Netherlands, on 18 September 1991. From 2004 to 2008 he attended the Auris College Rotterdam, where he obtained his VMBO-TL diploma. From 2008 to 2011 he attended the VAVO Rijnmond College in Capelle aan den IJssel, where he obtained his HAVO diploma in 2010 and his VWO diploma in 2011. In 2011 he continued his education at Leiden University. There he completed his bachelor Wiskunde in 2015. He continued his studies there and obtained his master Applied Mathematics (cum laude) in 2018.

In 2018 Thijs started working as PhD candidate in Leiden under supervision of Prof.dr. A.J. Schmidt-Hieber. During his work as PhD candidate, he was teaching assistant for the course Mathematical Statistics for 5 years. Furthermore, he co-supervised one master thesis.

List of publications

- Thijs Bos and Johannes Schmidt-Hieber (2022). Convergence rates of deep ReLU networks for multiclass classification. In Electronic Journal of Statistics **16**, 2724-2773. (Chapter 2)
- Thijs Bos and Johannes Schmidt-Hieber (2023). A supervised deep learning method for nonparametric density estimation. Preprint arXiv:2306.10471. (Chapter 3)
- Thijs Bos and Johannes Schmidt-Hieber. Convergence guarantees for forward gradient descent in the linear regression model. To appear in Journal of Statistical Planning and Inference, Volume **233**. (Chapter 4)