

Six Lectures on Linearized Neural Networks

Theodor Misiakiewicz¹ and Andrea Montanari²

August 24, 2023

¹Department of Statistics, Stanford University

²Department of Electrical Engineering and Department of Statistics, Stanford University

Contents

1	Models and motivations	1
1.1	Setting	1
1.2	The optimization question	3
1.3	The linear regime	4
1.4	Linearization of two-layer networks	6
1.5	Outline of this tutorial	7
2	Linear regression under feature concentration assumptions	9
2.1	Setting and sharp characterization	9
2.2	Non-Gaussian covariates	13
2.3	An example: Analysis of a latent space model	13
2.4	Bounds and benign overfitting	15
3	Kernel ridge regression in high dimension	19
3.1	Infinite-width limit and kernel ridge regression	19
3.2	Test error of KRR in the polynomial high-dimensional regime	23
3.3	Diagonalization of inner-product kernels on the sphere	26
3.4	Proof sketch	28
4	Random features	31
4.1	The random feature model	31
4.2	Test error in the polynomial high-dimensional regime	32
4.3	Double descent and proportional asymptotics	34
4.4	Polynomial asymptotics: Proof sketch	35
5	Neural tangent features	39
5.1	Finite-width neural tangent model	39
5.2	Approximation by infinite-width KRR: Kernel matrix	41
5.3	Approximation by infinite-width KRR: Test error	43
5.4	Proof sketch: Kernel concentration	45
6	Why stop being lazy (and how)	47
6.1	Lazy training fails on ridge functions	47
6.2	Non-lazy infinite-width networks (a.k.a. mean field)	49
6.3	Learning ridge functions in mean field: basics	52

A Summary of notations	55
B Literature survey	57
References	73

Acknowledgments

These lecture notes are based on two series of lectures given by Andrea Montanari, first at the “Deep Learning Theory Summer School” at Princeton from July 27th to August 4th 2021, and then at the summer school “Statistical Physics & Machine Learning”, that took place at Les Houches School of Physics in France from 4th to 29th July 2022. Theodor Misiakiewicz was Teaching Assistant to AM’s lectures at the “Deep Learning Theory Summer School”. We thank Boris Hanin for organizing the summer school in Princeton, and Florent Krzakala and Lenka Zdeborová from EPFL for organizing the summer school in Les Houches.

AM was supported by the NSF through award DMS-2031883, the Simons Foundation through Award 814639 for the Collaboration on the Theoretical Foundations of Deep Learning, the NSF grant CCF-2006489 and the ONR grant N00014-18-1-2729

Chapter 1

Models and motivations

This tutorial examines what can be learnt about the behavior of multi-layer neural networks from the analysis of linear models. While there are important gaps between neural networks and their linear counterparts, many useful lessons can be learnt by studying the latter.

A few preliminary remarks, before diving into the math:

- We will not assume specific background in machine learning, let alone neural networks. On the other hand, we will assume some graduate-level mathematics, in particular probability theory (however, we will refer to the literature for complete proofs.)
- Some of the notations that are used throughout the text will be summarized in Appendix A.
- We will keep bibliographic references in the main text to a minimum. A short guide to the literature is given in Appendix B.

This chapter is devoted to describing the correspondence between nonlinear and linear models via the so-called neural tangent model.

1.1 Setting

We will focus on supervised learning. We are given data $\{(y_i, \mathbf{x}_i)\}_{i \leq n} \sim_{iid} \mathbb{P}$ where $\mathbf{x}_i \in \mathbb{R}^d$ is a vector of covariates, $y_i \in \mathbb{R}$ is a response or label, and \mathbb{P} a probability distribution on $\mathbb{R} \times \mathbb{R}^d$. We denote the space of such probability distributions by $\mathcal{P}(\mathbb{R} \times \mathbb{R}^d)$.

We want to learn a model, that is a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ that, given a new covariate vector \mathbf{x}_{new} allows to predict the corresponding response y_{new} . The quality of a prediction is measured by the test error $R(f; \mathbb{P}) = \mathbb{E}\{\ell(y_{\text{new}}, f(\mathbf{x}_{\text{new}}))\}$ where $\ell : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a loss function. We will focus on the simplest choice for the latter, square loss. Namely

$$R(f; \mathbb{P}) = \mathbb{E}\{(y_{\text{new}} - f(\mathbf{x}_{\text{new}}))^2\}, \quad (y_{\text{new}}, \mathbf{x}_{\text{new}}) \sim \mathbb{P}. \quad (1.1.1)$$

Depending on the context, we will modify the notation for the arguments of R . In particular, we will be mostly interested in parametric models $f(\mathbf{x}) = f(\mathbf{x}; \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^p$ is a vector of parameters (e.g. the network weights), and therefore the test error can be thought of as a function of these parameters. With a slight abuse of notation, we might therefore write

$$R(\boldsymbol{\theta}; \mathbb{P}) = \mathbb{E}\{(y_{\text{new}} - f(\mathbf{x}_{\text{new}}; \boldsymbol{\theta}))^2\}. \quad (1.1.2)$$

Often we will drop the argument \mathbb{P} or replace it by a proxy. Also, it is sometimes convenient to subtract from the test error the minimum error achieved by any predictor f (also known as the ‘Bayes risk’). The result is the so called ‘excess risk’

$$\begin{aligned} R_{\text{exc}}(f; \mathbb{P}) &= R(f; \mathbb{P}) - R_{\mathbb{B}}(\mathbb{P}) \\ &:= \mathbb{E}\{(y_{\text{new}} - f(\mathbf{x}_{\text{new}}))^2\} - \inf_{f_0: \mathbb{R}^d \rightarrow \mathbb{R}} \mathbb{E}\{(y_{\text{new}} - f_0(\mathbf{x}_{\text{new}}))^2\}, \end{aligned} \quad (1.1.3)$$

where the infimum is taken over all measurable functions. In the case of square loss treated here, the Bayes risk is just the conditional variance: $R_{\mathbb{B}}(\mathbb{P}) = \mathbb{E}\{(y - \mathbb{E}(y|\mathbf{x}))^2\}$.

The main approach to learn the parametric model $f(\cdot; \boldsymbol{\theta})$ consists in minimizing the empirical risk

$$\hat{R}_n(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^n (y_i - f(\mathbf{x}_i; \boldsymbol{\theta}))^2. \quad (1.1.4)$$

Modern machine learning systems attempt to achieve approximate minimization of this objective via first order methods. This term is used to refer to algorithms that access the cost function $\hat{R}_n(\boldsymbol{\theta})$ only by obtaining its gradient $\nabla \hat{R}_n(\boldsymbol{\theta}^i)$ and value $\hat{R}_n(\boldsymbol{\theta}^i)$ at query points $\boldsymbol{\theta}^0, \dots, \boldsymbol{\theta}^k$. The next query point $\boldsymbol{\theta}^{k+1}$ is computed from this information.

The simplest example is, of course, gradient descent (GD):

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \varepsilon_k \mathbf{S} \nabla_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}^k).$$

Here $\mathbf{S} \in \mathbb{R}^{p \times p}$ is a scaling matrix that allows us to choose different step sizes for different groups of parameters. In practice stochastic gradient descent (SGD) is preferred for a number of reasons. In its simplest implementation SGD takes a gradient step with respect to the loss incurred on a single, randomly chosen, sample

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + 2\tilde{\varepsilon}_k (y_{I(k)} - f(\mathbf{x}_{I(k)}; \boldsymbol{\theta}^k)) \mathbf{S} \nabla_{\boldsymbol{\theta}} f(\mathbf{x}_{I(k)}; \boldsymbol{\theta}^k).$$

Both GD and SGD can sometimes be well approximated by gradient flow (GF) for the sake of analysis. GF corresponds to the vanishing stepsize limit, and operates in continuous time

$$\dot{\boldsymbol{\theta}}(t) = -\mathbf{S} \nabla_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}(t)).$$

Extra care should be paid—in general—when working with such continuous time dynamics, as they do not necessarily correspond to practical algorithms. However in the case of GD and SGD the correspondence is relatively straightforward: these algorithms are often well approximated by GF for reasonable choices of the stepsize. (Namely, stepsize that is an inverse polynomial in the dimension d .)

For future reference, it is useful to note that the empirical risk (1.1.4) only depends on the model $f(\mathbf{x}; \boldsymbol{\theta})$ through its evaluation at the n datapoints:

$$f_n(\boldsymbol{\theta}) = (f(\mathbf{x}_1; \boldsymbol{\theta}), f(\mathbf{x}_2; \boldsymbol{\theta}), \dots, f(\mathbf{x}_n; \boldsymbol{\theta}))^\top \quad (1.1.5)$$

This define the *evaluation map* $f_n: \mathbb{R}^p \rightarrow \mathbb{R}^n$. The empirical risk can then be written as

$$\hat{R}_n(\boldsymbol{\theta}) := \frac{1}{n} \|\mathbf{y} - f_n(\boldsymbol{\theta})\|_2^2, \quad (1.1.6)$$

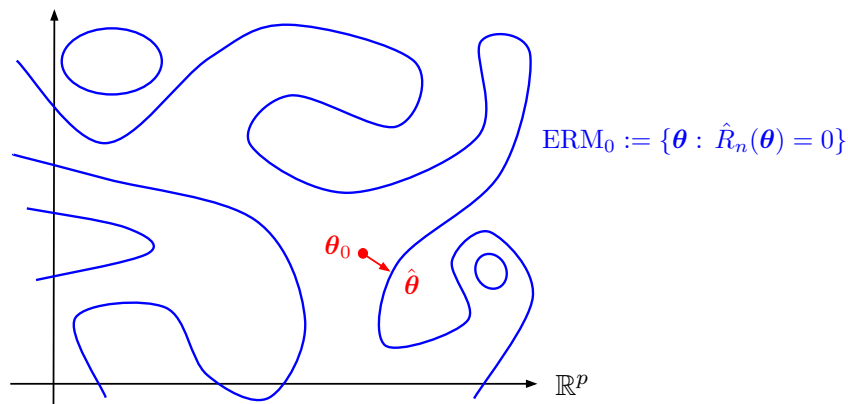


Figure 1.1

with $\mathbf{y} = (y_1, \dots, y_n)^\top$.

These lectures are mainly devoted to two-layer (one-hidden layer) networks. In this case the parametric model reads

$$f(\mathbf{x}; \boldsymbol{\theta}) = \alpha \sum_{i=1}^N a_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle), \quad \boldsymbol{\theta} = (a_1, \dots, a_N; \mathbf{w}_1, \dots, \mathbf{w}_N) \in \mathbb{R}^{N(d+1)}, \quad (1.1.7)$$

where the activation function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is fixed. (The scaling factor α will be useful in the following.)

1.2 The optimization question

For nonlinear models such as the two-layer network (1.1.7), the empirical risk $\hat{R}_n(\boldsymbol{\theta})$ of Eq. (1.1.4) is highly non-convex. Despite this, GD or SGD (and their variants) appear to be able to optimize the empirical risk of real neural networks to near global optimality.

This leads to the following

Optimization question: How is it possible that simple first order methods optimize the empirical risk of neural networks to near global optimality?

Over the last few years, an hypothesis has emerged to explain this puzzle: tractability of empirical risk minimization is due to the fact that the network is overparametrized. Let us briefly describe informally this scenario, which is essentially conjectural at the moment. Since the number of parameters p is larger than the sample size n , we expect there to be many global empirical risk minimizers that achieve zero risk. We denote the set of such global minimizers by

$$\text{ERM}_0 = \left\{ \boldsymbol{\theta} \in \mathbb{R}^p : f_n(\boldsymbol{\theta}) = \mathbf{y} \right\}. \quad (1.2.1)$$

These form a sub-manifold in \mathbb{R}^p , see Fig. 1.1 for a cartoon. Points of this manifold correspond to models that perfectly interpolate the data, i.e. ‘interpolators’.

For most initializations $\boldsymbol{\theta}_0$, the manifold of interpolators ERM_0 passes close to $\boldsymbol{\theta}_0$. GD or SGD converge quickly to a specific point on ERM_0 which is close in a suitable sense.

As emphasized, this scenario is an hypothesis, and we do not know precise conditions under which it holds. In fact, it is easy to construct counter-examples. For instance if the function $f(\cdot; \boldsymbol{\theta})$ depends on the parameter $\boldsymbol{\theta}$ only through its first k coordinates $\theta_1, \dots, \theta_k$. Then there might not¹ be a solution to the n equations $\mathbf{y} = f_n(\boldsymbol{\theta})$ for $k < n$ (despite $n \leq p$).

One setting in which the conditions for interpolation are better understood is the ‘linear’ regime which we discuss next.

1.3 The linear regime

In certain cases, it can be proved that the weights do not change much during training, and it is therefore accurate to approximate $f(\cdot; \boldsymbol{\theta})$ by its first order Taylor expansion in $\boldsymbol{\theta}_0$. Justifying this approximation is not the main focus of these lectures which instead take it as a starting point and derive some of its consequences.

We will outline nevertheless an explanation, deferring to the literature for a rigorous treatment (see [DZPS18, COB19, BMR21] and Appendix B). We are looking for a solution to the interpolating equation $f_n(\boldsymbol{\theta}) = \mathbf{y}$. By Taylor’s theorem, this yields

$$\mathbf{y} - f_n(\boldsymbol{\theta}_0) = f_n(\boldsymbol{\theta}) - f_n(\boldsymbol{\theta}_0) \tag{1.3.1}$$

$$= \mathbf{D}f_n(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \int_0^1 (\mathbf{D}f_n(\boldsymbol{\theta}_t) - \mathbf{D}f_n(\boldsymbol{\theta}_0))(\boldsymbol{\theta} - \boldsymbol{\theta}_0) dt \tag{1.3.2}$$

$$=: \boldsymbol{\Phi}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \mathbf{e}(\boldsymbol{\theta}), \tag{1.3.3}$$

where $\boldsymbol{\theta}_t := (1-t)\boldsymbol{\theta}_0 + t\boldsymbol{\theta}$ and we denoted by $\boldsymbol{\Phi} := \mathbf{D}f_n(\boldsymbol{\theta}_0) \in \mathbb{R}^{n \times p}$ the Jacobian of the evaluation map at the initialization $\boldsymbol{\theta}_0$. Further

$$\|\mathbf{e}(\boldsymbol{\theta})\|_2 \leq L_n \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2, \tag{1.3.4}$$

$$L_n := \sup_{\boldsymbol{\theta} \neq \boldsymbol{\theta}_0} \frac{\|\mathbf{D}f_n(\boldsymbol{\theta}) - \mathbf{D}f_n(\boldsymbol{\theta}_0)\|_{\text{op}}}{\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2}. \tag{1.3.5}$$

Neglecting the second order contribution in the above equation suggests a solution of the form (here $\boldsymbol{\Phi}^+$ denotes the pseudoinverse)

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 + \boldsymbol{\Phi}^+(\mathbf{y} - f_n(\boldsymbol{\theta}_0) + \boldsymbol{\delta}), \tag{1.3.6}$$

with the following equation for $\boldsymbol{\delta}$ (writing $\tilde{\mathbf{y}} := \mathbf{y} - f_n(\boldsymbol{\theta}_0)$):

$$\boldsymbol{\delta} = -\mathbf{e}(\boldsymbol{\theta}_0 + \boldsymbol{\Phi}^+(\tilde{\mathbf{y}} + \boldsymbol{\delta})). \tag{1.3.7}$$

Defining $\mathbf{F}(\boldsymbol{\delta}) := -\mathbf{e}(\boldsymbol{\theta}_0 + \boldsymbol{\Phi}^+(\tilde{\mathbf{y}} + \boldsymbol{\delta}))$, by Eq. (1.3.4) we have $\|\mathbf{F}(\boldsymbol{\delta})\|_2 \leq L_n(\|\boldsymbol{\Phi}^+\tilde{\mathbf{y}}\|_2 + \|\boldsymbol{\Phi}^+\|_{\text{op}}\|\boldsymbol{\delta}\|_2)^2$. In other words, \mathbf{F} maps the ball of radius r into the ball of radius $L_n(\|\boldsymbol{\Phi}^+\tilde{\mathbf{y}}\|_2 + \|\boldsymbol{\Phi}^+\|_{\text{op}}r)^2$. By taking $r = \|\boldsymbol{\Phi}^+\tilde{\mathbf{y}}\|_2 / \|\boldsymbol{\Phi}^+\|_{\text{op}}$ we obtain that, for

$$L_n \leq \frac{1}{4\|\boldsymbol{\Phi}^+(\mathbf{y} - f_n(\boldsymbol{\theta}_0))\|_2\|\boldsymbol{\Phi}^+\|_{\text{op}}}, \tag{1.3.8}$$

¹The situation is a bit more complicated: if all we are interested in is interpolation, then less than n parameters can be sufficient for certain parametric classes $f(\cdot; \boldsymbol{\theta})$. However, these solutions are typically fragile to noise and hard to compute.

the ball of radius r is mapped into itself. Hence, by Brouwer's fixed point theorem, there exists a solution (an interpolator) of the form (1.3.6) with $\|\boldsymbol{\delta}\|_2 < r$.

Summarizing, under condition (1.3.8), there exists an interpolator that is well approximated by replacing the nonlinear model $f(\cdot; \boldsymbol{\theta})$ by its linearization

$$f_{\text{lin}}(\mathbf{x}; \boldsymbol{\theta}) := f(\mathbf{x}; \boldsymbol{\theta}_0) + \langle \boldsymbol{\theta} - \boldsymbol{\theta}_0, \nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta}_0) \rangle. \quad (1.3.9)$$

Indeed, for $\boldsymbol{\delta} = 0$, the expression in Eq. (1.3.6) coincides with the solution of the linearized equations

$$f_{\text{lin}}(\mathbf{x}_i; \boldsymbol{\theta}) = y_i, \quad \forall i \leq n, \quad (1.3.10)$$

which minimizes the ℓ_2 distance from initialization.

This suggests to define the following linearized empirical risk

$$\hat{R}_{\text{lin},n}(\boldsymbol{\theta}) := \frac{1}{n} \|\mathbf{y} - f_{\text{lin},n}(\boldsymbol{\theta})\|_2^2 \quad (1.3.11)$$

$$= \frac{1}{n} \|\tilde{\mathbf{y}} - \boldsymbol{\Phi}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\|_2^2, \quad (1.3.12)$$

and the corresponding linearized test error:

$$R_{\text{lin}}(\boldsymbol{\theta}) := \mathbb{E}\{(y_{\text{new}} - f_{\text{lin}}(\mathbf{x}_{\text{new}}; \boldsymbol{\theta}))^2\}. \quad (1.3.13)$$

Several papers prove that, under conditions analogous to (1.3.8), the original problem and the linearized one are close to each other, see Appendix B. Informally, denoting by $\boldsymbol{\theta}(t)$ the gradient flow for \hat{R}_n , and by $\boldsymbol{\theta}_{\text{lin}}(t)$ the gradient flow for $\hat{R}_{\text{lin},n}$, these proofs imply the following:

1. The empirical risk converges exponentially fast to 0. Namely, for all t , we have $\hat{R}_n(\boldsymbol{\theta}(t)) \leq \hat{R}_n(\boldsymbol{\theta}_0) \exp(-\lambda_0 t/2)$, where $\lambda_0 := \|\boldsymbol{\Phi}^+\|_{\text{op}}^2 = \sigma_{\min}(\boldsymbol{\Phi})^2$.
2. The parameters of one model are tracked by the ones of the other model, i.e. $\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}_{\text{lin}}(t)\|_2 \ll \|\boldsymbol{\theta}(t) - \boldsymbol{\theta}_0\|_2$ for all t .
3. The linearized model is a good approximation for a fully nonlinear model. Namely, for all t

$$\|f_{\text{lin}}(\cdot; \boldsymbol{\theta}_{\text{lin}}(t)) - f(\cdot; \boldsymbol{\theta}(t))\|_{L^2} \ll \|f(\cdot; \boldsymbol{\theta}(t))\|_{L^2}. \quad (1.3.14)$$

(Here $\|g\|_{L^2} := \mathbb{E}\{g(\mathbf{x})^2\}^{1/2}$.)

From a statistical perspective, the most important point is the last one, cf. Eq. (1.3.14), since it says that the model learnt within the linear theory is a good approximation of the original nonlinear model *at a random test point*, rather than at a training point. In particular, by triangular inequality

$$|R(\boldsymbol{\theta}(t))^{1/2} - R_{\text{lin}}(\boldsymbol{\theta}_{\text{lin}}(t))^{1/2}| \ll \mathbb{E}\{f(\mathbf{x}; \boldsymbol{\theta}(t))^2\}^{1/2}. \quad (1.3.15)$$

From here on, we will focus on such linearized models, and try to understand their generalization error. Notice that the parameter vector $\boldsymbol{\theta}_{\text{lin}}(t)$ depends on time and hence in principle we would like to study the whole function $t \mapsto R_{\text{lin}}(\boldsymbol{\theta}_{\text{lin}}(t))$. Of particular interest is the limit $t \rightarrow \infty$ (which

corresponds to interpolation). It is an elementary fact that gradient flow with a quadratic cost function converges to the minimizer that is closest to the initialization in $\|\cdot\|_{\mathbf{S}^{-1}}$ norm:

$$\lim_{t \rightarrow \infty} \boldsymbol{\theta}_{\text{lin}}(t) = \operatorname{argmin} \left\{ \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_{\mathbf{S}^{-1}} : \mathbf{D}f_n(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0) = \mathbf{y} - f_n(\boldsymbol{\theta}_0) \right\} \quad (1.3.16)$$

$$= \boldsymbol{\theta}_0 + \operatorname{argmin} \left\{ \|\mathbf{b}\|_{\mathbf{S}^{-1}} : \mathbf{D}f_n(\boldsymbol{\theta}_0)\mathbf{b} = \mathbf{y} - f_n(\boldsymbol{\theta}_0) \right\}, \quad (1.3.17)$$

where $\|\mathbf{b}\|_{\mathbf{S}^{-1}} = \|\mathbf{S}^{-1/2}\mathbf{b}\|_2 = \langle \mathbf{b}, \mathbf{S}^{-1}\mathbf{b} \rangle^{1/2}$.

Hereafter, we will typically drop the subscripts ‘lin’. Rather than studying the gradient flow path, we will focus on a different path that also interpolates between the $\boldsymbol{\theta}_0$ and the min- ℓ_2 distance interpolator. Namely, we will consider ridge regression

$$\hat{\mathbf{b}}(\lambda) := \operatorname{argmin}_{\mathbf{b} \in \mathbb{R}^p} \left\{ \frac{1}{n} \|\tilde{\mathbf{y}} - \boldsymbol{\Phi}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_{\mathbf{S}^{-1}}^2 \right\}, \quad (1.3.18)$$

where we recall that $\tilde{\mathbf{y}} = \mathbf{y} - f_n(\boldsymbol{\theta}_0)$ and $\boldsymbol{\Phi} = \mathbf{D}f_n(\boldsymbol{\theta}_0)$.

1.4 Linearization of two-layer networks

As mentioned above, we will focus on the case of two-layer neural networks, cf. Eq. (1.1.7). We will assume σ to be weakly differentiable with weak derivative σ' . A simple calculation yields

$$f_{\text{lin}}(\mathbf{x}; \boldsymbol{\theta}_0 + \mathbf{b}) = f(\mathbf{x}; \boldsymbol{\theta}_0) + \alpha \sum_{i=1}^N b_{1,i} \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle) + \alpha \sum_{i=1}^N a_i \langle \mathbf{b}_{2,i}, \mathbf{x} \rangle \sigma'(\langle \mathbf{w}_i, \mathbf{x} \rangle), \quad (1.4.1)$$

$$\boldsymbol{\theta}_0 := (a_1, \dots, a_N; \mathbf{w}_1, \dots, \mathbf{w}_N) \in \mathbb{R}^{N(d+1)}, \quad (1.4.2)$$

$$\mathbf{b} := (b_{1,1}, \dots, b_{1,N}; \mathbf{b}_{2,1}, \dots, \mathbf{b}_{2,N}) \in \mathbb{R}^{N(d+1)}. \quad (1.4.3)$$

We can rewrite this linear model in terms of the following *featurization maps*:

$$\boldsymbol{\phi}_{\text{RF}}(\mathbf{x}) = \frac{1}{\sqrt{N}} [\sigma(\langle \mathbf{w}_1, \mathbf{x} \rangle); \dots; \sigma(\langle \mathbf{w}_N, \mathbf{x} \rangle)], \quad (1.4.4)$$

$$\boldsymbol{\phi}_{\text{NT}}(\mathbf{x}) = \frac{1}{\sqrt{Nd}} [\sigma'(\langle \mathbf{w}_1, \mathbf{x} \rangle) \mathbf{x}^\top; \dots; \sigma'(\langle \mathbf{w}_N, \mathbf{x} \rangle) \mathbf{x}^\top], \quad (1.4.5)$$

We have $\boldsymbol{\phi}_{\text{RF}} : \mathbb{R}^d \rightarrow \mathbb{R}^N$ and $\boldsymbol{\phi}_{\text{NT}} : \mathbb{R}^d \rightarrow \mathbb{R}^{Nd}$. Setting for convenience $\alpha = 1/\sqrt{N}$, we get

$$f_{\text{lin}}(\mathbf{x}; \boldsymbol{\theta}_0 + \mathbf{b}) = f(\mathbf{x}; \boldsymbol{\theta}_0) + \langle \mathbf{b}_1, \boldsymbol{\phi}_{\text{RF}}(\mathbf{x}) \rangle + \sqrt{d} \langle \mathbf{b}_2, \boldsymbol{\phi}_{\text{NT}}(\mathbf{x}) \rangle. \quad (1.4.6)$$

We define the design matrix

$$\boldsymbol{\Phi} := \begin{bmatrix} \text{---} \boldsymbol{\phi}_{\text{RF}}(\mathbf{x}_1) \text{---} & \text{---} \boldsymbol{\phi}_{\text{NT}}(\mathbf{x}_1) \text{---} \\ \text{---} \boldsymbol{\phi}_{\text{RF}}(\mathbf{x}_2) \text{---} & \text{---} \boldsymbol{\phi}_{\text{NT}}(\mathbf{x}_2) \text{---} \\ \vdots & \\ \text{---} \boldsymbol{\phi}_{\text{RF}}(\mathbf{x}_n) \text{---} & \text{---} \boldsymbol{\phi}_{\text{NT}}(\mathbf{x}_n) \text{---} \end{bmatrix}, \quad (1.4.7)$$

and consider the stepsize scaling matrix

$$\mathbf{S} = \operatorname{diag}(\underbrace{1, \dots, 1}_N; \underbrace{sd, \dots, sd}_{Nd}). \quad (1.4.8)$$

With suitable redefinition of $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2) \in \mathbb{R}^N \times \mathbb{R}^{Nd}$, the ridge regression problem thus reads

$$\hat{\mathbf{b}}(\lambda) := \operatorname{argmin}_{\mathbf{b} \in \mathbb{R}^p} \left\{ \frac{1}{n} \|\tilde{\mathbf{y}} - \Phi \mathbf{b}\|_2^2 + \lambda_{\text{RF}} \|\mathbf{b}_1\|_2^2 + \lambda_{\text{NT}} \|\mathbf{b}_2\|_2^2 \right\}, \quad (1.4.9)$$

where $\lambda_{\text{RF}} := \lambda$, $\lambda_{\text{NT}} := \lambda/s$. We conclude this section with two remarks.

Remark 1.4.1. We saw that GD with respect to a quadratic cost function converges to the closest minimizer to the initialization, where ‘closest’ is measured by a suitably weighted ℓ_2 distance. This is the simplest example of a more general phenomenon known as ‘implicit regularization’: when learning overparametrized models, common optimization algorithms select empirical risk minimizers that present special simplicity properties (smallness of certain norms). While this is normally achieved by explicitly regularizing the risk to promote those properties, in overparametrized system it is implicitly induced (to some extent) by the dynamics of the optimization algorithm. Examples of this phenomenon include [GWB⁺17, GLSS18a, SHN⁺18, JT18b] (see also Appendix B).

Note that Eq. (1.4.9) also illustrates how the precise form of implicit regularization depends on the optimization algorithm. We mentioned that gradient flow converges to the $\lambda = 0+$ solution of Eq. (1.4.9), which corresponds to the interpolator that minimizes $\|\mathbf{b}_1\|_2^2 + \|\mathbf{b}_2\|_2^2/s$. The precise form of the norm that is implicitly regularized depends on the details of the optimization algorithm (in this case the ratio of learning rates s).

Remark 1.4.2. The ridge regression problem of Eq. (1.4.9) presents another peculiarity. The responses \mathbf{y} have been replaced by the residuals at initialization $\tilde{\mathbf{y}} = \mathbf{y} - f_n(\boldsymbol{\theta}_0)$. In general, this fact should be taken into account when analyzing the linear system. However, it is possible to construct initializations close to standard random initializations and yet have $f_n(\boldsymbol{\theta}_0) = \mathbf{0}$. We will therefore neglect the difference between \mathbf{y} and $\tilde{\mathbf{y}}$.

1.5 Outline of this tutorial

The rest of this tutorial is organized as follows.

Chapter 2 studies ridge regression under a simpler model in which the feature vectors $\phi(\mathbf{x}) \in \mathbb{R}^p$ are completely characterized by their mean (which we will assume to be zero) and covariance $\boldsymbol{\Sigma}$. Under suitable concentration assumption on the feature vector, the resulting risk is universal and can be precisely characterized. Despite its simplicity, this model displays many interesting phenomena.

In fact, while the more complex settings in the following chapters do not fit the required concentration assumptions, their behavior is correctly predicted by this simpler model, pointing at a remarkable universality phenomenon.

Chapter 3 focuses on the infinite width ($N \rightarrow \infty$) limit of the neural tangent model we derived above. This is described by kernel ridge regression for a rotationally invariant kernel. We derive the generalization behavior of KRR in high dimension.

Chapter 4 studies ‘random feature models,’ which correspond to setting $\mathbf{b}_2 = 0$ in (1.4.9), i.e. fitting only the second-layer weights. This clarifies what happens when moving from an infinitely wide networks to finite width.

Chapter 5 considers the other limit case in which we set $\mathbf{b}_1 = 0$ and only learn the first layer weights (in the linear approximation).

It is not hard to see that the case in which both $\mathbf{b}_1, \mathbf{b}_2$ are fit to data is very close to the one in which $\mathbf{b}_1 = 0$ and \mathbf{b}_2 is fit to data. Therefore this case yields the correct insights into the generalization behavior of finite width neural tangent models.

Chapter 6 finally discusses the limitations of linear theory. In particular, we discuss some simple examples in which we need to go beyond the neural tangent theory to capture the behavior of actual neural networks trained via gradient descent.

Chapter 2

Linear regression under feature concentration assumptions

In this section we consider ridge regression and its limit for vanishing regularization (minimum ℓ_2 -norm regression), focusing on the overparametrized regime $p > n$. Our objective is to understand if and when interpolation or overfitting is compatible with good generalization. With this in mind, we start by considering a simple model in which the feature vectors are completely characterized by their covariance Σ . With little loss of generality, we will assume the vectors to be centered. Under certain concentration assumptions on these vectors, a sharp characterization of the prediction risk can be derived. This theoretical prediction captures in a precise way numerous interesting phenomena such as benign overfitting and double descent.

2.1 Setting and sharp characterization

In this chapter we assume to be given i.i.d. samples $\{(y_i, \mathbf{z}_i)\}_{i \leq n}$ where responses are given by

$$y_i = \langle \boldsymbol{\beta}_*, \mathbf{z}_i \rangle + w_i, \quad \mathbb{E}(w_i | \mathbf{z}_i) = 0, \quad \mathbb{E}(w_i^2 | \mathbf{z}_i) = \tau^2. \quad (2.1.1)$$

We assume $\mathbb{E}(\mathbf{z}_i) = \mathbf{0}$, $\mathbb{E}(\mathbf{z}_i \mathbf{z}_i^\top) = \Sigma$. (The case of non-zero $\mathbb{E}(\mathbf{z}_i) \neq \mathbf{0}$ could be treated at the cost of some notational burden, but does not present conceptual novelties.)

While we will state results that hold for a broad class of non-Gaussian vectors, for pedagogical reasons we will begin with the simplest example:

$$\mathbf{z}_i \sim \mathbf{N}(0, \Sigma) \perp w_i \sim \mathbf{N}(0, \tau^2). \quad (2.1.2)$$

The covariance $\Sigma \in \mathbb{R}^{p \times p}$ and coefficient vector $\boldsymbol{\beta}_* \in \mathbb{R}^p$ are unknown, alongside the noise level τ . We estimate $\boldsymbol{\beta}_*$ using ridge regression:

$$\hat{\boldsymbol{\beta}}(\lambda) := \arg \min_{\mathbf{b} \in \mathbb{R}^p} \left\{ \frac{1}{n} \|\mathbf{y} - \mathbf{Z}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_2^2 \right\}, \quad (2.1.3)$$

where $\mathbf{Z} \in \mathbb{R}^{n \times p}$ is the matrix whose i -th row is \mathbf{z}_i and $\mathbf{y} \in \mathbb{R}^n$ is the vector whose i -th entry is y_i .

Remark 2.1.1. We emphasize that the above definitions make perfect sense if $p = \infty$. In this case \mathbb{R}^p is interpreted to be the Hilbert space of square summable sequences ℓ_2 and $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^\top \mathbf{v}$ is the scalar product in ℓ_2 . In fact, unless specified otherwise, the results below cover this infinite-dimensional case.

We are interested in the excess test error. With a slight abuse of notation, we can use the notation $R_{\text{exc}}(\hat{f}) = R_{\text{exc}}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_0, \mathbf{w})$ since $\hat{f}(\mathbf{x}) = \langle \hat{\boldsymbol{\beta}}(\lambda), \mathbf{x} \rangle$ is a function of $\lambda, \mathbf{Z}, \boldsymbol{\beta}_0, \mathbf{w}$. We have:

$$R_{\text{exc}}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_0, \mathbf{w}) := \mathbb{E}_{\mathbf{z}_{\text{new}}} \{ (\langle \hat{\boldsymbol{\beta}}(\lambda), \mathbf{z}_{\text{new}} \rangle - \langle \boldsymbol{\beta}_*, \mathbf{z}_{\text{new}} \rangle)^2 \} \quad (2.1.4)$$

$$= \|\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta}_*\|_{\boldsymbol{\Sigma}}^2. \quad (2.1.5)$$

Recall that in Chapter 1, Eq. (1.1.3), we defined the excess test error to be the difference between test error and Bayes error. In this case the Bayes error is equal to τ^2 , and therefore the relation is particularly simple: $R(f) = R_{\text{exc}}(f) + \tau^2$ (reverting to using the f to denote the argument).

We also note that the ridge regression estimator can be written explicitly as

$$\hat{\boldsymbol{\beta}}(\lambda) = \frac{1}{n} \mathbf{S}_\lambda \mathbf{Z}^\top \mathbf{y}, \quad \mathbf{S}_\lambda = \left(\frac{1}{n} \mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{I}_p \right)^{-1}, \quad (2.1.6)$$

whence we obtain the expression:

$$R_{\text{exc}}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_*, \mathbf{w}) := \left\| \lambda \mathbf{S}_\lambda \boldsymbol{\beta}_* - \frac{1}{n} \mathbf{S}_\lambda \mathbf{Z}^\top \mathbf{w} \right\|_{\boldsymbol{\Sigma}}^2. \quad (2.1.7)$$

Note that the risk (2.1.5) depends on the noise vector \mathbf{w} . It is useful to define its expectation with respect to \mathbf{w} , which can be exactly decomposed in a bias term and a variance term:

$$\bar{R}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_*) := \mathbb{E}_{\mathbf{w}} R_{\text{exc}}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_*, \mathbf{w}) \quad (2.1.8)$$

$$= \mathbf{B}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_*) + \mathbf{V}(\lambda; \mathbf{Z}). \quad (2.1.9)$$

The bias and variance are given by

$$\mathbf{B}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_*) := \lambda^2 \langle \boldsymbol{\beta}_*, \mathbf{S}_\lambda \boldsymbol{\Sigma} \mathbf{S}_\lambda \boldsymbol{\beta}_* \rangle, \quad (2.1.10)$$

$$\mathbf{V}(\lambda; \mathbf{Z}) := \frac{\tau^2}{n} \text{Tr} \left(\mathbf{S}_\lambda^2 \frac{1}{n} \mathbf{Z}^\top \mathbf{Z} \boldsymbol{\Sigma} \right). \quad (2.1.11)$$

Of course these formulas do not provide always simple insights into the qualitative behavior of the test error. In particular, $\mathbf{B}(\lambda; \mathbf{Z}, \boldsymbol{\beta}_*)$ and $\mathbf{V}(\lambda; \mathbf{Z})$ are random quantities because of the randomness in \mathbf{Z} . The next theorem shows that these quantities concentrate around deterministic predictions that depend uniquely on the geometry of $(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*)$.

Before stating this characterization, we introduce some important notions.

Definition 2.1.1 (Effective dimension). *We say that $\boldsymbol{\Sigma}$ has effective dimension $\mathbf{d}_{\boldsymbol{\Sigma}}(n)$ if, for all $1 \leq k \leq \min\{n, p\}$,*

$$\sum_{l=k}^p \sigma_l \leq \mathbf{d}_{\boldsymbol{\Sigma}} \sigma_k.$$

Without loss of generality, we will always choose $\mathbf{d}_{\boldsymbol{\Sigma}}(n) \geq n$.

Definition 2.1.2 (Bounded varying spectrum). *We say that $\boldsymbol{\Sigma}$ has bounded varying spectrum if there exists a monotone decreasing function $\psi : (0, 1] \rightarrow [1, \infty)$ with $\lim_{\delta \downarrow 0} \psi(\delta) = \infty$, such that $\sigma_{\lfloor \delta i \rfloor} / \sigma_i \leq \psi(\delta)$ for all $\delta \in (0, 1]$, $i \in \mathbb{N}$ and $\delta i \geq 1$.*

Given Σ , λ , let $\lambda_*(\lambda) \geq 0$ be the unique positive solution of

$$n\left(1 - \frac{\lambda}{\lambda_*}\right) = \text{Tr}\left(\Sigma(\Sigma + \lambda_*\mathbf{I})^{-1}\right). \quad (2.1.12)$$

(with $\lambda_* = 0$ if $\lambda = 0$ and $n \geq p$.) Define $\mathcal{B}(\Sigma, \beta_*)$ and $\mathcal{V}(\Sigma)$ by

$$\mathcal{B}(\Sigma, \beta_*) := \frac{\lambda_*^2 \langle \beta_*, (\Sigma + \lambda_*\mathbf{I})^{-2} \Sigma \beta_* \rangle}{1 - n^{-1} \text{Tr}(\Sigma^2 (\Sigma + \lambda_*\mathbf{I})^{-2})}, \quad (2.1.13)$$

$$\mathcal{V}(\Sigma) := \frac{\tau^2 \text{Tr}(\Sigma^2 (\Sigma + \lambda_*\mathbf{I})^{-2})}{n - \text{Tr}(\Sigma^2 (\Sigma + \lambda_*\mathbf{I})^{-2})}. \quad (2.1.14)$$

Let us emphasize that $\mathcal{B}(\Sigma, \beta_*)$, $\mathcal{V}(\Sigma)$ are deterministic quantities.

The next theorem gives sufficient conditions under which $\mathcal{B}(\Sigma, \beta_*)$, $\mathcal{V}(\Sigma)$ are accurate multiplicative approximations of the actual bias and variance.

Theorem 1. *Assume $\|\Sigma\|_{\text{op}} = 1$, a setting which we can always reduce to by rescaling \mathbf{Z} . Further assume $\|\Sigma^{-1/2}\beta_*\| < \infty$, and that one of the following scenarios holds:*

1. **Proportional regime.** *There exists a constant $M < \infty$ such that $p/n \in [1/M, M]$, $\sigma_p \geq 1/M$, $\lambda \in [1/M, M]$.*
2. **Dimension-free regime.** *Σ has effective dimension $\mathbf{d}_\Sigma(n)$, bounded-varying spectrum, and there exist constants $M > 0$ and $\gamma \in (0, 1/3)$ such that the following hold. Define*

$$\rho(\lambda) := \frac{\langle \beta_*, \Sigma(\lambda_*\mathbf{I} + \Sigma)^{-1} \beta_* \rangle}{\|\beta_*\|^2 \text{Tr}(\Sigma(\lambda_*\mathbf{I} + \Sigma)^{-1})}. \quad (2.1.15)$$

Then we have $\lambda_(0) \leq M$, $\lambda \in [\lambda_*(0)/M, \lambda_*(0)M]$ and $\mathbf{d}_\Sigma(n) \leq \rho(\lambda)^{1/6} n^{1+\gamma}$.*

Then, there exists a constant C_0 (depending uniquely on the constants in the assumptions), such that for $n \geq C_0$, the following holds with probability at least $1 - n^{-10}$. We have

$$\mathbf{B}(\lambda; \mathbf{Z}, \beta_*) = (1 + \text{err}_B(n)) \cdot \mathcal{B}(\Sigma, \beta_*), \quad (2.1.16)$$

$$\mathbf{V}(\lambda; \mathbf{Z}) = (1 + \text{err}_V(n)) \cdot \mathcal{V}(\Sigma), \quad (2.1.17)$$

where, under the proportional regime (scenario 1 above), we have $|\text{err}_B(n)| \leq n^{-0.49}$, $|\text{err}_V(n)| \leq n^{-0.99}$, while, in the dimension-free regime (scenario 2 above), $|\text{err}_B(n)| \leq (\mathbf{d}_\Sigma(n)/n)^3 / (\rho(\lambda)^{1/2} n^{0.99})$, $|\text{err}_V(n)| \leq (\mathbf{d}_\Sigma(n)/n)^3 / n^{0.99}$.

Remark 2.1.2. Defining $F_\Sigma(x) := n^{-1} \text{Tr}(\Sigma(\Sigma + x\mathbf{I})^{-1})$, Eq. (2.1.12) reads:

$$1 - \frac{\lambda}{\lambda_*} = F_\Sigma(\lambda_*). \quad (2.1.18)$$

For $\lambda > 0$, this equation has always a unique solution by monotonicity.

For $\lambda = 0$, the left hand side is constant and equal to 1. The right-hand side is strictly monotone decreasing with $F_\Sigma(0) = p/n$ and $\lim_{x \rightarrow \infty} F_\Sigma(x) = 0$. Hence for $p/n > 1$ (overparametrized regime), the equation has a unique solution $\lambda_* > 0$.

For $p/n \leq 1$ (underparametrized regime), we set $\lambda_* = 0$ by definition.

Remark 2.1.3 (Ridgeless limit). By virtue of the previous remark, the predicted bias and variance $\mathcal{B}(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*)$ and $\mathcal{V}(\boldsymbol{\Sigma})$, make perfect sense for the case $\lambda = 0$. Indeed, Theorem 1 holds for $\lambda = 0+$ as well, although this requires an additional argument and possibly larger error terms.

Theorem 1 has a simple interpretation in terms of a simpler *sequence model*, which we next define. In the sequence model we want to estimate $\boldsymbol{\beta}_*$ from observation \mathbf{y}^s given by

$$\mathbf{y}^s = \boldsymbol{\Sigma}^{1/2} \boldsymbol{\beta}_* + \frac{\omega}{\sqrt{n}} \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_p). \quad (2.1.19)$$

We use ridge regression at regularization level λ_* as defined in Eq. (2.1.12):

$$\hat{\boldsymbol{\beta}}^s(\lambda_*) := \operatorname{argmin}_{\mathbf{b} \in \mathbb{R}^p} \left\{ \|\mathbf{y}^s - \boldsymbol{\Sigma}^{1/2} \mathbf{b}\|_2^2 + \lambda_* \|\mathbf{b}\|_2^2 \right\}. \quad (2.1.20)$$

Then the prediction for the risk of the original model, namely $\mathcal{B}(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*) + \mathcal{V}(\boldsymbol{\Sigma})$, coincides with the risk of the sequence model, provided we choose ω to be the unique positive solution of

$$\omega^2 = \frac{\tau^2}{n} + \mathbb{E}_{\boldsymbol{\varepsilon}} \left\{ \|\hat{\boldsymbol{\beta}}^s(\lambda_*) - \boldsymbol{\beta}_*\|_{\boldsymbol{\Sigma}}^2 \right\}. \quad (2.1.21)$$

The solution of this equation is easy to express in terms of quantities appearing in the theorem statement

$$\omega^2 = \frac{\tau^2 + \lambda_*^2 \langle \boldsymbol{\beta}_*, (\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-2} \boldsymbol{\Sigma} \boldsymbol{\beta}_* \rangle}{1 - n^{-1} \operatorname{Tr}(\boldsymbol{\Sigma}^2 (\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-2})}.$$

To summarize we have the following correspondence:

	Gaussian feature model	Sequence model
Design matrix	Random design matrix \mathbf{Z}	Deterministic design $\boldsymbol{\Sigma}^{1/2}$
Ridge penalty	λ	$\lambda_* > \lambda$
Noise variance	τ^2	$\omega^2 > \tau^2$

Note in particular, as pointed out above (Remark 2.1.2), in the overparametrized regime we have $\lambda_* > 0$ even if $\lambda = 0$. We refer to this phenomenon as ‘*self-induced regularization*’: the noisy feature vectors in the original unregularized problem induce an effective regularization in the equivalent sequence model.

Self-induced regularization is a key mechanism by which an interpolating model can generalize well. Roughly speaking, although $\lambda = 0+$, the model behaves as if the sample covariance was replaced by the population one, but the regularization was increased from 0 to λ_* . The noise in the covariates acts as a regularizer.

In order to make this correspondence even more concrete, we note that

$$\frac{1}{n} \operatorname{Tr}(\boldsymbol{\Sigma}^2 (\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-2}) < \frac{1}{n} \operatorname{Tr}(\boldsymbol{\Sigma} (\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-1}) < 1 \quad (2.1.22)$$

Let us assume that this inequality holds uniformly. Namely, there exists a constant $c_1 < \infty$ such that

$$\frac{1}{n} \operatorname{Tr}(\boldsymbol{\Sigma}^2 (\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-2}) \leq 1 - \frac{1}{c_1}. \quad (2.1.23)$$

Substituting in Eqs. (2.1.14), (2.1.13), we get

$$\mathcal{V}(\boldsymbol{\Sigma}) \leq \frac{c_1 \tau^2}{n} \text{Tr}(\boldsymbol{\Sigma}^2 (\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-2}), \quad (2.1.24)$$

$$\mathcal{B}(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*) \leq c_1 \lambda_*^2 \langle \boldsymbol{\beta}_*, (\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-2} \boldsymbol{\Sigma} \boldsymbol{\beta}_* \rangle. \quad (2.1.25)$$

Notice that the expressions on the right-hand side (up to the constant c_1) are the bias and variance of the sequence model with $\omega^2 = \tau^2$. In other words, in some cases of interest, simply considering the sequence model with $\omega^2 = \tau^2$ allows to bound (up to constants) the risk of the original problem.

2.2 Non-Gaussian covariates

Theorem 1 holds beyond Gaussian covariates, and was proven under the following assumptions on the covariates.

Let $\mathbf{z}_i := \boldsymbol{\Sigma}^{-1/2} \mathbf{x}_i$, so that \mathbf{z}_i is isotropic, namely $\mathbb{E}\{\mathbf{z}_i\} = \mathbf{0}$, $\mathbb{E}\{\mathbf{z}_i \mathbf{z}_i^\top\} = \mathbf{I}$. We then consider the following two models for \mathbf{z}_i , depending on a constant $\kappa_{\mathbf{x}} > 0$:

- (a) **Independent sub-Gaussian coordinates:** \mathbf{z}_i has independent but not necessarily identically distributed coordinates with uniformly bounded sub-Gaussian norm. Namely: each coordinate z_{ij} of \mathbf{z}_i satisfies $\mathbb{E}[z_{ij}] = 0$, $\text{Var}(z_{ij}) = 1$ and $\|z_{ij}\|_{\psi_2} := \sup_{p \geq 1} p^{-\frac{1}{2}} \mathbb{E}\{|z_{ij}|^p\}^{\frac{1}{p}} \leq \kappa_{\mathbf{x}}$.
- (b) **Convex concentration:** allowing \mathbf{z}_i to have dependent coordinates, the following holds for any 1-Lipschitz convex function $\varphi : \mathbb{R}^p \rightarrow \mathbb{R}$, and for every $t > 0$

$$\mathbb{P}\{|\varphi(\mathbf{z}_i) - \mathbb{E}\varphi(\mathbf{z}_i)| \geq t\} \leq 2 e^{-t^2/\kappa_{\mathbf{x}}^2}.$$

For independent sub-Gaussian coordinates, a version of Theorem 1 was proven in [HMRT22], although with larger error terms than stated here. The form stated here, the infinite-dimensional case, with regularly varying spectrum, and the case of covariates satisfying convex concentration were proven in [CM22].

2.3 An example: Analysis of a latent space model

As an application of the general theory in Section 2.1, it is instructive to consider the following *latent space model*. We assume the response y to be linear in an underlying covariate vector $\mathbf{x} \in \mathbb{R}^d$. However, we fit a model in the features $\mathbf{z} \in \mathbb{R}^p$, which are also linear in \mathbf{x} :

$$y_i = \langle \boldsymbol{\theta}_*, \mathbf{x}_i \rangle + \xi_i, \quad \xi_i \sim \mathbf{N}(0, \tau^2), \quad (2.3.1)$$

$$\mathbf{z}_i = \mathbf{W} \mathbf{x}_i + \mathbf{u}_i, \quad \mathbf{u}_i \sim \mathbf{N}(0, \mathbf{I}_p), \quad (2.3.2)$$

where the matrix \mathbf{W} is fixed (at random). We perform ridge regression of y_i on \mathbf{z}_i as in Eq. (2.1.3) (with \mathbf{Z} the matrix whose i -th row is given by \mathbf{z}_i).

We assume a simple model for the latent features, namely $\mathbf{x}_i \sim \mathbf{N}(0, \mathbf{I}_d)$, and \mathbf{W} proportional to an orthogonal matrix. Namely $\mathbf{W}^\top \mathbf{W} = (p\mu/d) \mathbf{I}_d$. We consider the proportional asymptotics $p, n, d \rightarrow \infty$ with

$$\frac{p}{n} \rightarrow \gamma \in (0, \infty), \quad \frac{d}{p} \rightarrow \psi \in (0, 1), \quad (2.3.3)$$

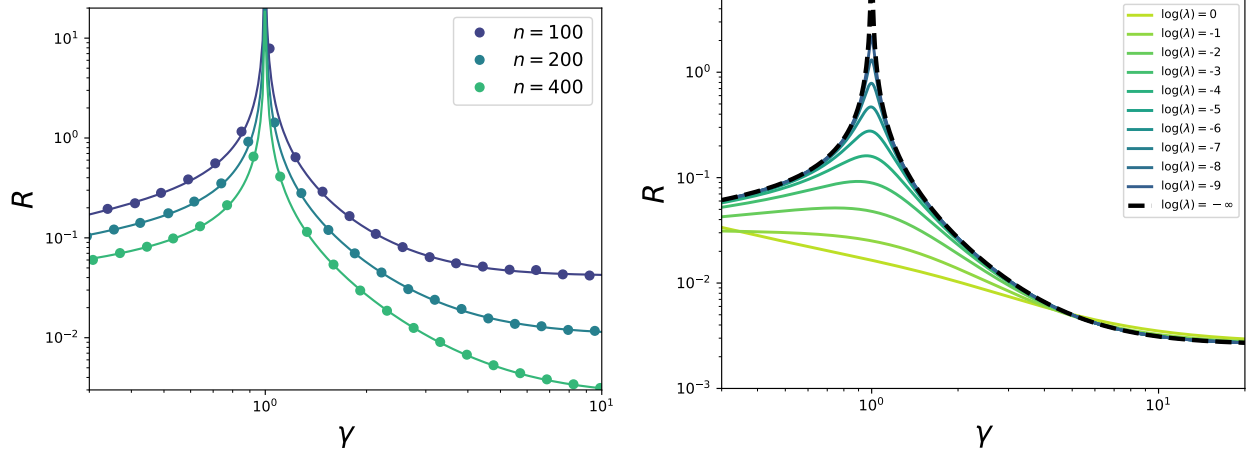


Figure 2.1: Test error of ridge regression under the latent space model. Left: circles are empirical results and curves are theoretical predictions within the proportional asymptotics, in the ridgeless limit $\lambda = 0+$. Here $d = 20$, $\tau = 0$, $r_\theta = 1$, $\mu = 1$. Different curves correspond to different sample sizes. Right: curves correspond to different values of the regularization parameter. Here $\gamma\psi = 1/20$, $\tau = 0$, $r_\theta = 1$, $\mu = 1$.
Figure from [HMRT22].

and $\|\boldsymbol{\theta}_*\| \rightarrow r_\theta$. In particular, γ has the interpretation of number of parameters per sample, and $\gamma > 1$ corresponds to the overparametrized regime.

Explicit formulas for the asymptotic risk can be obtained in this limit using Theorem 1. In particular, in the minimum norm limit $\lambda \rightarrow 0$, after a tedious but straightforward calculation, one obtains

$$\mathcal{B}_{\text{lat}}(\psi, \gamma) := \left\{ 1 + \gamma c_0 \frac{\mathcal{E}_1(\psi, \gamma)}{\mathcal{E}_2(\psi, \gamma)} \right\} \cdot \frac{\mu\psi^{-1}r_\theta^2}{(1 + \mu\psi^{-1})(1 + c_0\gamma(1 + \mu\psi^{-1}))^2}, \quad (2.3.4)$$

$$\mathcal{V}_{\text{lat}}(\psi, \gamma) := \sigma^2 \gamma c_0 \frac{\mathcal{E}_1(\psi, \gamma)}{\mathcal{E}_2(\psi, \gamma)}, \quad (2.3.5)$$

$$\mathcal{E}_1(\psi, \gamma) := \frac{1 - \psi}{(1 + c_0\gamma)^2} + \frac{\psi(1 + \mu\psi^{-1})^2}{(1 + c_0(1 + \mu\psi^{-1})\gamma)^2}, \quad (2.3.6)$$

$$\mathcal{E}_2(\psi, \gamma) := \frac{1 - \psi}{(1 + c_0\gamma)^2} + \frac{\psi(1 + \mu\psi^{-1})}{(1 + c_0(1 + \mu\psi^{-1})\gamma)^2}. \quad (2.3.7)$$

where $\sigma^2 = \tau^2 + r_\theta^2/(1 + \mu\psi^{-1})$, and $c_0 = c_0(\psi, \gamma) \geq 0$ is the unique non-negative solution of the following second order equation

$$1 - \frac{1}{\gamma} = \frac{1 - \psi}{1 + c_0\gamma} + \frac{\psi}{1 + c_0(1 + \mu\psi^{-1})\gamma}. \quad (2.3.8)$$

In Figure 2.1, left frame, we plot theoretical predictions and empirical results for the test error of minimum norm interpolation, as a function of the overparametrization ratio. A few features of this plot are striking:

1. A large spike is present at the interpolation threshold $\gamma = 1$: the test error grows when the model becomes overparametrized but then descends again at large overparametrization.

This behavior has been dubbed ‘double descent’ in [BHMM19] (see Appendix B for further discussions).

2. The minimum test error is achieved at large overparametrization $\lambda \gg 1$.
3. In particular overparametrized models behave better than underparametrized ones. Intuitively, the reason is that the latent space can be identified better when the dimension of \mathbf{z}_i grows.

In Figure 2.1, right frame, we plot theoretical predictions for the test error of ridge regression as a function of the overparametrization ratio γ . The number of samples per latent dimension n/d is kept fixed to $(\psi\gamma)^{-1} = 20$. Different curves correspond to different values of λ . The test error with optimal lambda is given by the lower envelope of these curves.

Two remarkable phenomena can be observed in these plots:

1. The spike at $\gamma = 1$ is smoothed for positive λ and disappears completely for the optimal regularizations. In other words, the double descent is not an intrinsic statistical phenomenon, and is instead due to under-regularization.
2. Nevertheless, at large overparametrization, the optimal regularization is $\lambda = 0+$.

The last point is particularly surprising: at large overparametrization (and large signal-to-noise ratio) the optimal test error is achieved by an interpolator. The same phenomenon survives at positive τ , and can be traced back to the fact that the effective regularization λ_* is strictly positive despite $\lambda = 0+$.

2.4 Bounds and benign overfitting

For a general Σ, β_* , the characterization given by Eqs. (2.1.12), (2.1.13), (2.1.14) may still be too detailed to provide a simple intuition. In this section we derive upper bounds on bias and variance that expose a particularly interesting phenomenon, which was first established in [BLLT20, TB20] using a technically different approach: benign overfitting.

We work under the simplifying assumption

$$\text{Tr}\{\Sigma^2(\Sigma + \lambda_*\mathbf{I})^{-2}\} \leq n(1 - c_*^{-1}), \quad (2.4.1)$$

for a constant $c_* \in (1, \infty)$. Let $\Sigma := \sum_{i \geq 1} \sigma_i \mathbf{v}_i \mathbf{v}_i^\top$ be the eigendecomposition of Σ , with $\sigma_1 \geq \sigma_2 \geq \dots \geq 0$. Denote by $\beta_{*, \leq k} := \sum_{i \leq k} \langle \beta_*, \mathbf{v}_i \rangle \mathbf{v}_i$ the orthogonal projection of β onto the span of the top k eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_k$, and by $\beta_{*, > k} := \beta_* - \beta_{*, \leq k}$ its complement. Finally, let $k_* := \max\{k : \sigma_k \geq \lambda_*\}$.

The above definitions are quite natural in view of the latent space model discussed in the previous section. We want to separate the contribution of ‘signal’ directions in covariate space (corresponding to the span of $(\mathbf{v}_i : i \leq k_*)$), from the contribution due to ‘junk’ covariates (the projection onto the span of $(\mathbf{v}_i : i > k_*)$). Technically, we bound all traces in Eqs. (2.1.12), (2.1.13), (2.1.14) after splitting the contributions of each two subspaces.

Consider Eq. (2.1.12), which implies $n \geq \text{Tr}\{\boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-1}\}$. We get

$$\begin{aligned} n &\geq \sum_{\ell=1}^{k_*} \frac{\sigma_\ell}{\sigma_\ell + \lambda_*} + \sum_{\ell=k_*+1}^p \frac{\sigma_\ell}{\sigma_\ell + \lambda_*} \\ &\geq \sum_{\ell=1}^{k_*} \frac{\sigma_\ell}{2\sigma_\ell} + \sum_{\ell=k_*+1}^p \frac{\sigma_\ell}{2\lambda_*} \\ &\geq \frac{k_*}{2} + \frac{r_1(k_*)}{2b_{k_*}}, \end{aligned}$$

where we defined $b_k := \sigma_k/\sigma_{k+1}$ and

$$r_q(k) := \sum_{\ell>k} \left(\frac{\sigma_\ell}{\sigma_{k+1}} \right)^q. \quad (2.4.2)$$

Also note that the reverse inequality $n \leq 2k_* + 2r_1(k_*)$ can be proved along the same line, as long as $\lambda \leq \lambda_*/2$.

In many situations of interest eigenvalues become less spaced as $k \rightarrow \infty$, and therefore b_k is bounded by a constant. For instance, this is the case if $\sigma_k \asymp k^{-\alpha+o_k(1)}$. Further $r_q(k)$ can be regarded as a measure of the number of eigenvalues of the same order as σ_{k+1} , and is therefore an ‘effective rank’ at level k .

Next we bound $\mathcal{V}(\boldsymbol{\Sigma})$. Recalling that $\text{Tr}(\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \lambda_* \mathbf{I})^{-2}) \leq n(1 - c_*^{-1})$, we have

$$\begin{aligned} \mathcal{V}(\boldsymbol{\Sigma}) &\leq \frac{c_* \tau^2}{n} \cdot \left(\sum_{\ell=1}^{k_*} \frac{\sigma_\ell^2}{(\sigma_\ell + \lambda_*)^2} + \sum_{\ell=k_*+1}^p \frac{\sigma_\ell^2}{(\sigma_\ell + \lambda_*)^2} \right) \\ &\leq \frac{c_* \tau^2}{n} \cdot \left(k_* + \sum_{\ell=k_*+1}^d \frac{\sigma_\ell^2}{\lambda_*^2} \right) \\ &\leq c_* \tau^2 \left(\frac{k_*}{n} + \frac{r_2(k_*)}{n} \right) \\ &\leq c_* \tau^2 \left(\frac{k_*}{n} + \frac{4b_{k_*}^2 n}{\bar{r}(k_*)} \right), \end{aligned}$$

where in the last step we defined

$$\bar{r}(k) := \frac{r_1(k)^2}{r_2(k)}, \quad (2.4.3)$$

and used the bound (derived above) $n \geq r_1(k_*)/(2b_{k_*})$.

We proceed similarly for the bias, namely:

$$\begin{aligned} \mathcal{B}(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*) &\leq c_* \sum_{\ell=1}^d \frac{\lambda_*^2 \sigma_\ell}{(\sigma_\ell + \lambda_*)^2} \langle \boldsymbol{\beta}_*, \mathbf{v}_\ell \rangle^2 \\ &\leq c_* \left(\sum_{\ell=1}^{k_*} \lambda_*^2 \sigma_\ell^{-1} \langle \boldsymbol{\beta}_*, \mathbf{v}_\ell \rangle^2 + \sum_{\ell=k_*+1}^d \sigma_\ell \langle \boldsymbol{\beta}_*, \mathbf{v}_\ell \rangle^2 \right) \\ &\leq c_* \left(\sigma_{k_*}^2 \|\boldsymbol{\beta}_{*, \leq k_*}\|_{\boldsymbol{\Sigma}^{-1}}^2 + \|\boldsymbol{\beta}_{*, > k_*}\|_{\boldsymbol{\Sigma}}^2 \right). \end{aligned}$$

We summarize these bounds in the statement below.

Proposition 2.4.1. Define $r_q(k)$ via Eq. (2.4.2), $\bar{r}(k)$ via Eq. (2.4.3), $k_* := \max\{k : \sigma_k \geq \lambda_*\}$, and $b_k := \sigma_k/\sigma_{k+1}$.

Under condition (2.4.1), we have $n \geq k_*/2 + r_1(k_*)/(2b_{k_*})$ and (for $\lambda \leq \lambda_*/2$) $n \leq 2k_* + 2r_1(k_*)$. Further

$$\mathcal{V}(\boldsymbol{\Sigma}) \leq c_* \tau^2 \left(\frac{k_*}{n} + \frac{r_2(k_*)}{n} \right) \leq c_* \tau^2 \left(\frac{k_*}{n} + \frac{4b_{k_*}^2 n}{\bar{r}(k_*)} \right), \quad (2.4.4)$$

$$\mathcal{B}(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*) \leq c_* \left(\sigma_{k_*}^2 \|\boldsymbol{\beta}_{*, \leq k_*}\|_{\boldsymbol{\Sigma}^{-1}}^2 + \|\boldsymbol{\beta}_{*, > k_*}\|_{\boldsymbol{\Sigma}}^2 \right). \quad (2.4.5)$$

As mentioned above, bounds of this form¹ were first proven in [BLLT20, TB20]. These bounds have an interesting consequence. They allow us to characterize pairs $\boldsymbol{\Sigma}, \boldsymbol{\beta}_*$ (with $p = \infty$) for which min-norm interpolation (the $\lambda = 0+$ limit of ridge regression) is ‘consistent’ even if $\tau > 0$. In other words, two things are happening at the same time:

1. The fitted model perfectly interpolates the train data (the train error vanishes).
2. The excess test error vanishes as $n \rightarrow \infty$. (In statistics language, model is consistent.)

When these two elements occur together, we speak of benign overfitting.

Proposition 2.4.1 allows to determine sufficient conditions for benign overfitting. We will assume for simplicity $\sigma_k \rightarrow 0$ as $k \rightarrow \infty$ and b_k bounded.

Note that the condition $n \leq 2k_* + 2r_1(k_*)$ implies $k_* \rightarrow \infty$ as $n \rightarrow \infty$. Hence, for the bias to vanish it is sufficient that $\|\boldsymbol{\beta}\|_{\boldsymbol{\Sigma}^{-1}} < \infty$. Indeed, this implies $\mathcal{B}(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*) \leq c_*(\sigma_{k_*}^2 \|\boldsymbol{\beta}_*\|_{\boldsymbol{\Sigma}^{-1}}^2 + \sigma_{k_*+1}^2 \|\boldsymbol{\beta}_*\|_{\boldsymbol{\Sigma}^{-1}}^2) \rightarrow 0$. Summarizing

$$\|\boldsymbol{\beta}_*\|_{\boldsymbol{\Sigma}^{-1}}^2 < \infty \Rightarrow \mathcal{B}(\boldsymbol{\Sigma}, \boldsymbol{\beta}_*) \rightarrow 0. \quad (2.4.6)$$

For instance, if $\langle \boldsymbol{\beta}_*, \mathbf{v}_k \rangle \neq 0$ only for finitely many k , then this condition is obviously satisfied. More generally, it conveys the intuition that $\boldsymbol{\beta}_*$ should be mostly aligned with the top eigendirections of $\boldsymbol{\Sigma}$. The number of eigendirections one should take into account diverges with n .

Next consider the variance $\mathcal{V}(\boldsymbol{\Sigma})$. Clearly, in order for the bound in Eq. (2.4.4) to vanish, the following conditions must be satisfied:

$$\frac{k_*}{n} \rightarrow 0, \quad \frac{\bar{r}(k_*)}{n} \rightarrow \infty. \quad (2.4.7)$$

In order to get some intuition, let us start by considering the case of polynomially decaying eigenvalues $\sigma_k \asymp k^{-\alpha}$. We need to take $\alpha > 1$ to ensure $\|\mathbf{x}_i\|^2 < \infty$ almost surely. We then have, for $q \geq 1$,

$$r_q(k) \asymp \sum_{\ell > k} \left(\frac{k}{\ell} \right)^{q\alpha} \asymp k,$$

whence

$$k_*(n) \asymp n, \quad \bar{r}(k_*) \asymp n.$$

¹There are some technical differences between the present statement and the results of [BLLT20, TB20]. We refer to [CM22] for a discussion of the differences.

Hence the conditions (2.4.7) for the vanishing of variance do not hold in this case.

We insist, and consider a more slowly decaying sequence of eigenvalues $\sigma_k \asymp k^{-1}(\log k)^{-\beta}$, $\beta > 1$. In this case

$$\begin{aligned} r_q(k) &\asymp k^q(\log k)^{q\beta} \sum_{\ell > k} \frac{1}{\ell^q(\log \ell)^{q\beta}} \\ &\asymp k^q(\log k)^{q\beta} \int_k^\infty \frac{1}{x^q(\log x)^{q\beta}} dx \\ &\asymp k^q(\log k)^{q\beta} \int_{\log k}^\infty \frac{e^{-(q-1)t}}{t^{q\beta}} dt. \end{aligned}$$

Therefore

$$r_q(k) \asymp \begin{cases} k \log k & \text{if } q = 1, \\ k & \text{if } q > 1, \end{cases}$$

whence $\bar{r}(k) \asymp k(\log k)^2$. Using the bounds for $k_*(n)$ in Proposition 2.4.1, we conclude that

$$k_*(n) \asymp \frac{n}{\log n}, \quad \bar{r}(k_*) \asymp n \log n.$$

The conditions of Eq. (2.4.7) are therefore satisfied in this case.

Chapter 3

Kernel ridge regression in high dimension

In this chapter we consider linearized neural networks in the infinite-width limit $N \rightarrow \infty$. There are two reasons for beginning our analysis from this limit case:

1. In the $N \rightarrow \infty$ limit, the ridge regression problem (1.4.9) simplifies to *kernel ridge regression* (KRR) with respect to an inner-product kernel. On one hand, this is a simpler problem than the original one. On the other KRR is an interesting method for its own sake.
2. As we will see in the next two chapters, mildly overparametrized networks in the linear regime behave similarly to their $N = \infty$ limit.

The specific kernel arising by taking the wide limit of neural networks is commonly referred to as *neural tangent kernel (NTK)* [JGH18]. For the sake of clarity we will refer to it as the *infinite width NTK*. In the case of fully connected networks (for any constant depth), there is hardly anything special about the NTK. As we will see, the analysis can be carried out in a unified fashion for any inner product kernel and the behavior is qualitatively independent of the specific kernel under some genericity assumptions.

The goal of this chapter is to obtain a tight characterization of the test error of KRR in high dimension.

3.1 Infinite-width limit and kernel ridge regression

Recall from Section 1.4 that we are interested in the ridge regression estimator

$$\hat{\mathbf{b}}(\lambda) = \operatorname{argmin}_{\mathbf{b} \in \mathbb{R}^p} \left\{ \|\tilde{\mathbf{y}} - \Phi \mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_2^2 \right\}, \quad (3.1.1)$$

where $\Phi = [\phi(\mathbf{x}_1)^\top, \dots, \phi(\mathbf{x}_n)^\top]^\top \in \mathbb{R}^{n \times p}$ is the matrix containing the feature vectors, and $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^p$ is a featurization map. We are particularly interested in the featurization map obtained by linearizing a two-layer neural network, as in Eq. (1.4.6) (in this case $p = N(d+1)$). For simplicity, we will assume $f(\mathbf{x}; \boldsymbol{\theta}_0) = 0$ and replace $\tilde{\mathbf{y}}$ by \mathbf{y} (cf. Remark 1.4.2).

The minimizer of Problem (3.1.1) can be written explicitly as

$$\hat{\mathbf{b}}(\lambda) = (\lambda \mathbf{I}_p + \Phi^\top \Phi)^{-1} \Phi^\top \mathbf{y} = \Phi^\top (\lambda \mathbf{I}_n + \Phi \Phi^\top)^{-1} \mathbf{y}. \quad (3.1.2)$$

The prediction function is given by

$$\hat{f}_\lambda(\mathbf{x}) = \langle \hat{\mathbf{b}}(\lambda), \phi(\mathbf{x}) \rangle = \phi(\mathbf{x})^\top \Phi^\top (\lambda \mathbf{I}_n + \Phi \Phi^\top)^{-1} \mathbf{y}. \quad (3.1.3)$$

Let us introduce the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ defined by $K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle$, where $\langle \mathbf{u}, \mathbf{v} \rangle = u_1 v_1 + \dots + u_p v_p$ denotes the standard euclidean inner-product on \mathbb{R}^p . We further define $\mathbf{K}_n = (K(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n} = \Phi \Phi^\top \in \mathbb{R}^{n \times n}$ the empirical kernel matrix evaluated at the n data points. We can rewrite the prediction function (3.1.3) as

$$\hat{f}_\lambda(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \cdot)^\top (\lambda \mathbf{I}_n + \mathbf{K}_n)^{-1} \mathbf{y}, \quad (3.1.4)$$

where we denoted $\mathbf{K}(\mathbf{x}, \cdot) = (K(\mathbf{x}, \mathbf{x}_1), \dots, K(\mathbf{x}, \mathbf{x}_n))^\top \in \mathbb{R}^n$. Equivalently, the prediction function (3.1.4) corresponds to the kernel ridge regression estimator with kernel K and regularization parameter λ (see Remark 3.1.2 below). In other words, performing ridge regression on a linear model with featurization map ϕ is equivalent to performing kernel ridge regression with kernel $K(\cdot, \cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$.

Remark 3.1.1 (Reproducing kernel Hilbert space). In general, consider a feature map $\phi : \mathbb{R}^d \rightarrow \mathcal{F}$ where \mathcal{F} is a Hilbert space, often called the ‘feature space’, with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ and norm $\|\cdot\|_{\mathcal{F}} = \langle \cdot, \cdot \rangle_{\mathcal{F}}^{1/2}$. Introduce the function space

$$\mathcal{H} := \{h(\cdot) = \langle \boldsymbol{\theta}, \phi(\cdot) \rangle_{\mathcal{F}} : \boldsymbol{\theta} \in \mathcal{F}, \|\boldsymbol{\theta}\|_{\mathcal{F}} < \infty\}. \quad (3.1.5)$$

Then \mathcal{H} is a *reproducing kernel Hilbert space* (RKHS) with reproducing kernel given by $K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle_{\mathcal{F}}$ and RKHS norm

$$\|h\|_{\mathcal{H}} = \inf\{\|\boldsymbol{\theta}\|_{\mathcal{F}} : \boldsymbol{\theta} \in \mathcal{F}, h(\cdot) = \langle \boldsymbol{\theta}, \phi(\cdot) \rangle_{\mathcal{F}}\}. \quad (3.1.6)$$

(Conversely, any RKHS with reproducing kernel K induces a featurization map, e.g. taking $\mathcal{F} = \mathcal{H}$ and $\phi(\mathbf{x}) = K(\mathbf{x}, \cdot)$.)

In the simple case of Eq. (3.1.1), $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^p$, $\mathcal{F} = \mathbb{R}^p$ and the RKHS is simply the finite-dimensional set of linear functions $h(\mathbf{x}) = \langle \mathbf{b}, \phi(\mathbf{x}) \rangle$ with $\mathbf{b} \in \mathbb{R}^p$ and $\|\mathbf{b}\|_2 < \infty$. However, in general, we can take \mathcal{F} to be infinite-dimensional. The RKHS framework is particularly useful because of this flexibility (see next remark). We refer the reader to [BTA11] for a general introduction to the theory of RKHS and kernel methods.

Remark 3.1.2 (Kernel ridge regression). Kernel ridge regression is a general approach to learning that abstracts the specific examples of ridge regression that we studied so far. In a first step, we map the data $\mathbf{x} \mapsto \phi(\mathbf{x})$ into a feature space $(\mathcal{F}, \langle \cdot, \cdot \rangle_{\mathcal{F}})$. We then fit a low-norm linear predictor with respect to this embedding, i.e. $\hat{f}_\lambda(\mathbf{x}) = \langle \hat{\boldsymbol{\theta}}(\lambda), \phi(\mathbf{x}) \rangle_{\mathcal{F}}$ where

$$\hat{\boldsymbol{\theta}}(\lambda) := \operatorname{argmin}_{\boldsymbol{\theta} \in \mathcal{F}} \left\{ \sum_{i=1}^n (y_i - \langle \boldsymbol{\theta}, \phi(\mathbf{x}_i) \rangle_{\mathcal{F}})^2 + \lambda \|\boldsymbol{\theta}\|_{\mathcal{F}}^2 \right\}.$$

From Remark 3.1.1, this is equivalent to the following:

$$\hat{f}_\lambda = \operatorname{argmin}_{f \in \mathcal{H}} \left\{ \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\}, \quad (3.1.7)$$

where \mathcal{H} is the RKHS associated to the feature map ϕ . By the Representer Theorem, the solution (3.1.7) is given explicitly by $\hat{f}_\lambda(\mathbf{x}) = \hat{a}_1(\lambda)K(\mathbf{x}, \mathbf{x}_1) + \dots + \hat{a}_n(\lambda)K(\mathbf{x}, \mathbf{x}_n)$ where K is the reproducing kernel of \mathcal{H} and

$$\hat{\mathbf{a}}(\lambda) = \operatorname{argmin}_{\mathbf{a} \in \mathbb{R}^n} \left\{ \|\mathbf{y} - \mathbf{K}\mathbf{a}\|_2^2 + \lambda \mathbf{a}^\top \mathbf{K}\mathbf{a} \right\} = (\lambda \mathbf{I}_n + \mathbf{K})^{-1} \mathbf{y},$$

with $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j \in [n]}$ the kernel matrix. This is indeed the solution found in Eq. (3.1.4). Note the following important observation: we do not need to evaluate the (potentially infinite-dimensional) feature maps $\phi(\mathbf{x}_i)$ but only their inner-products $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}$ which can often be done efficiently. This is known as the *kernel trick*.

In our case, the feature map is induced by the linearization of a two-layer neural network, $\phi(\mathbf{x}) = [\phi_{\text{RF}}(\mathbf{x}), \phi_{\text{NT}}(\mathbf{x})] \in \mathbb{R}^{N(d+1)}$, where we recall that

$$\phi_{\text{RF}}(\mathbf{x}) = \frac{1}{\sqrt{N}} [\sigma(\langle \mathbf{w}_1, \mathbf{x} \rangle); \dots; \sigma(\langle \mathbf{w}_N, \mathbf{x} \rangle)], \quad (3.1.8)$$

$$\phi_{\text{NT}}(\mathbf{x}) = \frac{1}{\sqrt{Nd}} [\sigma'(\langle \mathbf{w}_1, \mathbf{x} \rangle) \mathbf{x}^\top; \dots; \sigma'(\langle \mathbf{w}_N, \mathbf{x} \rangle) \mathbf{x}^\top]. \quad (3.1.9)$$

The associated kernel is given by

$$K_N(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle = K_N^{\text{RF}}(\mathbf{x}_1, \mathbf{x}_2) + K_N^{\text{NT}}(\mathbf{x}_1, \mathbf{x}_2), \quad (3.1.10)$$

where

$$K_N^{\text{RF}}(\mathbf{x}_1, \mathbf{x}_2) := \langle \phi_{\text{RF}}(\mathbf{x}_1), \phi_{\text{RF}}(\mathbf{x}_2) \rangle = \frac{1}{N} \sum_{i=1}^N \sigma(\langle \mathbf{w}_i, \mathbf{x}_1 \rangle) \sigma(\langle \mathbf{w}_i, \mathbf{x}_2 \rangle), \quad (3.1.11)$$

$$K_N^{\text{NT}}(\mathbf{x}_1, \mathbf{x}_2) := \langle \phi_{\text{NT}}(\mathbf{x}_1), \phi_{\text{NT}}(\mathbf{x}_2) \rangle = \frac{1}{Nd} \sum_{i=1}^N \langle \mathbf{x}_1, \mathbf{x}_2 \rangle \sigma'(\langle \mathbf{w}_i, \mathbf{x}_1 \rangle) \sigma'(\langle \mathbf{w}_i, \mathbf{x}_2 \rangle). \quad (3.1.12)$$

These kernels are random because of the random weights \mathbf{w}_i , and are finite-dimensional, with rank at most N and Nd respectively.

We draw $\mathbf{w}_1, \dots, \mathbf{w}_N$ i.i.d. from a common distribution ν on \mathbb{R}^d . As the number of neurons goes to infinity, both kernels converge pointwise to their expectations by law of large numbers

$$K_N^{\text{RF}}(\mathbf{x}_1, \mathbf{x}_2) \rightarrow K^{\text{RF}}(\mathbf{x}_1, \mathbf{x}_2), \quad K_N^{\text{NT}}(\mathbf{x}_1, \mathbf{x}_2) \rightarrow K^{\text{NT}}(\mathbf{x}_1, \mathbf{x}_2), \quad (3.1.13)$$

where (with $\mathbf{w} \sim \nu$)

$$K^{\text{RF}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbb{E}_{\mathbf{w}} \left\{ \sigma(\langle \mathbf{w}, \mathbf{x}_1 \rangle) \sigma(\langle \mathbf{w}, \mathbf{x}_2 \rangle) \right\}, \quad (3.1.14)$$

$$K^{\text{NT}}(\mathbf{x}_1, \mathbf{x}_2) = \frac{1}{d} \langle \mathbf{x}_1, \mathbf{x}_2 \rangle \mathbb{E}_{\mathbf{w}} \left\{ \sigma'(\langle \mathbf{w}, \mathbf{x}_1 \rangle) \sigma'(\langle \mathbf{w}, \mathbf{x}_2 \rangle) \right\}. \quad (3.1.15)$$

Let us emphasize that the pointwise convergence of Eqs. (3.1.13) does not provide any quantitative control on how large N has to be for finite- N ridge regression to behave similarly to $N = \infty$ ridge regression. This question will be addressed in the next two chapters.

We will consider either $\mathbf{w} \sim \text{Unif}(\mathbb{S}^{d-1})$ (with $\mathbb{S}^{d-1} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_2 = 1\}$ the unit sphere in d dimensions) or $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_d/d)$. These correspond to standard initializations used in neural

networks and both are very close to each other for $d \gg 1$. Since the distribution of \mathbf{w} is invariant under rotations in \mathbb{R}^d , the kernels K^{RF} and K^{NT} are also invariant under rotations and can therefore be written as functions of $\|\mathbf{x}_1\|_2$, $\|\mathbf{x}_2\|_2$ and $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$.

We will assume hereafter that the data is normalized with $\|\mathbf{x}\|_2 = \sqrt{d}$. Then we can write

$$K_N^{\text{RF}}(\mathbf{x}_1, \mathbf{x}_2) = h_{\text{RF}}^{(d)} \left(\frac{\langle \mathbf{x}_1, \mathbf{x}_2 \rangle}{d} \right), \quad K_N^{\text{NT}}(\mathbf{x}_1, \mathbf{x}_2) = h_{\text{NT}}^{(d)} \left(\frac{\langle \mathbf{x}_1, \mathbf{x}_2 \rangle}{d} \right). \quad (3.1.16)$$

Kernels that only depend on the inner product of their inputs are also called *inner-product* or *dot-product* kernels.

Remark 3.1.3. Let us comment on the normalization choice in these notes. First, for $\|\mathbf{x}\|_2 = \sqrt{d}$ and $\mathbf{w}_i \sim \text{Unif}(\mathbb{S}^{d-1})$ or $\text{N}(0, \mathbf{I}_d/d)$, the input $\langle \mathbf{w}_i, \mathbf{x} \rangle$ to the non-linearity σ has variance of order 1. This is the correct behavior for the activation function to behave in a nontrivial manner. Second, we allow the kernels $h_{\text{RF}}^{(d)}$ and $h_{\text{NT}}^{(d)}$ to depend on the dimension, and hence the scaling of the argument $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle/d$ is somewhat arbitrary. However, the specific choice of Eq. (3.1.16) is motivated by the fact that, with this scaling the functions $h_{\text{RF}}^{(d)}$, $h_{\text{NT}}^{(d)}$ converge to a well defined limit as $d \rightarrow \infty$. More precisely:

- 1) For $\mathbf{w} \sim \text{N}(0, \mathbf{I}_d/d)$, $h_{\text{RF}}^{(d)}$ and $h_{\text{NT}}^{(d)}$ are independent of d and are given by (with $(G_1, G_2) \sim \text{N}(0, \mathbf{I}_2)$)

$$h_{\text{RF}}(t) = \mathbb{E}_{G_1, G_2} \left\{ \sigma(G_1) \sigma(tG_1 + \sqrt{1-t^2}G_2) \right\}, \quad (3.1.17)$$

$$h_{\text{NT}}(t) = t \mathbb{E}_{G_1, G_2} \left\{ \sigma'(G_1) \sigma'(tG_1 + \sqrt{1-t^2}G_2) \right\}, \quad (3.1.18)$$

where the expectation is taken with respect to $(G_1, G_2) \sim \text{N}(0, \mathbf{I}_2)$.

- 2) For $\mathbf{w} \sim \text{Unif}(\mathbb{S}^{d-1})$, $h_{\text{RF}}^{(d)}$ and $h_{\text{NT}}^{(d)}$ are nearly independent of d . Namely, because of the concentration of the norm of Gaussian random vectors, we have $h_{\text{RF}}^{(d)} \rightarrow h_{\text{RF}}$ and $h_{\text{NT}}^{(d)} \rightarrow h_{\text{NT}}$ as $d \rightarrow \infty$, the same kernels as for isotropic Gaussian weights.

Remark 3.1.4. More generally, consider a multilayer fully-connected neural network defined as

$$f(\mathbf{x}; \boldsymbol{\theta}) := \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\mathbf{W}_{L-2} \cdots \sigma(\mathbf{W}_1 \mathbf{x}) \cdots)), \quad (3.1.19)$$

where the parameters are $\boldsymbol{\theta} = (\mathbf{W}_1, \dots, \mathbf{W}_L)$ with $\mathbf{W}_L \in \mathbb{R}^{1 \times d_L}$, $\mathbf{W}_\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$ and $d_1 = d$. If $(\mathbf{W}_\ell)_{ij} \sim_{iid} \text{N}(0, \tau_\ell^2)$, then the expected kernel

$$K^{\text{NT},L}(\mathbf{x}_1, \mathbf{x}_2) := \mathbb{E}_{\boldsymbol{\theta}} \left\{ \langle \nabla_{\boldsymbol{\theta}} f(\mathbf{x}_1; \boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} f(\mathbf{x}_2; \boldsymbol{\theta}) \rangle \right\}, \quad (3.1.20)$$

is rotationally invariant. In particular, if the inputs are normalized $\|\mathbf{x}_i\|_2 = \sqrt{d}$, then the kernel must take the form

$$K^{\text{NT},L}(\mathbf{x}_1, \mathbf{x}_2) = h_{\text{NT},L}^{(d)} \left(\frac{\langle \mathbf{x}_1, \mathbf{x}_2 \rangle}{d} \right). \quad (3.1.21)$$

Similarly to the two-layer case, if we scale the parameters $(\mathbf{W}_\ell)_{ij} \sim_{iid} \text{N}(0, \tau_\ell^2/d_\ell)$ and take $\min_{\ell=1, \dots, L} d_\ell \rightarrow \infty$, we have pointwise convergence

$$K_N^{\text{NT},L}(\mathbf{x}_1, \mathbf{x}_2) := \langle \nabla_{\boldsymbol{\theta}} f(\mathbf{x}_1; \boldsymbol{\theta}), \nabla_{\boldsymbol{\theta}} f(\mathbf{x}_2; \boldsymbol{\theta}) \rangle \rightarrow K^{\text{NT},L}(\mathbf{x}_1, \mathbf{x}_2). \quad (3.1.22)$$

This limiting kernel is referred to as the *neural tangent kernel* [JGH18].

3.2 Test error of KRR in the polynomial high-dimensional regime

In the previous section, we saw that under suitable distribution of the weights, ridge regression with large enough network size can be approximated by kernel ridge regression with an inner-product kernel. In the following, we characterize the risk of KRR for general inner-product kernels (in particular the neural tangent kernel with any number of layers by Remark 3.1.4). We defer to Chapters 4 and 5 the important question of how many neurons is needed for the finite-width networks to have similar risk as their infinite-width limits.

Throughout this section we consider an isotropic model for the distribution of the covariates $\mathbf{x}_i \in \mathbb{R}^d$. We assume to be given i.i.d. samples $\{(y_i, \mathbf{x}_i)\}_{i \leq n}$ with

$$y_i = f_\star(\mathbf{x}_i) + \varepsilon_i, \quad \mathbf{x}_i \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d})), \quad (3.2.1)$$

where $f_\star \in L^2 := L^2(\mathbb{S}^{d-1}(\sqrt{d}))$ is a general square-integrable function on the sphere of radius \sqrt{d} , i.e. $\mathbb{E}\{f_\star(\mathbf{x})^2\} < \infty$, and the noise ε_i is independent of \mathbf{x}_i , with $\mathbb{E}\{\varepsilon_i\} = 0$, $\mathbb{E}\{\varepsilon_i^2\} = \tau^2$.

We will consider a general rotationally invariant kernel $K(\mathbf{x}_1, \mathbf{x}_2) = h(\langle \mathbf{x}_1, \mathbf{x}_2 \rangle / d)$, where we take $h : [-1, +1] \rightarrow \mathbb{R}$ independent of the dimension. We further assume h to be ‘generic’, meaning that in the basis of Hermite polynomials $\{\text{He}_k\}_{k \geq 0}$,

$$h(t/\sqrt{d}) = \sum_{k=0}^{\infty} c_{d,k} \text{He}_k(t), \quad c_{d,k} := \frac{1}{k!} \mathbb{E}_{G \sim \mathcal{N}(0,1)}[\text{He}_k(G)h(G/\sqrt{d})], \quad (3.2.2)$$

has all its Hermite coefficients¹ satisfying $d^{k/2}c_{d,k} \rightarrow c_k > 0$ as $d \rightarrow \infty$ for all $k \geq 0$. This corresponds to a universality condition: if $c_k = 0$, then the KRR estimator will not fit degree- k spherical components of the target function, no matter the number of samples.

Several of the assumptions above have been partially relaxed in the literature, and we will provide pointers below. However, for clarity of exposition, we consider the simplest possible setting.

We are interested in the KRR estimator \hat{f}_λ with kernel h as described above. Recall that \hat{f}_λ is given by (see Eq. (3.1.4))

$$\hat{f}_\lambda(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \cdot)^\top (\lambda \mathbf{I}_n + \mathbf{K}_n)^{-1} \mathbf{y}. \quad (3.2.3)$$

We are interested in the excess test error under square loss which we denote

$$R_{\text{KRR}}(f_\star; \mathbf{X}, \mathbf{y}, \lambda) := \mathbb{E}_{\mathbf{x}} \left\{ (f_\star(\mathbf{x}) - \hat{f}_\lambda(\mathbf{x}))^2 \right\}. \quad (3.2.4)$$

The next theorem characterizes the risk of KRR up to a vanishing constant in the high-dimensional polynomial regime. For $\ell \in \mathbb{N}$, we denote by $\mathbf{P}_{\leq \ell} : L^2 \rightarrow L^2$ the orthogonal projector onto the subspace of polynomials of degree at most ℓ , $\mathbf{P}_{> \ell} := \mathbf{I} - \mathbf{P}_{\leq \ell}$ and $\mathbf{P}_\ell := \mathbf{P}_{\leq \ell} \mathbf{P}_{> \ell - 1}$. Further, $o_{d,\mathbb{P}}$ will denote the standard little-o in probability: $h_1(d) = o_{d,\mathbb{P}}(h_2(d))$ if $h_1(d)/h_2(d)$ converges to zero in probability.

Theorem 2. *Assume the data $\{(y_i, \mathbf{x}_i)\}_{i \leq n}$ are distributed according to model (4.2.1) and the kernel function h satisfies the genericity condition (3.2.2).*

¹The normalization is not important, but to be definite, we choose the standard one $\mathbb{E}[\text{He}_j(G)\text{He}_k(G)] = k! \mathbb{1}_{j=k}$ ($G \sim \mathcal{N}(0, 1)$).

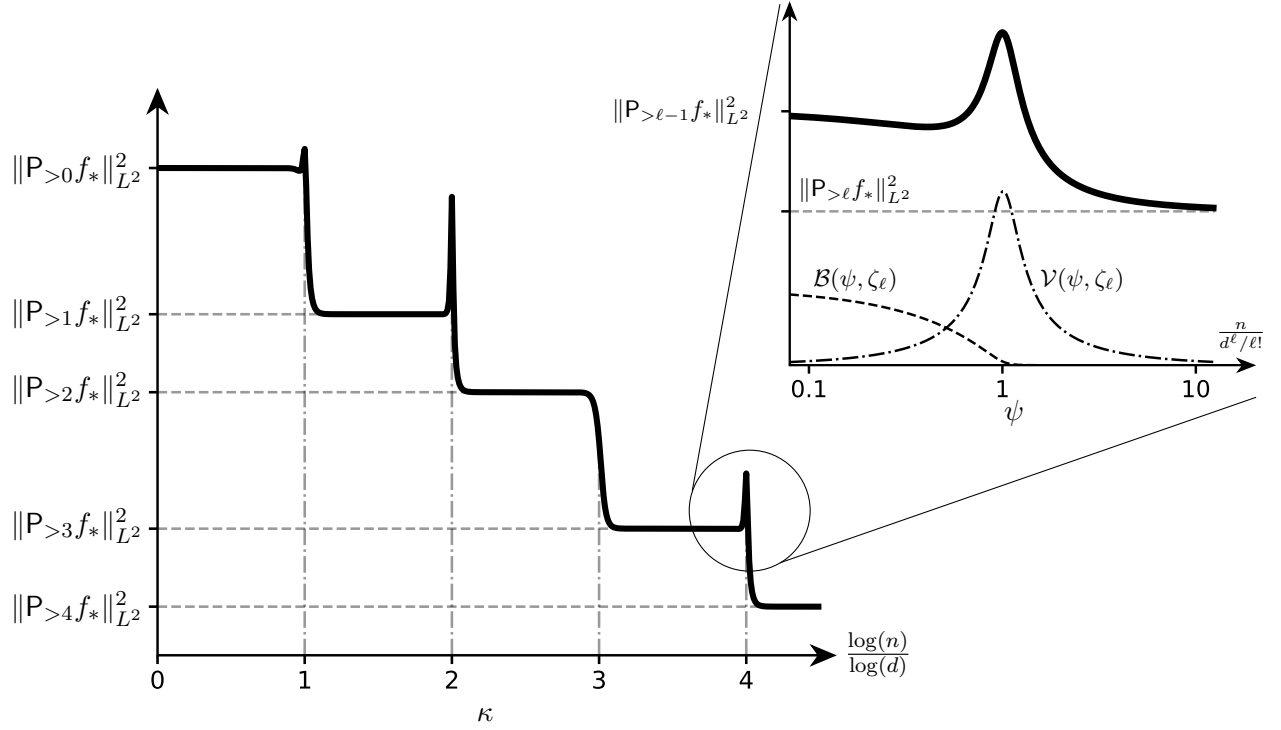


Figure 3.1: A cartoon illustration of the test error of KRR in the polynomial high-dimensional scaling $n/d^\kappa \rightarrow \psi$, as $n, d \rightarrow \infty$, for any $\kappa, \psi \in \mathbb{R}_{>0}$. The test error follows a staircase, with peaks that can occur at each $\kappa = \ell \in \mathbb{N}$, depending on the effective regularization ζ_ℓ and effective signal-to-noise ratio SNR_ℓ at that scale.

- (a) If $d^{\ell+\delta} \leq n \leq d^{\ell+1-\delta}$ for some integer ℓ and constant $\delta > 0$, then there exists a constant $\lambda_\star = \Theta_d(1)$ such that for any regularization parameter $\lambda \in [0, \lambda_\star]$, we have (cf. Eq. (3.2.4))

$$R_{\text{KRR}}(f_\star; \mathbf{X}, \mathbf{y}, \lambda) = \|P_{>\ell} f_\star\|_{L^2}^2 + o_{d, \mathbb{P}}(1) \cdot (\|f_\star\|_{L^2}^2 + \tau^2). \quad (3.2.5)$$

Furthermore, no kernel method with dot-product kernel can do better (i.e. have smaller risk).

- (b) If $n/(d^\ell/\ell!) \rightarrow \psi$ for some integer ℓ and constant $\psi > 0$, then denoting $\zeta_\ell = (\lambda + c_{>\ell})/c_\ell$ the effective regularization at level ℓ with $c_{>\ell} = \sum_{k>\ell} c_k$, we have

$$R_{\text{KRR}}(f_\star; \mathbf{X}, \mathbf{y}, \lambda) = \|P_\ell f_\star\|_{L^2}^2 \cdot \mathcal{B}(\psi, \zeta_\ell) + (\|P_{>\ell} f_\star\|_{L^2}^2 + \tau^2) \cdot \mathcal{V}(\psi, \zeta_\ell) + \|P_{>\ell} f_\star\|_{L^2}^2 + o_{d, \mathbb{P}}(1) \cdot (\|f_\star\|_{L^2}^2 + \tau^2), \quad (3.2.6)$$

where the definitions of $\mathcal{B}(\psi, \zeta_\ell)$ and $\mathcal{V}(\psi, \zeta_\ell)$ can be found in [Mis22].

Remark 3.2.1. Part (a) of the above theorem was proven in [GMMM21] and was later generalized to other RKHS in [MMM22] under a ‘spectral gap’ assumption. Part (b) was proven in [Mis22, HL22a, XHM⁺22].

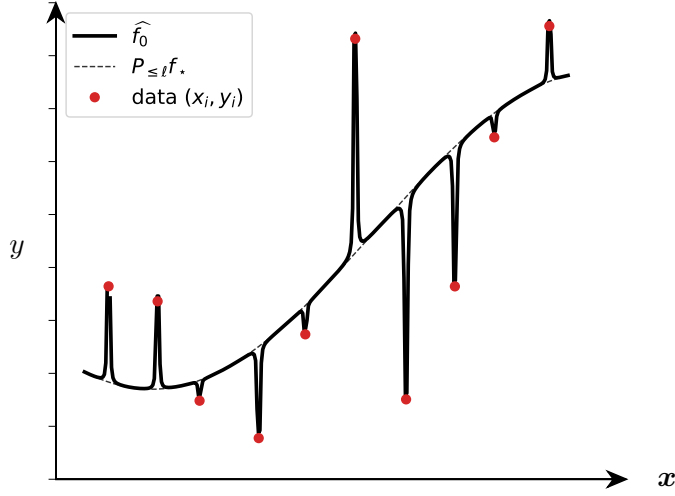


Figure 3.2: A cartoon illustration of the benign overfitting phenomenon in kernel ridgeless regression. This cartoon is supposed to capture the qualitative behavior of kernel min-norm interpolators in high dimension. The interpolator decomposes in the sum of a smooth part, that captures the best low-degree approximation of the target, and a spiky part that interpolates the noisy data.

In words, Eq. (3.2.5) implies that \hat{f}_λ only fits the projection of f_\star onto low degree polynomials: if $d^\ell \ll n \ll d^{\ell+1}$, KRR learns the best degree- ℓ polynomial approximation to f_\star and nothing else. Equivalently, we can decompose

$$\hat{f}_\lambda = P_{\leq \ell} f_\star + \Delta, \quad (3.2.7)$$

where $\|\Delta\|_{L^2} = (\|f_\star\|_{L^2} + \tau^2) \cdot o_{d, \mathbb{P}}(1)$ does not contribute to the test error. For example, this theorem implies that if $n \leq d^{1.99}$, KRR can only fit the linear component of f_\star . Each time $\log(n)/\log(d)$ crosses an integer value, KRR learns one more degree polynomial approximation: the risk presents a *staircase decay* when n increases, with peaks that can occur at $n = d^\ell/\ell!$. This phenomena is illustrated with a cartoon in Figure 3.1.

As mentioned above, the conclusions of Theorem 2 apply to any rotationally invariant kernel (under the genericity assumption). Notably, they apply to the neural tangent kernel of fully-connected networks of any depth (as explained in Remark 3.1.4). In the linear regime and in high dimension, depth does not appear to play an important role for fully-connected neural networks.

Remark 3.2.2 (Benign Overfitting). As discussed in Section 1.3, one particularly interesting solution is the case $\lambda = 0+$:

$$\hat{f}_0 := \operatorname{argmin}_{f \in \mathcal{H}} \left\{ \|f\|_{\mathcal{H}} : f(\mathbf{x}_i) = y_i, \forall i \in [n] \right\}, \quad (3.2.8)$$

which corresponds to the *minimum-norm interpolating solution*. In this case, the KRR solution perfectly interpolates the noisy data $y_i = f_\star(\mathbf{x}_i) + \varepsilon_i$. Problem (3.2.8) is sometimes referred to as kernel ‘ridgeless’ regression following [LR20].

Theorem 2 establishes that KRR with $\lambda = 0+$ is optimal among all kernel methods when $d^\ell \ll n \ll d^{\ell+1}$, in the sense that it achieves the best possible test error. This is another example of *benign overfitting* where interpolation does not harm generalization, as described in Section 2.4. Recalling Eq. (3.2.7), we can decompose $\hat{f}_0 = P_{\leq \ell} f_\star + \Delta$ where

- 1) $P_{\leq \ell} f_\star$ is a smooth component good for prediction;
- 2) Δ is a spiky component useful for interpolation, $\Delta(\mathbf{x}_i) = P_{> \ell} f_\star(\mathbf{x}_i) + \varepsilon_i$, but which does not contribute to the test error, since $\|\Delta\|_{L^2} \ll 1$.

We illustrate benign overfitting in kernel regression with a cartoon in Figure 3.2.

Remark 3.2.3 (Multiple peaks in the risk curve). For $n \asymp d^\ell$, KRR transitions from not fitting the degree- ℓ polynomial components at all for $n \ll d^\ell$ to fitting all degree- ℓ polynomials when $n \gg d^\ell$. The behavior between these two regimes, for $n \asymp d^\ell$, is more complex. Because of the degeneracy of the kernel operator eigenvalues at that scale, the spectrum of the kernel matrix follows a shifted and rescaled Marchenko-Pastur distribution. This can lead to a peak in the risk curve, whenever the effective regularization ζ_ℓ or the effective signal-to-noise ratio $\text{SNR}_\ell := \|P_\ell f_\star\|_{L^2}^2 / (\|P_{> \ell} f_\star\|_{L^2}^2 + \tau^2)$ are small enough. We refer the interested reader to [Mis22] for more details.

3.3 Diagonalization of inner-product kernels on the sphere

In the next section we will present some ideas of the proof of Theorem 2. Before doing that, it is useful to provide some background on the functional space $L^2(\mathbb{S}^{d-1}(\sqrt{d}))$ of square-integrable functions on the sphere. For a more in-depth presentation, we refer to [Sze39, Chi11, EF14].

We use the orthogonal decomposition

$$L^2(\mathbb{S}^{d-1}(\sqrt{d})) = \bigoplus_{k=0}^{\infty} \mathcal{V}_{d,k}, \quad (3.3.1)$$

where $\mathcal{V}_{d,\ell}$ is the space of degree- ℓ polynomials orthogonal to $\bigoplus_{k=0}^{\ell-1} \mathcal{V}_{d,k}$ (orthogonality is with respect to the L^2 inner-product $\langle f, g \rangle_{L^2} = \mathbb{E}\{f(\mathbf{x})g(\mathbf{x})\}$). Equivalently, $\mathcal{V}_{d,k}$ is the space of all spherical harmonics of degree k , and has dimension

$$\dim(\mathcal{V}_{d,k}) = B(d, k) := \frac{2k + d - 2}{k} \binom{k + d - 3}{k - 1}. \quad (3.3.2)$$

Note that $B(d, k) = (d^k/k!)(1 + o_d(1))$. Consider $\{Y_{ks}\}_{s \leq B(d,k)}$ an orthonormal basis of $\mathcal{V}_{d,k}$. In particular, we have $\langle Y_{ku}, Y_{\ell v} \rangle_{L^2} = \delta_{k\ell} \delta_{uv}$, and the set $\{Y_{ks}\}_{k \geq 0, s \leq B(d,k)}$ forms a complete orthonormal basis of $L^2(\mathbb{S}^{d-1}(\sqrt{d}))$.

It will be convenient to introduce the following notations: $\mathcal{V}_{d, \leq \ell} = \bigoplus_{k=0}^{\ell} \mathcal{V}_{d,k}$ the space of all polynomials of degree at most ℓ , and its orthogonal complement $\mathcal{V}_{d, > \ell} = \bigoplus_{k=\ell+1}^{\infty} \mathcal{V}_{d,k}$. Further we denote by P_k the orthogonal projection on $\mathcal{V}_{d,k}$, by $P_{\leq \ell} = P_0 + \dots + P_\ell$ the orthogonal projection on $\mathcal{V}_{d, \leq \ell}$, and $P_{> \ell} = \mathbf{I} - P_{\leq \ell}$ the orthogonal projection on $\mathcal{V}_{d, > \ell}$.

Consider now a rotationally invariant kernel defined on $\mathbb{S}^{d-1}(\sqrt{d})$, i.e. a positive semidefinite kernel² $K : \mathbb{S}^{d-1}(\sqrt{d}) \times \mathbb{S}^{d-1}(\sqrt{d}) \rightarrow \mathbb{R}$ such that $K(\mathbf{x}_1, \mathbf{x}_2) = h(\langle \mathbf{x}_1, \mathbf{x}_2 \rangle / d)$ for some mea-

²This means a measurable function such that $\sum_{i,j \leq m} K(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j \geq 0$ for any m and any collection of points $(\mathbf{x}_i)_{i \leq m}$, and weights $(\alpha_i)_{i \leq m}$.

surable function $h : [-1, 1] \rightarrow \mathbb{R}$. To the kernel function K , we associate its integral operator $\mathbb{K} : L^2(\mathbb{S}^{d-1}(\sqrt{d})) \rightarrow L^2(\mathbb{S}^{d-1}(\sqrt{d}))$ defined by

$$\mathbb{K}f(\mathbf{x}_1) = \mathbb{E}_{\mathbf{x}_2 \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))} \{h(\langle \mathbf{x}_1, \mathbf{x}_2 \rangle / d) f(\mathbf{x}_2)\}. \quad (3.3.3)$$

The proof of Theorem 2 relies on the eigendecomposition of inner-product kernels on the sphere. We know that by the spectral theorem for compact operators, the kernel function K can be diagonalized as

$$K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{\infty} \lambda_j^2 \psi_j(\mathbf{x}_1) \psi_j(\mathbf{x}_2), \quad (3.3.4)$$

where $\{\psi_j\}_{j \geq 1}$ is an orthonormal basis of $L^2(\mathbb{S}^{d-1}(\sqrt{d}))$, and the $\{\lambda_j^2\}_{j \geq 1}$ are the eigenvalues in nonincreasing order $\lambda_1^2 \geq \lambda_2^2 \geq \lambda_3^2 \geq \dots \geq 0$.

By rotational invariance, the subspaces $\mathcal{V}_{d,k}$ are eigenspaces of the kernel operator \mathbb{K} , i.e. we can write

$$\mathbb{K} = \sum_{k=0}^{\infty} \xi_{d,k}^2 \mathbb{P}_k, \quad (3.3.5)$$

where we now denote $\xi_{d,k}^2$ the eigenvalue associated to the eigenspace $\mathcal{V}_{d,k}$.

Since $\{Y_{ks} : s \leq B(d, k)\}$ form an orthonormal basis of $\mathcal{V}_{d,k}$, we have

$$\mathbb{P}_k f = \sum_{s=1}^{B(d,k)} Y_{ks} \langle Y_{ks}, f \rangle_{L^2}. \quad (3.3.6)$$

Equivalently, the projector is represented as an integral operator

$$\mathbb{P}_k f(\mathbf{x}_1) = B(d, k) \cdot \mathbb{E}_{\mathbf{x}_2} \{Q_k^{(d)}(\langle \mathbf{x}_1, \mathbf{x}_2 \rangle) f(\mathbf{x}_2)\}, \quad (3.3.7)$$

$$Q_k^{(d)}(\langle \mathbf{x}_1, \mathbf{x}_2 \rangle) := \frac{1}{B(d, k)} \sum_{s=1}^{B(d,k)} Y_{ks}(\mathbf{x}_1) Y_{ks}(\mathbf{x}_2), \quad (3.3.8)$$

Note that $Q_k^{(d)}(\cdot)$ must be a function of the inner product $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$ by rotational invariance, and must be a polynomial of degree k because so are the $\{Y_{ks}\}$. Indeed, this is a special function known as a Gegenbauer polynomial. The polynomials $\{Q_k^{(d)}\}_{k \geq 0}$ form an orthogonal basis of $L^2([-d, d], \nu_d)$ where ν_d is the distribution of $\sqrt{d}x_1$ with $\mathbf{x} \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$.

The following properties of Gegenbauer polynomials follow from the above representation:

1. $Q_k^{(d)}(d) = \mathbb{E}_{\mathbf{x}} \{Q_k^{(d)}(\langle \mathbf{x}, \mathbf{x} \rangle)\} = B(d, k)^{-1} \sum_{s=1}^{B(d,k)} \mathbb{E}\{Y_{ks}(\mathbf{x}_1)^2\}$. Therefore

$$Q_k^{(d)}(d) = 1. \quad (3.3.9)$$

2. Since projectors are idempotent ($\mathbb{P}_k^2 = \mathbb{P}_k$), it follows that

$$\mathbb{E}_{\mathbf{x}} \{Q_k^{(d)}(\langle \mathbf{x}_1, \mathbf{x} \rangle) Q_k^{(d)}(\langle \mathbf{x}, \mathbf{x}_2 \rangle)\} = \frac{1}{B(d, k)} Q_k^{(d)}(\langle \mathbf{x}_1, \mathbf{x}_2 \rangle). \quad (3.3.10)$$

3. In particular

$$\langle Q_j^{(d)}, Q_k^{(d)} \rangle_{L^2(\tau_d)} = \frac{1}{B(d, k)} \mathbb{1}_{j=k}. \quad (3.3.11)$$

Representing the projection operator in Eq. (3.3.5), we get the following diagonalization of inner-product kernels,

$$K(\mathbf{x}_1, \mathbf{x}_2) = \sum_{k=0}^{\infty} \xi_{d,k}^2 \sum_{s=1}^{B(d,k)} Y_{ks}(\mathbf{x}_1) Y_{ks}(\mathbf{x}_2). \quad (3.3.12)$$

In particular, this means that

$$\xi_{d,k}^2 = \mathbb{E}_{\mathbf{x}} \left\{ Q_k^d(\sqrt{d}x_1) h(x_1/\sqrt{d}) \right\}. \quad (3.3.13)$$

When $d \rightarrow \infty$, the measure ν_d converges weakly to $\mathbf{N}(0, 1)$ and therefore $B(d, k)^{1/2} Q_k^{(d)}$ converges to the degree- k normalized Hermite polynomial $\text{He}_k/\sqrt{k!}$. The ‘genericity condition’ (3.2.2) can be restated as $\xi_{d,k}^2 B(d, k) \rightarrow c_k > 0$ as $d \rightarrow \infty$.

3.4 Proof sketch

In this section, we outline the proof of Eq. (3.2.5) in Theorem 2.(a). The proof crucially depends on controlling the empirical kernel matrix $\mathbf{K}_n = (K(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$. While this is a non-linear (in the data) random matrix, which is usually hard to study, \mathbf{K}_n simplifies in our polynomial high-dimensional regime. We explain below how \mathbf{K}_n can be approximately decomposed into a low-rank matrix (coming from the low-degree polynomials) plus a matrix proportional to the identity (coming from the high-degree non-linear part of the kernel).

Using the diagonalization of inner-product kernels in Eq. (3.3.12), we can decompose the kernel matrix into a low-frequency and a high-frequency component,

$$\mathbf{K}_n = \sum_{k=0}^{\ell} \xi_{d,k}^2 \mathbf{Y}_k \mathbf{Y}_k^{\top} + \sum_{k \geq \ell+1} \xi_{d,k}^2 \mathbf{Y}_k \mathbf{Y}_k^{\top} =: \mathbf{K}_n^{\leq \ell} + \mathbf{K}_n^{> \ell}, \quad (3.4.1)$$

where $\mathbf{Y}_k = (Y_{ks}(\mathbf{x}_i))_{i \leq n, s \leq B(d,k)} \in \mathbb{R}^{n \times B(d,k)}$ is the matrix of degree- k spherical harmonics evaluated at the training data points. We claim that the following properties hold:

Low-frequency part: $\mathbf{K}_n^{\leq \ell}$ has rank at most $B(d, 0) + \dots + B(d, \ell) = \Theta_d(d^\ell)$ much lower than n (recall that we assumed $n \geq d^{\ell+\delta}$). Furthermore, one can show that for any $k \leq \ell$,

$$\|n^{-1} \mathbf{Y}_k^{\top} \mathbf{Y}_k - \mathbf{I}_{B(d,k)}\|_{\text{op}} = o_{d, \mathbb{P}}(1). \quad (3.4.2)$$

Therefore, we have

$$\sigma_{\min}(\mathbf{K}_n^{\leq \ell}) \geq \min_{k \leq \ell} \left\{ \xi_{d,k}^2 \sigma_{\min}(\mathbf{Y}_k \mathbf{Y}_k^{\top}) \right\} = \Theta_{d, \mathbb{P}}(nd^{-\ell}) = \Theta_{d, \mathbb{P}}(d^\delta), \quad (3.4.3)$$

where we used $n \geq d^{\ell+\delta}$. Hence $\mathbf{K}_n^{\leq \ell}$ corresponds to a low rank matrix with diverging eigenvalues along the low-frequency component (degree at most ℓ polynomials) of \mathbf{K}_n .

High-frequency part: $\mathbf{K}_n^{>\ell}$ is approximately proportional to the identity with $\|\mathbf{K}_n^{>\ell} - \gamma \mathbf{I}_n\|_{\text{op}} = o_d(\mathbb{P}(1))$, where

$$\gamma = \sum_{k \geq \ell+1} \xi_{d,k}^2 B(d,k) = \sum_{k \geq \ell+1} c_k + o_d(1). \quad (3.4.4)$$

We are now ready to sketch the proof of Theorem 2.(a). Let us decompose the KRR solution into a low-frequency (the projection on degree- ℓ polynomials) and a high-frequency component:

$$\hat{f}_\lambda(\mathbf{x}) \approx \mathbf{K}(\mathbf{x}, \cdot) \left(\mathbf{K}_n^{\leq \ell} + (\lambda + \gamma) \mathbf{I}_n \right)^{-1} \mathbf{y} = \hat{f}_{\lambda, \leq \ell}(\mathbf{x}) + \hat{f}_{\lambda, > \ell}(\mathbf{x}), \quad (3.4.5)$$

where

$$\hat{f}_{\lambda, \leq \ell}(\mathbf{x}) := \mathbf{P}_{\leq \ell} \hat{f}_\lambda(\mathbf{x}) = \mathbf{K}_{\leq \ell}(\mathbf{x}, \cdot) \left(\mathbf{K}_n^{\leq \ell} + (\lambda + \gamma) \mathbf{I}_n \right)^{-1} \mathbf{y}, \quad (3.4.6)$$

$$\hat{f}_{\lambda, > \ell}(\mathbf{x}) := \mathbf{P}_{> \ell} \hat{f}_\lambda(\mathbf{x}) = \mathbf{K}_{> \ell}(\mathbf{x}, \cdot) \left(\mathbf{K}_n^{\leq \ell} + (\lambda + \gamma) \mathbf{I}_n \right)^{-1} \mathbf{y}. \quad (3.4.7)$$

The risk can be written as the sum of contributions along $\mathbf{P}_{\leq \ell}$ and $\mathbf{P}_{> \ell}$:

$$\mathbb{E} \left\{ (f_\star(\mathbf{x}) - \hat{f}_\lambda(\mathbf{x}))^2 \right\} = \underbrace{\mathbb{E} \left\{ (\mathbf{P}_{\leq \ell} f_\star(\mathbf{x}) - \hat{f}_{\lambda, \leq \ell}(\mathbf{x}))^2 \right\}}_{\text{(I)}} + \underbrace{\mathbb{E} \left\{ (\mathbf{P}_{> \ell} f_\star(\mathbf{x}) - \hat{f}_{\lambda, > \ell}(\mathbf{x}))^2 \right\}}_{\text{(II)}} \quad (3.4.8)$$

We can bound these two terms separately:

Term (I): This term is equivalent to the test error of doing kernel ridge regression with kernel $K_{\leq \ell}$, regularization parameter $\lambda + \gamma$, target function $\mathbf{P}_{\leq \ell} f_\star(\mathbf{x})$ and data $y_i = \mathbf{P}_{\leq \ell} f_\star(\mathbf{x}_i) + \tilde{\varepsilon}_i$ where $\tilde{\varepsilon}_i = \mathbf{P}_{> \ell} f_\star(\mathbf{x}_i) + \varepsilon_i$ (note that the noise $\tilde{\varepsilon}$ is not independent of $\mathbf{P}_{\leq \ell} f_\star(\mathbf{x}_i)$ anymore but is still uncorrelated). The dimension of the target space (i.e. $\mathcal{V}_{d, \leq \ell}$) and the rank of the kernel are now $\Theta_d(d^\ell)$ much smaller than the number of samples n . Hence, term (I) corresponds to the test error of KRR in low-dimension. We can therefore expect (and show) that this error is vanishing:

$$\text{(I)} = \|\mathbf{P}_{\leq \ell} f_\star\|_{L^2}^2 \cdot o_d(\mathbb{P}(1)) + \text{Var}(\tilde{\varepsilon}_i) \cdot o_d(\mathbb{P}(1)) = (\|f_\star\|_{L^2}^2 + \tau^2) \cdot o_d(\mathbb{P}(1)). \quad (3.4.9)$$

Term (II): Similarly to $\mathbf{K}_{> \ell}$, we can show that

$$\left\| n \mathbb{E}_{\mathbf{x}} \left\{ \mathbf{K}_{> \ell}(\mathbf{x}, \cdot)^\top \mathbf{K}_{> \ell}(\mathbf{x}, \cdot) \right\} - \kappa \cdot \mathbf{I}_n \right\|_{\text{op}} = o_d(\mathbb{P}(1)), \quad (3.4.10)$$

with $\kappa = \sum_{k \geq \ell+1} n \xi_{d,k}^4 B(d,k) = \Theta_d(nd^{-\ell-1}) = o_d(1)$ where we used that $n \leq d^{\ell+1-\delta}$. We can therefore bound

$$\begin{aligned} \|\hat{f}_{\lambda, > \ell}\|_{L^2}^2 &\leq \left\| \mathbb{E}_{\mathbf{x}} \left\{ \mathbf{K}_{> \ell}(\mathbf{x}, \cdot)^\top \mathbf{K}_{> \ell}(\mathbf{x}, \cdot) \right\} \right\|_{\text{op}} \left\| \left(\mathbf{K}_n^{\leq \ell} + (\lambda + \gamma) \mathbf{I}_n \right)^{-1} \mathbf{y} \right\|_2^2 \\ &\leq o_d(\mathbb{P}(1)) \cdot \frac{1}{(\lambda + \gamma)^2} \cdot \frac{\|\mathbf{y}\|_2^2}{n} = o_d(\mathbb{P}(1)) \cdot (\|f_\star\|_{L^2}^2 + \tau^2), \end{aligned} \quad (3.4.11)$$

where we used by Markov's inequality that

$$n^{-1} \|\mathbf{y}\|_2^2 = O_d(\mathbb{P}(1)) \cdot \mathbb{E}\{n^{-1} \|\mathbf{y}\|_2^2\} = O_d(\mathbb{P}(1)) \cdot (\|f_\star\|_{L^2}^2 + \tau^2). \quad (3.4.12)$$

Therefore the contribution of the second term is

$$\text{(II)} = \|\mathbf{P}_{> \ell} f_\star\|_{L^2}^2 + (\|f_\star\|_{L^2}^2 + \tau^2) \cdot o_d(\mathbb{P}(1)). \quad (3.4.13)$$

Combining the bounds (3.4.11) and (3.4.13) in Eq. (3.4.8) yields part (a) of Theorem 2.

Remark 3.4.1 (Self-induced regularization). From the proof of Theorem 2, we see that the high-frequency component $\mathbf{K}_n^{>\ell}$ of the kernel matrix concentrates on identity and plays the role of an additive *self-induced ridge regularization* $\gamma > 0$. Hence the effective regularization parameter of the KRR solution is $\lambda + \gamma$ (see for example Eq. (3.4.6)), which is bounded away from zero even when taking $\lambda = 0+$. This explains why kernel ridgeless regression generalizes well even when interpolating noisy training data.

Chapter 4

Random features

The random feature (RF) model is a two-layer neural network in which the first layer weights are not trained, and kept equal to their random initialization. It was initially introduced in [RR07] as a randomized approximation for kernel methods. Here we regard it as a particularly simple limit case of linearized neural networks, cf. Section 1.4.

We will use the RF model as a toy model to explore two important questions about neural networks trained in the linear regime:

- (i) How large should N be for the generalization error of RF to match the error associated with its infinite-width kernel limit ($N \rightarrow \infty$) obtained in Chapter 3?
- (ii) What is the impact on the performance of RF when N is not chosen sufficiently large?

While the RF model presents a simpler structure than the (finite width) neural tangent model, we expect them to display similar behavior, provided we match the number of parameters in the two models. Chapter 5 partly confirms this picture.

4.1 The random feature model

The random feature model is given by

$$f_{\text{RF}}(\mathbf{x}; \mathbf{a}) = \frac{1}{\sqrt{N}} \sum_{i=1}^N a_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle), \quad (4.1.1)$$

where $\mathbf{a} = (a_1, \dots, a_N) \in \mathbb{R}^N$ are the free parameters that are trained, while the weights $(\mathbf{w}_i)_{i \leq N}$ are fixed random $\mathbf{w}_i \sim_{iid} \nu$. Equivalently, the RF model can be seen as training the second layer weights of a two-layer neural network, while keeping the first-layer weights fixed.

Recall the definition of the featurization map $\phi_{\text{RF}} : \mathbb{R}^d \rightarrow \mathbb{R}^N$ given by

$$\phi_{\text{RF}}(\mathbf{x}) = \frac{1}{\sqrt{N}} [\sigma(\langle \mathbf{w}_1, \mathbf{x} \rangle); \dots; \sigma(\langle \mathbf{w}_N, \mathbf{x} \rangle)], \quad (4.1.2)$$

so that $f_{\text{RF}}(\mathbf{x}; \mathbf{a}) = \langle \mathbf{a}, \phi_{\text{RF}}(\mathbf{x}) \rangle$. We are interested in the ridge regression solution

$$\hat{\mathbf{a}}(\lambda) = \operatorname{argmin}_{\mathbf{a} \in \mathbb{R}^N} \left\{ \sum_{i=1}^n (y_i - f_{\text{RF}}(\mathbf{x}_i, \mathbf{a}))^2 + \lambda \|\mathbf{a}\|_2^2 \right\} \quad (4.1.3)$$

$$= \operatorname{argmin}_{\mathbf{a} \in \mathbb{R}^N} \left\{ \|\mathbf{y} - \Phi \mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_2^2 \right\}, \quad (4.1.4)$$

where the design matrix $\Phi = [\phi_{\text{RF}}(\mathbf{x}_1)^\top, \dots, \phi_{\text{RF}}(\mathbf{x}_n)^\top]^\top \in \mathbb{R}^{n \times N}$ has entries $\Phi_{ij} = \frac{1}{\sqrt{N}} \sigma(\langle \mathbf{x}_i, \mathbf{w}_j \rangle)$. Throughout this chapter, we will consider sampling $\mathbf{w}_i \sim_{\text{iid}} \text{Unif}(\mathbb{S}^{d-1})$.

Let us emphasize here the reason why the RF model is much simpler to study than the neural tangent model. The entries of the featurization map (4.1.2) (and the columns of Φ) are iid with respect to the randomness in \mathbf{w}_i . On the other hand, in the case of the neural tangent model, the entries of the featurization map $\phi_{\text{NT}}(\mathbf{x}) = (Nd)^{-1/2} [\sigma'(\langle \mathbf{w}_1, \mathbf{x} \rangle) \mathbf{x}^\top; \dots; \sigma'(\langle \mathbf{w}_N, \mathbf{x} \rangle) \mathbf{x}^\top]$ are only block independent with respect to the randomness in \mathbf{w}_i . Tracking the dependency structure in the design matrix makes the analysis of the neural tangent model harder. See Chapter 5 for an example of such an analysis.

4.2 Test error in the polynomial high-dimensional regime

In Chapter 3, we saw that as $N \rightarrow \infty$, the ridge regression solution converges to the kernel ridge regression solution with inner-product kernel K_{RF} . What is the impact of finite N on the generalization error of the RF model?

We consider the same setting as in Section 3.2. Assume we are given i.i.d. samples $\{(y_i, \mathbf{x}_i)\}_{i \leq n}$ with

$$y_i = f_\star(\mathbf{x}_i) + \varepsilon_i, \quad \mathbf{x}_i \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d})), \quad (4.2.1)$$

where $f_\star \in L^2 := L^2(\mathbb{S}^{d-1}(\sqrt{d}))$ and the noise ε_i is independent of \mathbf{x}_i , with $\mathbb{E}\{\varepsilon_i\} = 0$, $\mathbb{E}\{\varepsilon_i^2\} = \tau^2$.

We consider the RF model as in Eq. (4.1.1). We will assume the following ‘genericity’ condition on the activation function σ : (i) $|\sigma(x)| \leq c_0 \exp(c_1|x|)$ for some constants $c_0, c_1 > 0$, and (ii) for any $k \geq 0$, σ has non-zero Hermite coefficient $\mu_k(\sigma) := \mathbb{E}_G\{\sigma(G)\text{He}_k(G)\} \neq 0$ (where $G \sim \mathcal{N}(0, 1)$).

We are interested in the random feature ridge regression (RFRR) solution

$$\hat{f}_{\text{RF}}(\mathbf{x}; \hat{\mathbf{a}}(\lambda)) = \phi_{\text{RF}}(\mathbf{x})^\top \Phi^\top (\lambda \mathbf{I}_n + \Phi \Phi^\top)^{-1} \mathbf{y}, \quad (4.2.2)$$

and consider the excess test error with square loss which we denote by

$$R_{\text{RF}}(f_\star; \mathbf{X}, \mathbf{y}, \mathbf{W}, \lambda) := \mathbb{E}_{\mathbf{x}} \left\{ (f_\star(\mathbf{x}) - f_{\text{RF}}(\mathbf{x}; \hat{\mathbf{a}}(\lambda)))^2 \right\}. \quad (4.2.3)$$

Theorem 3. *Assume $d^{\ell_1 + \delta} \leq n \leq d^{\ell_1 + 1 - \delta}$, $d^{\ell_2 + \delta} \leq N \leq d^{\ell_2 + 1 - \delta}$, $\max(N/n, n/N) \geq d^\delta$ for some integers ℓ_1, ℓ_2 and constant $\delta > 0$. Denote $\ell = \min(\ell_1, \ell_2)$. Further assume that σ verifies the genericity conditions. Then there exists a constant $\lambda_\star = \Theta_d((N/n) \wedge 1)$ such that for any regularization parameter $\lambda \in [0, \lambda_\star]$ and all $\eta > 0$, we have (cf. Eq. (4.2.3))*

$$R_{\text{RF}}(f_\star; \mathbf{X}, \mathbf{y}, \mathbf{W}, \lambda) = \|\mathbf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d, \mathbb{P}}(1) \cdot (\|f_\star\|_{L^{2+\eta}}^2 + \tau^2). \quad (4.2.4)$$

In this theorem, the number of neurons N and the number of samples n play symmetric roles. The test error of RFRR is determined by the minimum of N and n , as long as N , n and d^ℓ for integers ℓ are well separated.

We can distinguish two regimes:

Underparametrized regime: For $N \ll n$ (less parameters than number of samples), the test error is limited by the number of random features and the approximation error dominates, i.e.,

$$\begin{aligned} R_{\text{RF}}(f_\star; \mathbf{X}, \mathbf{y}, \mathbf{W}, \lambda) &= R_{\text{app}}(f_\star; \mathbf{W}) + o_{d, \mathbb{P}}(1), \\ R_{\text{app}}(f_\star; \mathbf{W}) &:= \inf_{\mathbf{a} \in \mathbb{R}^N} \mathbb{E}_{\mathbf{x}} \left\{ (f_\star(\mathbf{x}) - \hat{f}_{\text{RF}}(\mathbf{x}; \mathbf{a}))^2 \right\}, \end{aligned} \quad (4.2.5)$$

which corresponds to the best fit of f_\star by a linear combination of N random features. If $d^\ell \ll N \ll d^{\ell+1}$, we have $\hat{f}_{\text{RF}} \approx \mathbf{P}_{\leq \ell} f_\star$ and the model fits the best degree- ℓ polynomial approximation to f_\star and nothing else.

To get an intuitive picture of this phenomenon, notice that the model \hat{f}_{RF} is contained in the subspace $\text{span}\{\sigma(\langle \mathbf{w}_i, \cdot \rangle) : i \leq N\} \subset L^2(\mathbb{S}^{d-1}(\sqrt{d}))$ of dimension N . By parameter-counting, in order to approximate the space of degree- ℓ polynomials of dimension $\Theta_d(d^\ell)$, we need the number of parameters to be $\Omega_d(d^\ell)$, which matches the requirement in our theorem.

Overparametrized regime: For $n \ll N$ (more parameters than number of samples), the test error is limited by the number of samples and the statistical error dominates, i.e.,

$$R_{\text{RF}}(f_\star; \mathbf{X}, \mathbf{y}, \mathbf{W}, \lambda) = R_{\text{KRR}}(f_\star; \mathbf{X}, \mathbf{y}, \lambda) + o_{d, \mathbb{P}}(1), \quad (4.2.6)$$

where R_{KRR} is the risk of KRR with kernel K_{RF} (corresponding to the infinite-width limit $N \rightarrow \infty$) as obtained in Theorem 2. If $d^\ell \ll n \ll d^{\ell+1}$, we have $\hat{f}_{\text{RF}} \approx \mathbf{P}_{\leq \ell} f_\star$. Again, this matches the parameter-counting heuristic: the information-theoretic lower bound to learn the space of degree- ℓ polynomials is $\Omega_d(d^\ell)$ samples.

To summarize, denoting $R_{\text{RF}}(n, N)$ the test error achieved by the RF model with n samples and N neurons, as long as n , N and d^ℓ for integers ℓ are well separated, we have

$$R_{\text{RF}}(n, N) = \max\{R_{\text{RF}}(n, \infty), R_{\text{RF}}(\infty, N)\} + o_{d, \mathbb{P}}(1), \quad (4.2.7)$$

where $R_{\text{RF}}(\infty, N)$ corresponds to the approximation error (4.2.5) and $R_{\text{RF}}(n, \infty)$ to the statistical error (4.2.6).

Before sketching the proof of Theorem 3, let us make three further comments:

Optimal overparametrization. In the RF model, we are free to choose the number of neurons N and it is interesting to ask the following question: given a sample size n , how large should we choose N ? Theorem 3 shows that the test error decreases until $N \approx n$, and then stays roughly the same as long as $N \geq n^{1+\delta}$ (for some $\delta > 0$, although this specific condition is mainly due to our proof technique) and achieves the same test error as the limiting kernel method $N = \infty$ (Theorem 2). Further overparametrization does not improve the risk.

Benign overfitting. The flipside of the last remark is that we can keep increasing N above $n^{1+\delta}$ (hence overparametrize the model by an arbitrary amount), without deteriorating the generalization error.

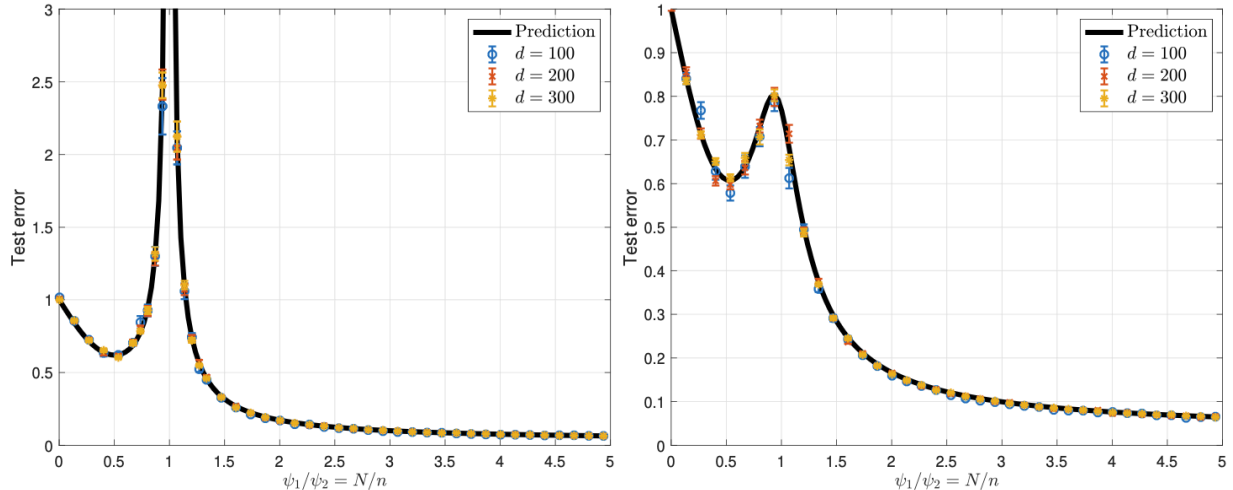


Figure 4.1: Double descent in the test error of random feature ridge regression with ReLu activation ($\sigma = \max\{0, x\}$). The data are generated using $y_i = \langle \beta, \mathbf{x}_i \rangle$ with $\|\beta\|_2 = 1$ and $\psi_2 = n/d = 3$. The regularization parameter is chosen to be $\lambda = 10^{-8}$ (left frame) and $\lambda = 10^{-3}$ (right frame). The continuous black lines correspond to theoretical predictions. The colored symbols are numerical simulations averaged over 20 instances and for several dimensions d . (Figure from [MM19].)

Optimality of interpolation. In the overparametrized regime, the minimum-norm interpolating solution $\lambda = 0+$ achieves the best test error. Hence, similarly to KRR as discussed in Chapter 3, the random feature model displays the *benign overfitting* property. The mechanism for RF is very similar to the KRR case (Remark 3.4.1): the high-degree non-linear part of the activation function σ acts as noise in the features and behaves as an effective *self-induced regularization* (see next section).

4.3 Double descent and proportional asymptotics

In Theorem 3, we assumed that the values of n and N are sufficiently well separated. Indeed, when N is roughly equal to n , the design matrix Φ becomes nearly square and its condition number becomes large. This leads to a peak in the test error at the interpolation threshold¹ $N = n$. This has been called the ‘double descent’ phenomenon, and we already encountered it in Section 2.3.

In the present context, the interpretation of this phenomenon is even clearer. As the number of parameters increases, the test error first follows the classical U-shaped curve, with error initially decreasing and then increasing as it approaches the interpolation threshold. This is in line with the classical picture of bias-variance tradeoff, where increasing the model complexity reduces model misspecification (i.e. the model is too simple to fit the data) but increases sensitivity to statistical fluctuations (i.e. the model may overfit to noise in the training data). However, unlike the traditional U-shaped curve, the test error descends again beyond the interpolation threshold, and the optimal test error is achieved when the number of parameters is significantly larger than the number of samples.

¹The name ‘interpolation threshold’ comes from the fact that as long as $N \geq n$, the RF model has enough parameters to interpolate the training data. If $\lambda = 0+$, then the training error becomes 0 at $N = n$.

The double descent phenomenon in the random feature model has been studied in the proportional asymptotic regime (when $N, n, d \rightarrow \infty$ with $N/d \rightarrow \psi_1$ and $n/d \rightarrow \psi_2$) in [MM19]. Figure 4.1 reports the test error of random feature ridge regression obtained numerically for a fixed value of $\psi_2 = 3$ and two different regularization parameters $\lambda = 10^{-8}$ (left) and $\lambda = 10^{-3}$ (right).

Continuous lines correspond to the theoretical predictions from [MM19], that take the form sketched below.

Theorem 4. *Let the activation function σ satisfy the genericity and growth assumptions stated above. Define*

$$\bar{b}_0 := \mathbb{E}\{\sigma(G)\}, \quad \bar{b}_1 := \mathbb{E}\{G\sigma(G)\}, \quad \bar{b}_*^2 := \mathbb{E}\{\sigma(G)^2\} - \bar{b}_0^2 - \bar{b}_1^2, \quad (4.3.1)$$

where expectation is with respect to $G \sim \mathbf{N}(0, 1)$. Assuming $0 < \bar{b}_0^2, \bar{b}_1^2, \bar{b}_*^2 < \infty$, define ζ by

$$\zeta := \frac{\bar{b}_1}{\bar{b}_*}. \quad (4.3.2)$$

and consider proportional asymptotics $N/d \rightarrow \psi_1$, $n/d \rightarrow \psi_2$. Finally, let the regression function $\{f_d\}_{d \geq 1}$ be such that $f_d(\mathbf{x}) = \beta_{d,0} + \langle \boldsymbol{\beta}_{d,1}, \mathbf{x} \rangle + f_d^{\text{NL}}(\mathbf{x})$, where $f_d^{\text{NL}}(\mathbf{x})$ is a centered rotationally invariant Gaussian process indexed by $\mathbf{x} \in \mathbb{S}^{d-1}(\sqrt{d})$, satisfying $\mathbb{E}_{\mathbf{x}}\{f_d^{\text{NL}}(\mathbf{x})\mathbf{x}\} = \mathbf{0}$. Assume $\mathbb{E}\{f_d^{\text{NL}}(\mathbf{x})^2\} \rightarrow F_*^2$, $\|\boldsymbol{\beta}_{d,1}\|_2^2 \rightarrow F_1^2$.

Then for any value of the regularization parameter $\lambda > 0$, the asymptotic prediction error of random feature ridge regression satisfies

$$\mathbb{E}_{\mathbf{X}, \mathbf{W}, \varepsilon, f_d^{\text{NL}}} \left| R_{\text{RF}}(f_d; \mathbf{X}, \mathbf{W}, \lambda) - \left[F_1^2 \mathcal{B}(\zeta, \psi_1, \psi_2, \lambda/\bar{b}_*^2) + (\tau^2 + F_*^2) \mathcal{V}(\zeta, \psi_1, \psi_2, \lambda/\bar{b}_*^2) + F_*^2 \right] \right| = o_d(1),$$

where $\mathbb{E}_{\mathbf{X}, \mathbf{W}, \varepsilon, f_d^{\text{NL}}}$ denotes expectation with respect to data covariates \mathbf{X} , first layer weight vectors \mathbf{W} , data noise ε , and f_d^{NL} the nonlinear part of the true regression function (as a Gaussian process). The functions \mathcal{B}, \mathcal{V} are explicitly given.

Overall, the behavior of random feature ridge regression (RFRR) in high dimension can be summarized as follows: as n increases while N is kept fixed, the test error of RFRR initially concentrates on the statistical error (test error of kernel ridge regression, with $N = \infty$) for $n \ll N$, then peaks at the interpolation threshold $n = N$ (the double descent phenomenon), and finally saturates on the approximation error (test error with $n = \infty$) for $n \gg N$.

4.4 Polynomial asymptotics: Proof sketch

In this section we outline the proof of Theorem 3.

Recall that the random feature ridge regression solution is given by

$$\hat{f}_{\text{RF}}(\mathbf{x}; \hat{\mathbf{a}}(\lambda)) = \phi_{\text{RF}}(\mathbf{x})^\top \boldsymbol{\Phi}^\top (\lambda \mathbf{I}_n + \boldsymbol{\Phi} \boldsymbol{\Phi}^\top)^{-1} \mathbf{y}. \quad (4.4.1)$$

To prove Theorem 3, we need to control the behavior of the feature matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n \times N}$:

$$\boldsymbol{\Phi} = \begin{bmatrix} \sigma(\langle \mathbf{x}_1, \mathbf{w}_1 \rangle) & \sigma(\langle \mathbf{x}_1, \mathbf{w}_2 \rangle) & \cdots & \sigma(\langle \mathbf{x}_1, \mathbf{w}_N \rangle) \\ \sigma(\langle \mathbf{x}_2, \mathbf{w}_1 \rangle) & \sigma(\langle \mathbf{x}_2, \mathbf{w}_2 \rangle) & \cdots & \sigma(\langle \mathbf{x}_2, \mathbf{w}_N \rangle) \\ \vdots & \vdots & & \vdots \\ \sigma(\langle \mathbf{x}_n, \mathbf{w}_1 \rangle) & \sigma(\langle \mathbf{x}_n, \mathbf{w}_2 \rangle) & \cdots & \sigma(\langle \mathbf{x}_n, \mathbf{w}_N \rangle) \end{bmatrix}. \quad (4.4.2)$$

Similarly to the kernel matrix in Section 3.4, we show that in the polynomial regime Φ can be approximately decomposed into a low-rank matrix (coming from the low-degree polynomials up to degree $\ell = \min(\ell_1, \ell_2)$) plus a matrix that is proportional to an orthogonal matrix, whose span is approximately orthogonal to the low rank part. This decomposition again rely on an operator diagonalization.

Consider the activation function σ , seen as an integral operator from $L^2(\mathbb{S}^{d-1}(\sqrt{d}))$ to $L^2(\mathbb{S}^{d-1}(\sqrt{d}))$ (for convenience, we rescale the argument to have $\mathbf{w} \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$):

$$g(\mathbf{w}) \rightarrow \int \sigma(\langle \mathbf{x}, \mathbf{w} \rangle / \sqrt{d}) g(\mathbf{w}) \tau_d(d\mathbf{w}),$$

where τ_d denotes the uniform measure on $\mathbb{S}^{d-1}(\sqrt{d})$. From assumption (i), we have $\sigma(\langle \cdot, \cdot \rangle / \sqrt{d}) \in L^2(\mathbb{S}^{d-1}(\sqrt{d})^{\otimes 2})$ and from the spectral theorem for compact operators, the function σ can be diagonalized as

$$\sigma(\langle \mathbf{x}, \mathbf{w} \rangle / \sqrt{d}) = \sum_{j=1}^{\infty} \lambda_j \psi_j(\mathbf{x}) \phi_j(\mathbf{w}), \quad (4.4.3)$$

where $(\psi_j)_{j \geq 1}$ and $(\phi_j)_{j \geq 1}$ are two orthonormal bases of $L^2(\mathbb{S}^{d-1}(\sqrt{d}))$, and the λ_j are the eigenvalues with nonincreasing absolute values $|\lambda_1| \geq |\lambda_2| \geq \dots \geq 0$. By rotational invariance of the operator (see Section 3.3), σ is diagonalizable with respect to the spherical harmonics orthogonal basis

$$\sigma(\langle \mathbf{x}, \mathbf{w} \rangle / \sqrt{d}) = \sum_{k=0}^{\infty} \xi_{d,k} \sum_{s=1}^{B(d,k)} Y_{ks}(\mathbf{x}) Y_{ks}(\mathbf{w}). \quad (4.4.4)$$

Assumption (ii) on the activation σ implies that $|\xi_{d,k}| \asymp d^{-k/2}$.

Using the notations in Eq. (4.4.3), let $\boldsymbol{\psi}_k = (\psi_k(\mathbf{x}_1), \dots, \psi_k(\mathbf{x}_n))^{\top}$ be the evaluation of the k -th left eigenfunction at the n data points, and let $\boldsymbol{\phi}_k = (\phi_k(\mathbf{w}_1), \dots, \phi_k(\mathbf{w}_N))^{\top}$ be the evaluation of the k -th right eigenfunction at the N random first-layer weights. Further, let $k(\ell) := \sum_{k \leq \ell} B(d, k)$. Similarly to Section 3.4, we expand Φ into a low-frequency and a high-frequency component:

$$\Phi = \sum_{j=1}^{\infty} \lambda_j \boldsymbol{\psi}_j \boldsymbol{\phi}_j^{\top} =: \Phi_{\leq \ell} + \Phi_{> \ell}, \quad (4.4.5)$$

$$\Phi_{\leq \ell} = \sum_{j=1}^{k(\ell)} \lambda_j \boldsymbol{\psi}_j \boldsymbol{\phi}_j^{\top} =: \boldsymbol{\psi}_{\leq \ell} \boldsymbol{\Lambda}_{\leq \ell} \boldsymbol{\phi}_{\leq \ell}^{\top}, \quad \Phi_{> \ell} = \sum_{j=k(\ell)+1}^{\infty} \lambda_j \boldsymbol{\psi}_j \boldsymbol{\phi}_j^{\top}, \quad (4.4.6)$$

where $\boldsymbol{\Lambda}_{\leq \ell} = \text{diag}(\lambda_1, \dots, \lambda_{k(\ell)})$, $\boldsymbol{\psi}_{\leq \ell} \in \mathbb{R}^{n \times k(\ell)}$ is the matrix whose j -th column is $\boldsymbol{\psi}_j$, and $\boldsymbol{\phi}_{\leq \ell} \in \mathbb{R}^{N \times k(\ell)}$ is the matrix whose j -th column is $\boldsymbol{\phi}_j$. Recalling $|\xi_{d,k}| \asymp d^{-k/2}$, for d sufficiently large, $\boldsymbol{\Lambda}_{\leq \ell}$ contains exactly all the singular values associated to the spherical harmonics of degree at most ℓ .

Notice that the matrix Φ has the same distribution under the mapping $\Phi \leftrightarrow \Phi^{\top}$, $n \leftrightarrow N$, $\mathbf{x}_i \leftrightarrow \mathbf{w}_j$. Hence, without loss of generality let us consider the overparametrized case $N \geq n^{1+\delta}$ with $d^{\ell+\delta} \leq n \leq d^{\ell+1-\delta}$. Then we have the following properties:

Low-frequency part: $\Phi_{\leq \ell}/\sqrt{N}$ has rank $k(\ell) = \Theta_d(d^\ell)$ much smaller than $\min(n, N) = n \geq d^{\ell+\delta}$. Furthermore,

$$\|\psi_{\leq \ell}^\top \psi_{\leq \ell}/n - \mathbf{I}_{k(\ell)}\|_{\text{op}} = o_{d, \mathbb{P}}(1), \quad \|\phi_{\leq \ell}^\top \phi_{\leq \ell}/N - \mathbf{I}_{k(\ell)}\|_{\text{op}} = o_{d, \mathbb{P}}(1), \quad (4.4.7)$$

and $\sigma_{\min}(\Phi_{\leq \ell}/\sqrt{N}) = \Omega_{d, \mathbb{P}}(\min_{k \leq \ell} n |\xi_{d, k}|) = \Omega_{d, \mathbb{P}}(d^\delta)$. Hence, $\Phi_{\leq \ell}/\sqrt{N}$ corresponds to a low-rank spiked matrix along the low-frequency components (degree- ℓ polynomials).

High-frequency part: The singular values of $\Phi_{> \ell}/\sqrt{N}$ concentrates, i.e., $\|\Phi_{> \ell} \Phi_{> \ell}^\top / N - \gamma \mathbf{I}_n\|_{\text{op}} = o_{d, \mathbb{P}}(1)$ where

$$\gamma = \sum_{k \geq \ell+1} \xi_{d, k}^2 B(d, k). \quad (4.4.8)$$

Furthermore, $\|\Phi_{> \ell} \phi_{\leq \ell}^\top / N\|_{\text{op}} = o_{d, \mathbb{P}}(\gamma^{1/2})$. Hence the high-frequency component $\Phi_{> \ell}$ behaves similarly to a random noise matrix, independent of $\phi_{\leq \ell}$.

Based on the above description of the structure of the feature matrix, ridge regression with respect to the random features $\sigma(\langle \mathbf{w}_j, \cdot \rangle)$ is essentially equivalent to doing a regression against a polynomial kernel of degree ℓ , with ℓ depending only on the smaller of the sample size and the network size. The same mechanism as in kernel regression appears: the high-frequency part of the activation σ effectively behaves as noise in the features and can be replaced by an effective ridge regularization γ (another example of *self-induced regularization*).

Remark 4.4.1. Working with the uniform measure over the sphere is particularly convenient because one can exploit known properties of spherical harmonics. However, the analysis in the polynomial regime can be extended to other distributions of the covariate vectors \mathbf{x}_i and of the weights \mathbf{w}_j under appropriate abstract conditions [MMM22].

Chapter 5

Neural tangent features

In this section we consider the neural tangent model associated to a finite-width fully connected two-layer neural network. This model was already introduced in Section 1.4 and captures the behavior of two-layer networks initialized at random, and trained in the linear/lazy regime. As throughout these notes, we focus on square loss.

As discussed in Chapter 3, the infinite-width limit of linearized neural networks corresponds to kernel ridge regression (KRR) with respect to a certain inner product kernel. Here we want to address the following key question:

How large the width of a two-layer network should be for the test error to be well approximated by its infinite-width limit?

It turns out that the answer to this question is a natural generalization of the one we obtained for the random feature model in the last chapter. Roughly speaking, if the number of parameters is large compared to the sample size (hence the model is overparametrized), then the test error is well approximated by the infinite width test error. Notice that the total number of parameters in this case is Nd , and therefore this condition translates into

$$Nd \gg n, \tag{5.0.1}$$

whereby for the random feature model, the overparametrization condition amounted to $N \gg d$.

5.1 Finite-width neural tangent model

Recall that the neural tangent model is defined by

$$f_{\text{NT}}(\mathbf{x}; \mathbf{b}) = \frac{1}{\sqrt{Nd}} \sum_{i=1}^N \langle \mathbf{b}_i, \mathbf{x} \rangle \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle). \tag{5.1.1}$$

Here $\mathbf{b} = (\mathbf{b}_1, \dots, \mathbf{b}_N) \in \mathbb{R}^{Nd}$ is a vector of parameters that are learnt from data, while $(\mathbf{w}_i)_{i \leq N}$ are i.i.d. first-layer weights at initialization, with common distribution ν . In our analysis, we will take $\nu = \text{Unif}(\mathbb{S}^{d-1})$ to be the uniform measure over the sphere. This model can be equivalently described in terms of the NT featurization map

$$\phi_{\text{NT}}(\mathbf{x}) = \frac{1}{\sqrt{Nd}} [\sigma'(\langle \mathbf{w}_1, \mathbf{x} \rangle) \mathbf{x}^\top; \dots; \sigma'(\langle \mathbf{w}_N, \mathbf{x} \rangle) \mathbf{x}^\top]. \tag{5.1.2}$$

We then have $f_{\text{NT}}(\mathbf{x}; \mathbf{b}) = \langle \mathbf{b}, \phi_{\text{NT}}(\mathbf{x}) \rangle$.

We are given i.i.d. samples $\{(y_i, \mathbf{x}_i)\}_{i \leq n}$ and are interested in the ridge regression solution

$$\hat{\mathbf{b}}(\lambda) = \operatorname{argmin}_{\mathbf{b} \in \mathbb{R}^{Nd}} \left\{ \|\mathbf{y} - \Phi \mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_2^2 \right\}, \quad (5.1.3)$$

where we introduced the design matrix $\Phi \in \mathbb{R}^{n \times Nd}$ by

$$\Phi := \begin{bmatrix} -\phi_{\text{NT}}(\mathbf{x}_1) \\ -\phi_{\text{NT}}(\mathbf{x}_2) \\ \vdots \\ -\phi_{\text{NT}}(\mathbf{x}_n) \end{bmatrix}. \quad (5.1.4)$$

We assume the same model as in the previous chapter, namely

$$y_i = f_*(\mathbf{x}_i) + \varepsilon_i, \quad \mathbf{x}_i \sim \operatorname{Unif}(\mathbb{S}^{d-1}(\sqrt{d})), \quad (5.1.5)$$

where ε_i is noise independent of \mathbf{x}_i , with $\mathbb{E}\{\varepsilon_i\} = 0$, $\mathbb{E}\{\varepsilon_i^2\} = \tau^2$. The corresponding test error is

$$R_{\text{NT}}(f_*, \mathbf{X}, \mathbf{y}, \mathbf{W}, \lambda) := \mathbb{E}_{\mathbf{x}} \left\{ (f_*(\mathbf{x}) - \hat{f}_{\text{NT}}(\mathbf{x}; \hat{\mathbf{b}}(\lambda)))^2 \right\}. \quad (5.1.6)$$

Remark 5.1.1 (RF+NT). Recall (cf. Section 1.4) that the full model obtained by linearizing two-layer neural networks with respect to both first layer and second layer weights takes the form

$$f_{\text{in}}(\mathbf{x}; \mathbf{a}, \mathbf{b}) = f_0(\mathbf{x}) + f_{\text{RF}}(\mathbf{x}; \mathbf{a}) + f_{\text{NT}}(\mathbf{x}; \mathbf{b}), \quad (5.1.7)$$

where $f_0(\mathbf{x})$ is the neural network at initialization.

We introduced two simplifications here. First, we dropped the initialization term $f_0(\mathbf{x})$. As already mentioned in Section 1.4, we can indeed endorse $f_0(\mathbf{x}) = 0$ at the price of doubling the number of neurons and choosing the initialization $(\mathbf{w}_j)_{j \leq 2N}$ so that $\mathbf{w}_{N+j} = \mathbf{w}_j$ and $a_{N+j} = -a_j$. The analysis of this initialization is the same as the one pursued here.

Second, we drop the term $f_{\text{RF}}(\mathbf{x}; \mathbf{a})$. This can be shown to have a negligible effect in high dimension (for a generic activation function). Informally, it amounts to reducing the number of parameters from $Nd + N$ to Nd , which is a negligible change.

Remark 5.1.2 (Complexity at prediction time). So far, our presentation mirrored the one of the RF model in the last chapter. However, there is an important practical difference in the complexity of evaluating the functions $f_{\text{RF}}(\cdot; \mathbf{a})$ and $f_{\text{NT}}(\cdot; \mathbf{b})$ which is the task we need to perform at prediction (a.k.a. inference) time.

Consider first the RF model. Organizing the weights \mathbf{w}_i as rows of a matrix $\mathbf{W} \in \mathbb{R}^{N \times d}$, the function $f_{\text{RF}}(\cdot; \mathbf{a})$ can be evaluated at a point \mathbf{x} using $O(Nd)$ operations. First, we compute $\mathbf{z} = \mathbf{W}\mathbf{x}$ (which can be done with $O(Nd)$ operations), and then we output $\sum_{i=1}^N a_i \sigma(z_i)$ which can be done in $O(N)$ operations (if evaluating the one-dimensional function σ takes $O(1)$ operations).

Next consider the NT model. We arrange the parameters \mathbf{b}_i into $\mathbf{B} \in \mathbb{R}^{N \times d}$. Next we evaluate $\mathbf{u} = \mathbf{B}\mathbf{x}$, $\mathbf{z} = \mathbf{W}\mathbf{x}$ and finally output $\sum_{i=1}^N u_i \sigma'(z_i)$. Under similar assumptions, this results in $O(Nd)$ operations.

At first sight, the two models behave similarly. However, as we saw in the last section, the ‘approximation power’ of the RF model is controlled by the number of parameters $p = N$. Similarly, in the NT model we will see that it is controlled by $p = Nd$. Hence the complexity of prediction at constant number of parameters is $O(pd)$ for the RF model compared to $O(p)$ for the NT model. In other words, the NT model is much simpler to evaluate in high dimension.

Remark 5.1.3 (Complexity of SGD training). Consider next training either the RF or the NT model using stochastic gradient descent (SGD) under quadratic loss. Of course, in this case the cost function is quadratic and we can use faster algorithms than SGD. However, these algorithms are not available, or of common use for actual non-linear networks, and we hope to gain some insights into those cases. We assume a batch of size B , which we denote by $\{(y_1, \mathbf{x}_1), \dots, (y_B, \mathbf{x}_B)\}$, and will just compute the complexity of a single SGD step.

For the RF model the gradient takes the form (here $\hat{R}_B(\mathbf{a})$ is the empirical risk with respect to the batch)

$$\nabla_{\mathbf{a}} \hat{R}_B(\mathbf{a}) = -\sigma(\mathbf{W}\mathbf{X}^\top)\mathbf{r}, \quad (5.1.8)$$

where $\mathbf{W} \in \mathbb{R}^{N \times d}$ is the matrix whose i -th row is \mathbf{w}_i , $\mathbf{X} \in \mathbb{R}^{B \times d}$ is the matrix whose j -th row is \mathbf{x}_j , and $\mathbf{r} \in \mathbb{R}^B$, with $r_i = y_i - f_{\text{RF}}(\mathbf{x}_i; \mathbf{a})$. Assuming σ can be evaluated in $O(1)$ time, the above gradient can be evaluated in time $O(NdB) = O(pdB)$.

For the NT model, we view the parameters as a matrix $\mathbf{b} \in \mathbb{R}^{N \times d}$ whose i -th row corresponds to neuron i . We then have

$$\nabla_{\mathbf{b}} \hat{R}_B(\mathbf{b}) = -\sum_{i=1}^B r_i \begin{bmatrix} -\sigma(\mathbf{w}_1^\top \mathbf{x}_i) \cdot \mathbf{x}_i^\top \\ -\sigma(\mathbf{w}_2^\top \mathbf{x}_i) \cdot \mathbf{x}_i^\top \\ \vdots \\ -\sigma(\mathbf{w}_N^\top \mathbf{x}_i) \cdot \mathbf{x}_i^\top \end{bmatrix}. \quad (5.1.9)$$

Again, this can be evaluated in time $O(NdB) = O(pB)$. The complexity is significantly smaller than for RF.

5.2 Approximation by infinite-width KRR: Kernel matrix

The empirical kernel matrix plays a crucial role in ridge regression, and hence it is useful to begin by analyzing its structure. We denote the kernel matrix associated to the finite-width NT model by $\mathbf{K}_N \in \mathbb{R}^{Nd \times Nd}$. By definition, its i, j -th entry is $(\mathbf{K}_N)_{ij} = \langle \phi_{\text{NT}}(\mathbf{x}_i), \phi_{\text{NT}}(\mathbf{x}_j) \rangle$. Explicitly:

$$(\mathbf{K}_N)_{ij} = \frac{1}{Nd} \sum_{k=1}^N \sigma'(\langle \mathbf{w}_k, \mathbf{x}_i \rangle) \sigma'(\langle \mathbf{w}_k, \mathbf{x}_j \rangle) \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \quad (5.2.1)$$

The corresponding infinite-width kernel matrix is given by the expectation of the above

$$(\mathbf{K})_{ij} = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{d} \cdot \mathbb{E}_{\mathbf{w}} \{ \sigma'(\langle \mathbf{w}, \mathbf{x}_i \rangle) \sigma'(\langle \mathbf{w}, \mathbf{x}_j \rangle) \}. \quad (5.2.2)$$

Here and below, we denote by $\mathbb{E}_{\mathbf{w}}$, $\mathbb{P}_{\mathbf{w}}$ expectation and probability with respect to \mathbf{w} at fixed $(\mathbf{x}_i : i \leq n)$.

By the law of large numbers (and under very minimal conditions on σ , e.g. $\mathbb{E}_{\mathbf{w}}[\sigma'(\langle \mathbf{w}, \mathbf{x}_i \rangle)^2] < \infty$), we have $\lim_{N \rightarrow \infty} (\mathbf{K}_N)_{ij} = (\mathbf{K})_{ij}$ for fixed i, j . However, we are interested in N, n, d all diverging simultaneously, and the convergence of the matrix entries does not provide strong information on the overall matrix (e.g. its eigenvalues), which is our main focus.

The activation function will enter our estimates through the following parameters

$$v_\ell(\sigma) := \sum_{k \geq \ell} \frac{1}{k!} \langle \text{He}_k, \sigma^\wedge \rangle_{L^2(\mathbf{N}(0,1))}^2. \quad (5.2.3)$$

Here $\langle f, g \rangle_{L^2(\mathcal{N}(0,1))} = \mathbb{E}\{f(G)g(G)\}$ (where $G \sim \mathcal{N}(0,1)$) is the usual L^2 inner product with respect to the Gaussian measure, and He_k is the k -th Hermite polynomial, with the standard normalization $\langle \text{He}_j, \text{He}_k \rangle_{L^2(\mathcal{N}(0,1))} = k! \delta_{jk}$. We further make the following assumption on σ .

Assumption 1 (Polynomial growth). *We assume that σ is weakly differentiable with weak derivative σ' satisfying $|\sigma'(x)| \leq B(1 + |x|)^B$ for some finite constant $B > 0$, and that $v_\ell(\sigma') > 0$. (The latter is equivalent to σ not being a maximum degree- ℓ polynomial.)*

We next state a kernel concentration result.

Theorem 5 (Kernel concentration). *Let $\gamma = (1 - \varepsilon_0)v_\ell(\sigma)$ for some constant $\varepsilon_0 \in (0, 1)$. Further, let $\beta > 0$ be arbitrary. Then, there exist constants $C', C_0 > 0$ such that the following hold.*

Define the event:

$$\mathcal{A}_\gamma := \{\mathbf{K} \succeq \gamma \mathbf{I}_n, \|\mathbf{X}\|_{\text{op}} \leq 2(\sqrt{n} + \sqrt{d})\}. \quad (5.2.4)$$

For any $\mathbf{X} \in \mathcal{A}_\gamma$, we have

$$\mathbb{P}_{\mathbf{w}} \left(\left\| \mathbf{K}^{-1/2} \mathbf{K}_N \mathbf{K}^{-1/2} - \mathbf{I}_n \right\|_{\text{op}} > \sqrt{\frac{(n+d)(\log(nNd))^{C'}}{Nd}} + \frac{(n+d)(\log(nNd))^{C'}}{Nd} \right) \leq d^{-\beta}. \quad (5.2.5)$$

Further if $n \leq d^{\ell+1}/(\log d)^{C_0}$, then $\mathbb{P}(\mathcal{A}_\gamma) \geq 1 - n^{-\beta}$ for all n large enough.

Remark 5.2.1. Note that the event \mathcal{A}_γ only depends on the matrix of covariates \mathbf{X} . Hence, we view \mathcal{A}_γ as a set in $\mathbb{R}^{n \times d}$ and write $\mathbf{X} \in \mathcal{A}_\gamma$ if \mathbf{X} satisfies the stated conditions. Equation (5.2.5) holds for any such \mathbf{X} , not necessarily random ones (the probability being only with respect to \mathbf{W}).

Since $\mathbb{P}(\mathcal{A}_\gamma) \geq 1 - n^{-\beta}$ under the stated conditions, we also have that, if $n \leq d^{\ell+1}/(\log d)^{C_0}$, then with probability at least $1 - d^{-\beta}$ (both with respect to \mathbf{X} and \mathbf{W})

$$\left\| \mathbf{K}^{-1/2} \mathbf{K}_N \mathbf{K}^{-1/2} - \mathbf{I}_n \right\|_{\text{op}} \leq \sqrt{\frac{(n+d)(\log(nNd))^{C'}}{Nd}} + \frac{(n+d)(\log(nNd))^{C'}}{Nd}. \quad (5.2.6)$$

Remark 5.2.2. A first attempt at proving concentration would be to try to control $\|\mathbf{K}_N - \mathbf{K}\|_{\text{op}}$. However, this results in suboptimal bounds because \mathbf{K} has eigenvalues on several well separated scales. Indeed, we saw in Chapter 3 that \mathbf{K} has one eigenvalue of order n , d eigenvalues of order n/d , approximately $d^k/k!$ eigenvalues of order n/d^k for any $k \leq \ell$, and $n - O(d^\ell)$ eigenvalues of order 1.

In Theorem 5 we bound $\|\mathbf{K}^{-1/2} \mathbf{K}_N \mathbf{K}^{-1/2} - \mathbf{I}_n\|_{\text{op}}$, hence effectively looking at each group of eigenvalues at different scales.

Assume for simplicity $n \geq c_1 d$ for some constant c_1 . There is little loss of generality in doing so. Indeed, even if we know that the target function is linear $f_*(\mathbf{x}) = \langle \boldsymbol{\beta}_*, \mathbf{x} \rangle$, it is information theoretically impossible to achieve prediction error less than (say) half the error of the trivial predictor $\hat{f}(\mathbf{x}) = 0$, unless $n \geq c_1 d$ for some constant c_1 .

In this setting the norm bound in Theorem 5 (equivalently, the right-hand side of Eq. (5.2.6)) is small as soon as

$$\frac{Nd}{(\log Nd)^C} \gg n. \quad (5.2.7)$$

Modulo the log factors, this is the sharp overparametrization condition $Nd \gg n$. This is also our first piece of evidence that the relevant control parameter is the ratio of number of parameters to sample size, rather than number of neurons to sample size.

5.3 Approximation by infinite-width KRR: Test error

By itself, controlling the kernel matrix as we did in Theorem 5 does not allow to bound the test error, which involves evaluating the regression function at a fresh test point. Explicitly, we have

$$\hat{f}(\mathbf{x}; \lambda) = K_N(\mathbf{x}, \mathbf{X})(\mathbf{K}_N + \lambda \mathbf{I}_N)^{-1} \mathbf{y} \quad (5.3.1)$$

$$= K_N(\mathbf{x}, \cdot)^\top (\mathbf{K}_N + \lambda \mathbf{I}_N)^{-1} \mathbf{f}_* + K_N(\mathbf{x}, \mathbf{X})(\mathbf{K}_N + \lambda \mathbf{I}_N)^{-1} \boldsymbol{\varepsilon}, \quad (5.3.2)$$

where we used the notations $K_N(\mathbf{x}, \cdot) := (K_N(\mathbf{x}, \mathbf{x}_i) : i \leq n) \in \mathbb{R}^n$ and $\mathbf{f}_* := (f_*(\mathbf{x}_i) : i \leq n) \in \mathbb{R}^n$. Consider, to be concrete, the bias contribution to the test error. We get

$$\begin{aligned} \mathbb{B}(\lambda) &= \mathbb{E}_{\mathbf{x}} \{ [f_*(\mathbf{x}) - K_N(\mathbf{x}, \cdot)^\top (\mathbf{K}_N + \lambda \mathbf{I}_N)^{-1} \mathbf{f}_*]^2 \} \\ &= \mathbb{E}_{\mathbf{x}} \{ f_*(\mathbf{x})^2 \} - 2 \mathbb{E}_{\mathbf{x}} \{ f_*(\mathbf{x}) K_N(\mathbf{x}, \cdot)^\top (\mathbf{K}_N + \lambda \mathbf{I}_N)^{-1} \mathbf{f}_* \} \\ &\quad + \mathbf{f}_*^\top (\mathbf{K}_N + \lambda \mathbf{I}_N)^{-1} \mathbf{K}_N^{(2)} (\mathbf{K}_N + \lambda \mathbf{I}_N)^{-1} \mathbf{f}_*, \end{aligned}$$

where $\mathbf{K}_N^{(2)} \in \mathbb{R}^{n \times n}$ is the matrix with entries $(\mathbf{K}_N^{(2)})_{ij} = \mathbb{E}_{\mathbf{x}} \{ K_N(\mathbf{x}_i, \mathbf{x}) K_N(\mathbf{x}, \mathbf{x}_j) \}$.

Clearly the bound on the kernel matrix afforded by Theorem 5 is not sufficient to control this expression. Indeed, proving the result below is significantly more challenging.

Theorem 6. *Let $B, c_0 > 0$, $\ell \in \mathbb{N}$ be fixed. Then, there exist constants $C_0, C, C' > 0$ such that the following holds.*

If σ satisfies Assumption 1, and further

$$c_0 d \leq n \leq \frac{d^2}{(\log d)^{C_0}}, \quad \text{if } \ell = 1, \quad (5.3.3)$$

$$d^\ell (\log d)^{C_0} \leq n \leq \frac{d^{\ell+1}}{(\log d)^{C_0}}, \quad \text{if } \ell > 1. \quad (5.3.4)$$

If $Nd/(\log(Nd))^C \geq n$, then for any $\lambda \geq 0$,

$$R_{\text{NT}}(f_*; \lambda) = R_{\text{KRR}}(f_*; \lambda) + O_{d, \mathbb{P}} \left(\tau_+^2 \sqrt{\frac{n(\log(Nd))^{C'}}{Nd}} \right)$$

where $\tau_+^2 := \|f_*\|_{L^2}^2 + \tau^2$.

As anticipated, this theorem establishes that the infinite width kernel is a good approximation to the finite width kernel under the overparametrization condition (5.2.7) which is only a polylogarithmic factor above the optimal condition $Nd \gg n$.

Roughly speaking, the condition of Eqs. (5.3.3) and (5.3.4) rules out cases in which n is comparable to d^ℓ for an integer ℓ . As we saw in Chapter 3, when $n \asymp d^\ell$, the structure of the infinite width kernel matrix \mathbf{K} is more complex, and the KRR with respect to inner product kernels ‘partially’ fits the degree ℓ component of the target function. We expect this condition to be a proof artifact.

Figures 5.1 and 5.2 (from [MZ22]) illustrate the content of Theorem 6 via numerical simulations. In these simulations we fix $d = 20$, $\tau = 0.5$ and consider the target function

$$f_*(\mathbf{x}) = \sqrt{\frac{4}{10}} \text{He}_1(\langle \boldsymbol{\beta}_*, \mathbf{x} \rangle) + \sqrt{\frac{2}{10}} \text{He}_2(\langle \boldsymbol{\beta}_*, \mathbf{x} \rangle) + \sqrt{\frac{1}{120}} \text{He}_4(\langle \boldsymbol{\beta}_*, \mathbf{x} \rangle). \quad (5.3.5)$$

We can clearly observe the main phenomena captured by Theorem 6. The train error vanishes for $Nd \geq n$, while the test error converges to a limit for large Nd/n . This limit is accurately reproduced by (infinite-width) KRR.

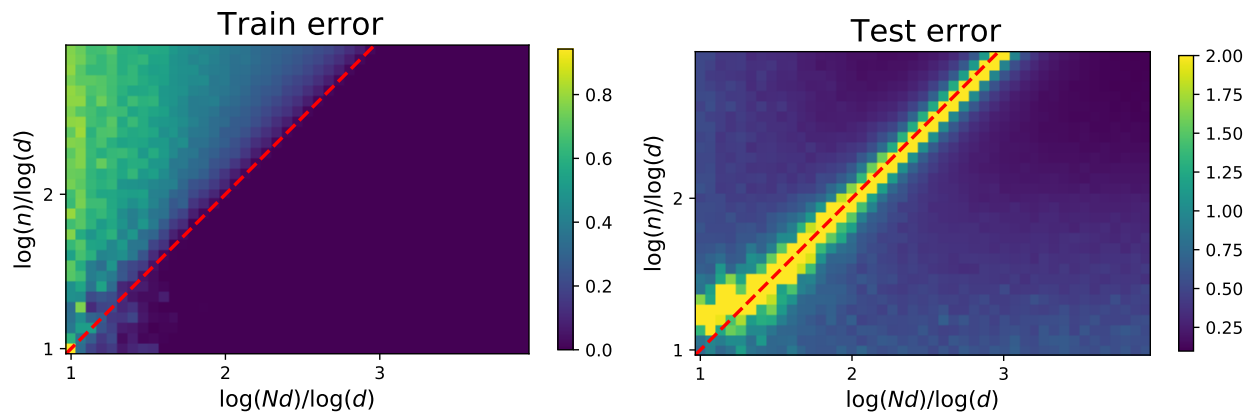


Figure 5.1: Neural Tangent min-norm regression ($\lambda = 0+$) in $d = 20$ dimensions (see text for details). Left: train error as a function of the sample size n and number of parameters Nd . Right: test error. Results are averaged over 10 realizations. (From [MZ22].)

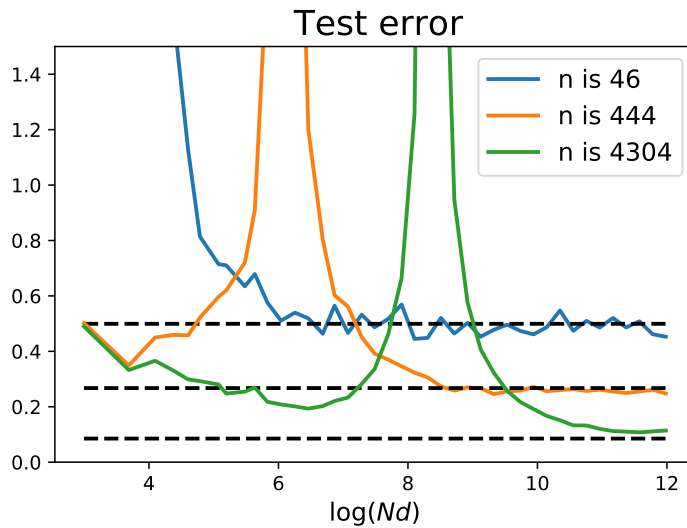


Figure 5.2: Test error Neural Tangent min-norm regression ($\lambda = 0+$) in $d = 20$ dimensions (see text for details). Color curves refer to $n = 46, 444$ and 4304 , and dashed lines report the corresponding KRR limit. (From [MZ22].)

Remark 5.3.1. As we saw in Chapter 3, under the conditions of Eqs. (5.3.3), (5.3.4), KRR with respect to an inner product kernel approximates the projection of the target function onto degree ℓ polynomials. Putting this result together with Theorem 6 (under the conditions in the theorem's statement) yields

$$R_{\text{NT}}(f_*; \lambda) = R_{\text{PRR}}(f_*; \lambda + v_\ell(\sigma)) + O_{d, \mathbb{P}}\left(\tau_+^2 \sqrt{\frac{n(\log(Nd))^{C'}}{Nd}} + \tau_+^2 \sqrt{\frac{n(\log n)^C}{d^{\ell+1}}}\right). \quad (5.3.6)$$

Here $R_{\text{PRR}}(f_*; \lambda')$ is the risk of polynomial ridge regression (PRR), i.e. ridge regression with respect to polynomials of maximum degree ℓ , at regularization λ' . Note that Eq. (5.3.6) is somewhat more quantitative than the statement Chapter 3. Namely, the difference between KRR and PRR is upper bounded by $(n/d^{\ell+1})^{1/2}$, up to a polylogarithmic factor.

We refer to [MZ22] for a more complete discussion, but point out that Eq. (5.3.6) displays one more example of self-induced regularization. Namely, the ridge regularization parameter in PRR is equal $\lambda + v_\ell(\sigma)$. The original parameter λ gets inflated by an additive term $v_\ell(\sigma)$ that comes from the high frequency part of the activations.

5.4 Proof sketch: Kernel concentration

The proof of Theorem 5 is simple and instructive, and hence we present a brief sketch. We will work under the simplifying assumption that $\sup_t |\sigma'(t)| \leq M < \infty$, and $n \geq d$.

The proof uses the following classical matrix concentration inequality [Ver18][Thm. 5.4.1]. (Below, for a random variable U , $\text{esssup}(U) := \inf\{t \in \mathbb{R} : \mathbb{P}(U \leq t) = 1\}$.)

Theorem 7 (Matrix Bernstein inequality). *Let $\{\mathbf{Z}_k\}_{k \leq N}$ be a sequence of symmetric and independent random matrices $\mathbf{Z}_k \in \mathbb{R}^{n \times n}$. Define*

$$v := \left\| \sum_{k=1}^N \mathbb{E}[\mathbf{Z}_k^2] \right\|_{\text{op}}, \quad L := \max_{k \leq N} \text{esssup} \|\mathbf{Z}_k\|_{\text{op}}. \quad (5.4.1)$$

Then, for any $t \geq 0$.

$$\mathbb{P}\left(\left\| \frac{1}{N} \sum_{k=1}^N \mathbf{Z}_k \right\|_{\text{op}} \geq t\right) \leq 2n \cdot \exp\left(-\frac{N^2 t^2 / 2}{v + NLt/3}\right), \quad (5.4.2)$$

In other words, matrix Bernstein states that, with probability at least $1 - n^{-\beta}$

$$\left\| \frac{1}{N} \sum_{k=1}^N \mathbf{Z}_k \right\|_{\text{op}} \leq \frac{C(\beta)}{N} \max\left(\sqrt{v \log n}, L \log n\right). \quad (5.4.3)$$

We define

$$\mathbf{D}_k := \text{diag}(\sigma'(\langle \mathbf{w}_k, \mathbf{x}_1 \rangle), \dots, \sigma'(\langle \mathbf{w}_k, \mathbf{x}_n \rangle)). \quad (5.4.4)$$

Letting $\mathbf{S}_N := \mathbf{K}^{-1/2} \mathbf{K}_N \mathbf{K}^{-1/2} - \mathbf{I}$ denote the matrix that we want to bound, we have

$$\mathbf{S}_N = \frac{1}{N} \sum_{k=1}^N \mathbf{Z}_k, \quad (5.4.5)$$

$$\mathbf{Z}_k := \frac{1}{d} \mathbf{K}^{-1/2} \mathbf{D}_k \mathbf{X} \mathbf{X}^\top \mathbf{D}_k \mathbf{K}^{-1/2} - \mathbf{I}. \quad (5.4.6)$$

We begin by bounding the coefficient L in Bernstein inequality. By triangle inequality

$$\|\mathbf{Z}_k\|_{\text{op}} \leq \frac{1}{d} \lambda_{\min}(\mathbf{K})^{-1} M^2 \|\mathbf{X}\mathbf{X}^\top\|_{\text{op}} + 1. \quad (5.4.7)$$

Therefore, on the event \mathcal{A}_γ

$$\begin{aligned} L &\leq \frac{M^2}{\gamma d} 4(\sqrt{n} + \sqrt{d})^2 + 1 \\ &\leq C \frac{n}{d}, \end{aligned}$$

where the constant C depends on M, γ .

Next consider the coefficient v . We have of course $\mathbb{E}[\mathbf{Z}_k^2] \succeq \mathbf{0}$. On the other hand,

$$\begin{aligned} \mathbb{E}[\mathbf{Z}_k^2] &= \frac{1}{d^2} \mathbb{E}\{\mathbf{K}^{-1/2} \mathbf{D}_k \mathbf{X} \mathbf{X}^\top \mathbf{D}_k \mathbf{K}^{-1} \mathbf{D}_k \mathbf{X} \mathbf{X}^\top \mathbf{D}_k \mathbf{K}^{-1/2}\} - \mathbf{I} \\ &\preceq \frac{M^2}{\gamma d^2} \mathbb{E}\{\mathbf{K}^{-1/2} \mathbf{D}_k \mathbf{X} \mathbf{X}^\top \mathbf{X} \mathbf{X}^\top \mathbf{D}_k \mathbf{K}^{-1/2}\} \\ &\preceq \frac{M^2}{\gamma d^2} 4(\sqrt{n} + \sqrt{d})^2 \mathbb{E}\{\mathbf{K}^{-1/2} \mathbf{D}_k \mathbf{X} \mathbf{X}^\top \mathbf{D}_k \mathbf{K}^{-1/2}\} \\ &= \frac{M^2}{\gamma d} 4(\sqrt{n} + \sqrt{d})^2 \mathbf{I}. \end{aligned}$$

Summing over k , we get $v \leq CNn/d$.

Summarizing, there exists a constant C (depending only on M and γ) such that

$$L \leq \frac{Cn}{d}, \quad v \leq \frac{CNn}{d}.$$

Applying Bernstein inequality (5.4.3), we get

$$\begin{aligned} \left\| \mathbf{K}^{-1/2} \mathbf{K}_N \mathbf{K}^{-1/2} - \mathbf{I} \right\|_{\text{op}} &\leq \frac{C}{N} \max \left(\sqrt{\frac{Nn}{d} \log n}, \frac{n}{d} \log n \right) \\ &\leq C \left(\sqrt{\frac{n}{Nd} \log n} + \frac{n}{Nd} \log n \right). \end{aligned}$$

This proves Theorem 5 in the case of bounded σ' . The general case requires to handle large values of $\sigma'(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)$ via a truncation argument. This results in the extra logarithmic factors, which are likely to be a proof artifact.

Chapter 6

Why stop being lazy (and how)

These lecture notes were focused so far on learning within a training regime in which the model $f(\mathbf{x}; \boldsymbol{\theta})$ is well approximated by its first order Taylor expansion (in the model parameters) around its initialization. This style of analysis is useful in elucidating some puzzling properties of modern machine learning:

1. Overparametrization helps optimization.
2. Gradient-based algorithms select a specific model among all the ones that minimize the empirical risk.
3. The specific models can overfit the training data (vanishing training error) and yet generalize to unseen data.

The linear regime provides a simple and analytically tractable setting in which these phenomena can be understood in detail.

Finally, there is experimental evidence that, in some settings, the linear theory captures the behavior of real SGD-trained neural networks [LXS⁺19, GSJW20] (see also Appendix B).

On the other hand, it is important to emphasize that linearized neural networks are significantly less powerful than fully trained neural networks. In other words, while neural networks can be trained in a regime in which they are well approximated by their first order Taylor expansion, this is the byproduct of specific choices on the parametrization. Other choices are possible, leading to different and potentially better models.

Following [COB19] we will use the somewhat irreverent term ‘lazy training’ to refer to training schemes that produce networks in the linear regime. In this chapter we will discuss limitations of this approach, and how other training schemes overcome them. In particular, we will clarify that lazy training is not a necessary consequence of the infinite width limit. Other non-lazy infinite-width limits exist, and can outperform lazy ones.

6.1 Lazy training fails on ridge functions

One of the simplest problems on which lazy training is suboptimal is provided by ‘ridge functions’, i.e. functions that depend on a one-dimensional projection of the covariate vector.

As in the previous chapters, assume to be given data $\{(\mathbf{x}_i, y_i)\}_{i \leq n}$, where $\mathbf{x}_i \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$, $y_i = f_*(\mathbf{x}_i) + \varepsilon_i$ and ε_i is noise with $\mathbb{E}\{\varepsilon_i\} = 0$, $\mathbb{E}\{\varepsilon_i^2\} = \tau^2$. A ridge function is a target function

of the form:

$$f_*(\mathbf{x}) = \varphi(\langle \mathbf{w}_*, \mathbf{x} \rangle), \quad (6.1.1)$$

where¹ $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ and $\|\mathbf{w}_*\|_2 = 1$.

We saw in the previous chapters that the learning under isotropic covariates in the linear regime is controlled by the decomposition of f_* into polynomials. Let us compute the mass of f_* in the subspace $\mathcal{V}_{d,k}$ of polynomials of degree k that are orthogonal to polynomials of maximum degree $k - 1$. Using the theory of Section 3.3, we obtain

$$\begin{aligned} \lim_{d \rightarrow \infty} \|\mathbf{P}_k f_*\|_{L^2}^2 &= \lim_{d \rightarrow \infty} B(d, k) \cdot \mathbb{E}_{\mathbf{x}} \{ \varphi(\langle \mathbf{w}_*, \mathbf{x} \rangle) Q_k^{(d)}(\langle \mathbf{w}_*, \mathbf{x} \rangle) \}^2 \\ &= \frac{1}{k!} \mathbb{E} \{ \varphi(G) \text{He}_k(G) \}^2. \end{aligned}$$

In particular²

$$\lim_{d \rightarrow \infty} \|\mathbf{P}_{>\ell} f_*\|_{L^2}^2 = \sum_{k=\ell+1}^{\infty} \frac{1}{k!} \mathbb{E} \{ \varphi(G) \text{He}_k(G) \}^2 =: b_\ell.$$

Unless φ is a polynomial, we have $b_\ell > 0$ strictly for all ℓ . This simple remark has an immediate consequence. We saw in previous chapters that if the sample size is $n \ll d^{\ell+1}$, then linearized neural network only learn the best approximation of the target by degree- ℓ polynomials. Hence, their excess risk is bounded away from zero. This consequence is stated more precisely below.

Corollary 6.1.1. *Fix $\delta > 0$. Let $R_{\mathbf{X}}(f_*)$, $\mathbf{X} \in \{\text{KRR}, \text{RF}, \text{NT}\}$ be (respectively) the excess test error of kernel ridge regression (with inner product kernel), random feature ridge regression, or neural tangent regression. For $\mathbf{X} = \text{RF}$, assume $N \geq n^{1+\delta}$, and for $\mathbf{X} = \text{NT}$, assume $Nd \geq n^{1+\delta}$. Under the data model above (in particular, $y_i = \varphi(\langle \mathbf{w}_*, \mathbf{x}_i \rangle) + \varepsilon_i$), if $n \leq d^{\ell+1-\delta}$ then the following lower bound holds in probability:*

$$\liminf_{n, d \rightarrow \infty} R_{\mathbf{X}}(f_*) \geq b_\ell. \quad (6.1.2)$$

Further unless φ is a polynomial, we have $b_\ell > 0$ strictly for all ℓ .

This is somewhat disappointing. After all, the function (6.1.1) looks extremely simple, and only has d free parameters (plus, eventually $O(1)$ parameters to approximate the one-dimensional function φ). One extreme case is given by $\varphi = \sigma$. Note that Corollary 6.1.1 does not change in this case: linearized neural networks cannot learn a target function that coincides with a single neuron from any polynomial number of samples.

It is natural to wonder whether learning ridge functions might be hard for some hidden reason. It turns out that ridge functions can often be learnt efficiently, even if we limit ourselves to gradient-based methods. For instance, consider using a one-neuron network $f(\mathbf{x}; \mathbf{w}) := \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$ in the case

¹In fact, the term ridge function refers in applied mathematics and statistics to the broader class of targets of the form $f_*(\mathbf{x}) = \psi(\mathbf{U}_*^T \mathbf{x})$ where $\psi : \mathbb{R}^k \rightarrow \mathbb{R}$ and $\mathbf{U}_*^T \mathbf{U}_* = \mathbf{I}_k$ [LS75]. Unfortunately, this terminology can be confusing because we are applying ridge regression to learn ridge functions, and the two uses of “ridge” are not related!

²Inverting follows from dominated convergence if, for instance $|\varphi(t)| \leq C \exp(C|t|)$ for a constant $C > 0$ and all $t \in \mathbb{R}$.

$\varphi = \sigma$. We perform gradient descent with respect to the empirical risk:

$$\hat{R}_n(\mathbf{w}) = \sum_{i=1}^n (y_i - \sigma(\langle \mathbf{w}, \mathbf{x}_i \rangle))^2. \quad (6.1.3)$$

The corresponding excess test error is $R(\mathbf{w}) = \mathbb{E}\{(\sigma(\langle \mathbf{w}_*, \mathbf{x} \rangle) - \sigma(\langle \mathbf{w}, \mathbf{x} \rangle))^2\}$. The following result from [MBM18] implies that gradient descent succeeds in learning this target.

Proposition 6.1.2. *Under the above data distribution (namely, $\mathbf{x}_i \sim \text{Unif}(\mathbb{S}^{d-1}(\sqrt{d}))$) and $y_i = \sigma(\langle \mathbf{w}_*, \mathbf{x}_i \rangle) + \varepsilon_i$) assume σ is bounded and three times differentiable with bounded derivatives. Further assume $\sigma'(t) > 0$ for all t . Then there exists a constant $C = C_{\sigma, \tau}$ such that, if $n \geq C d \log d$, the following happens:*

1. The empirical risk $\hat{R}_n(\mathbf{w})$ has a unique (local, hence global) minimizer $\hat{\mathbf{w}}_n \in \mathbb{R}^d$.
2. Gradient descent converges globally to $\hat{\mathbf{w}}_n$.
3. The excess test error at this minimizer is bounded as

$$R(\hat{\mathbf{w}}_n) \leq C \sqrt{\frac{d \log n}{n}}. \quad (6.1.4)$$

Considering —to be definite— the setting of this proposition, and learning using two layer networks. We find ourselves in the following peculiar situation:

1. A one-neuron network can learn the target from $n = O(d \log d)$ samples, which roughly matches the information theoretic lower bound.
2. As the number of neurons gets large, we learn from [JGH18] that a two-layer neural network is well approximated by the corresponding NT model.
3. The NT model cannot learn the same target from any polynomial (in d) number of samples.

In other words, the learning ability of two-layer networks seems to deteriorate as the number of neurons increases. Needless to say, this is a very counterintuitive behavior.

What is ‘going wrong’?

As we will see in the next section, point 2 above is what is going wrong. Wide neural networks are well approximated by their linear counterpart only under a specific scaling of the network weights. Other behavior (and other infinite-width limits) are obtained if different scalings are chosen. Under these scalings, the obstructions and counterintuitive behaviors above disappear.

6.2 Non-lazy infinite-width networks (a.k.a. mean field)

Let us restart from scratch, and write once more the parametric form of a two-layer neural network, which we first wrote down in Eq. (1.1.7):

$$\hat{f}(\mathbf{x}; \Theta) = \alpha_N \sum_{i=1}^N a_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle), \quad \theta_i = (a_i, \mathbf{w}_i) \in \mathbb{R}^{d+1}. \quad (6.2.1)$$

Here we denoted collectively by $\Theta := ((a_i, \mathbf{w}_i) : i \leq N)$ the network parameters. We mentioned briefly in Chapter 1 that in certain regimes, such a network is well approximated by its first order Taylor expansion around the initialization. Roughly speaking, this is the case when N is large, but only if we scale α_N in a suitable way, i.e. $\alpha_N \asymp N^{-1/2}$.

We want to revisit the infinite-width limit, under a different scaling of the model, namely

$$\hat{f}(\mathbf{x}; \Theta) = \frac{1}{N} \sum_{i=1}^N a_i \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle). \quad (6.2.2)$$

This is sometimes referred to as the ‘mean field’ scaling, after [MMN18], which we will follow here.

It is useful to denote by $\theta_i = (a_i, \mathbf{w}_i) \in \mathbb{R}^{d+1}$ the vector of parameters for neuron i , and define $\sigma_*(\mathbf{x}; (a, \mathbf{w})) := a\sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$. Consider SGD training with respect to square loss. We will fix a stepsize η and index iterations by $t \in \{0, \eta, 2\eta, \dots\}$. The SGD iteration reads:

$$\theta_i^{t+\eta} = \theta_i^t + \eta(y_{I(t)} - \hat{f}(\mathbf{x}_{I(t)}; \Theta^t)) \nabla_{\theta} \sigma_*(\mathbf{x}_{I(t)}; \theta_i^t), \quad (6.2.3)$$

where $I(t)$ is the index of the sample used at step t of SGD. We will have in mind two settings: (i) Online SGD, whereby at each step a fresh sample from the population is used (in this case we are optimizing the population error); (ii) Standard SGD with batch size 1, whereby $I(t) \sim \text{Unif}([n])$.

When the stepsize η becomes small, SGD effectively averages over a large number of samples in a fixed time interval $[t, t + \Delta]$, and it is natural to expect that the above dynamics is well approximated by the following gradient flow

$$\dot{\theta}_i^t = \mathbb{E}_{y, \mathbf{x}} \{ (y - \hat{f}(\mathbf{x}; \Theta^t)) \nabla_{\theta} \sigma_*(\mathbf{x}; \theta_i^t) \}. \quad (6.2.4)$$

Here $\mathbb{E}_{y, \mathbf{x}}$ is the expectation over the population distribution for online SGD, or over the sample for standard SGD. We can conveniently rewrite the above flow by introducing the functions

$$V(\theta) = -\mathbb{E}_{y, \mathbf{x}} [y \sigma_*(\mathbf{x}; \theta)] \quad (6.2.5)$$

$$= -\mathbb{E}_{y, \mathbf{x}} [y a \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)],$$

$$U(\theta_1, \theta_2) = \mathbb{E}_{\mathbf{x}} [\sigma_*(\mathbf{x}; \theta_1) \sigma_*(\mathbf{x}; \theta_2)] \quad (6.2.6)$$

$$= \mathbb{E}_{\mathbf{x}} [a_1 \sigma(\langle \mathbf{w}_1, \mathbf{x} \rangle) a_2 \sigma(\langle \mathbf{w}_2, \mathbf{x} \rangle)].$$

We can then rewrite Eq. (6.2.4) as

$$\dot{\theta}_i^t = -\nabla_{\theta} V(\theta_i^t) - \frac{1}{N} \sum_{j=1}^N \nabla U(\theta_i^t, \theta_j^t). \quad (6.2.7)$$

This expression makes it clear that the right-hand side depends only on θ_i^t and on the empirical distribution of the neurons

$$\hat{\rho}_{N,t} := \frac{1}{N} \sum_{i=1}^N \delta_{\theta_i^t}. \quad (6.2.8)$$

Indeed, we can further rewrite Eq. (6.2.7)

$$\dot{\theta}_i^t = -\nabla_{\theta} V(\theta_i^t) - \int \nabla V(\theta_i^t, \theta') \hat{\rho}_{N,t}(\mathrm{d}\theta'). \quad (6.2.9)$$

We can put on our physicist’s hat and interpret Eq. (6.2.9) as describing the evolution of N particles, where the velocity of each particle is a function of the density at time t . Hence the density must satisfy a continuity partial differential equation (PDE):

$$\partial_t \hat{\rho}_{N,t} = \nabla_{\boldsymbol{\theta}} \cdot (\hat{\rho}_{N,t}(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \hat{\rho}_{N,t})), \quad (6.2.10)$$

where

$$\Psi(\boldsymbol{\theta}; \rho) = V(\boldsymbol{\theta}) + \int U(\boldsymbol{\theta}, \boldsymbol{\theta}') \rho(d\boldsymbol{\theta}'). \quad (6.2.11)$$

Let us point out that Eq. (6.2.10) is a completely equivalent description of the gradient flow (6.2.4), and holds for any finite N , provided one is careful about defining gradients of delta functions³ and similar technicalities.

The formulation of Eq. (6.2.10) has two interesting properties:

- The evolution takes place in the same space for different values of N , this is the space of probability measures $\hat{\rho}_t$ in $d + 1$ dimensions. While we initially introduced Eq. (6.2.10) for probability measures that are sums of point masses, the same PDE makes sense more generally.
- Equation (6.2.10) ‘factors out’ the invariance of Eq. (6.2.4) under permutations of the neurons $\{1, \dots, N\}$.

Because of these reasons, it is very easy (at least formally) to take the $N \rightarrow \infty$ limit in Eq. (6.2.10). Assume that the initialization satisfies $\hat{\rho}_{N,0} \Rightarrow \rho_{\text{init}}$ as $N \rightarrow \infty$, where \Rightarrow denotes the weak convergence of measures. For instance this is the case if

$$\{(a_i^0, \mathbf{w}_i^0)\}_{i \leq N} \sim_{iid} \rho_{\text{init}}. \quad (6.2.12)$$

Then (under suitable technical conditions) it can be proven that $\hat{\rho}_{N,t} \Rightarrow \rho_t$ for any t , where ρ_t satisfies

$$\begin{aligned} \partial_t \rho_t &= \nabla_{\boldsymbol{\theta}} \cdot (\rho_t(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \Psi(\boldsymbol{\theta}; \rho_t)), \\ \rho_0 &= \rho_{\text{init}}. \end{aligned} \quad (6.2.13)$$

We also note that the potential $\Psi(\boldsymbol{\theta}; \rho)$ can be interpreted as the first order derivative of the population error with respect to a change of density of neuron at $\boldsymbol{\theta}$. Using physics notations:

$$\Psi(\boldsymbol{\theta}; \rho) = \frac{\delta R(\rho)}{\delta \rho(\boldsymbol{\theta})}, \quad (6.2.14)$$

where we defined

$$\begin{aligned} R(\rho) &:= \frac{1}{2} \mathbb{E} \left\{ \left(f_*(\mathbf{x}) - \int \sigma_*(\mathbf{x}; \boldsymbol{\theta}) \rho(d\boldsymbol{\theta}) \right)^2 \right\} \\ &= \frac{1}{2} \mathbb{E}(f_*(\mathbf{x})^2) + \int V(\boldsymbol{\theta}) \rho(d\boldsymbol{\theta}) + \frac{1}{2} \int U(\boldsymbol{\theta}, \boldsymbol{\theta}') \rho(d\boldsymbol{\theta}) \rho(d\boldsymbol{\theta}'). \end{aligned} \quad (6.2.15)$$

In conclusion, by scaling the function $f(\mathbf{x}; \boldsymbol{\Theta})$, we obtained a different limit description of infinitely wide networks. Rather than a linearly parametrized model, which is fitted via min-norm regression, we obtained a fully nonlinear model.

³More precisely, Eq. (6.2.10) has to hold in weak sense, i.e. integrated against a suitable test function.

Remark 6.2.1. Nonlinear continuity equations of the form (6.2.13) have been studied for a long time in mathematics, starting from the seminal work of Kac [Kac56] and McKean [MJ66]. Within this literature, this is known as the McKean-Vlasov equation. We refer to Appendix B for further pointers to this line of work.

6.3 Learning ridge functions in mean field: basics

Summarizing, the two-layer neural network (6.2.1) behaves in different ways depending on how we scale the coefficient α_N . In particular:

- If $\alpha_N \asymp 1/\sqrt{N}$, for large enough N , the neural network is well approximated by the neural tangent model. In particular, for $n \leq d^{1+\delta}$, the network only learns the projection of the target function onto linear functions.
- If $\alpha_N \asymp 1/\sqrt{N}$, the SGD trained network converges as $N \rightarrow \infty$ to the network $f(\mathbf{x}; \rho_t) := \int a \sigma(\langle \mathbf{w}, \mathbf{x} \rangle) \rho_t(da, d\mathbf{w})$, where ρ_t solves the PDE (6.2.13).

What is the behavior of the PDE (6.2.13) under the data distribution introduced in Section 6.1, namely $\mathbf{x}_i \sim \mathbf{N}(0, \mathbf{I}_d)$, $y_i = \varphi(\langle \mathbf{w}_*, \mathbf{x}_i \rangle) + \varepsilon_i$? A fully rigorous treatment of this question is somewhat intricate and goes beyond the scope of this overview. We will limit ourselves to some simple considerations, and overview recent progress on this topic in Appendix B. We will assume that the conditions of Proposition 6.1.2 hold.

Consider an uninformative initialization. In the present case, this means the first layer weights \mathbf{w}_i have spherically symmetric distribution. For the sake of concreteness, we can assume them to be Gaussian and take

$$\rho_{\text{init}}(da, d\mathbf{w}) = P_A \otimes \mathbf{N}(\mathbf{0}, \gamma^2 \mathbf{I}_d/d), \quad (6.3.1)$$

where the distribution P_A of the second-layer weights can be arbitrary. (The scaling of the covariance above with d is chosen so that $\|\mathbf{w}_i\| = \Theta(1)$ for large d .)

Now, the data distribution is invariant under rotations that leave \mathbf{w}_* unchanged. As a consequence, it is not hard to show that the evolution (6.2.13) is equivariant under the same group. Namely, let $\mathbf{R} \in \mathbb{R}^{d \times d}$ be a rotation such that $\mathbf{R}\mathbf{w}_* = \mathbf{w}_*$. For a probability distribution $\rho(da, d\mathbf{w})$ on $\mathbb{R} \times \mathbb{R}^d$, we let $\mathbf{R}_\# \rho$ be the push forward under this rotation⁴ (acting trivially on the first factor \mathbb{R}). Then if ρ_0 and $\rho'_0 = \mathbf{R}_\# \rho_0$ are two initializations for the PDE, if ρ_t, ρ'_t are the corresponding solutions to Eq. (6.2.13), then we necessarily have $\rho'_t = \mathbf{R}_\# \rho_t$.

Since the initialization (6.3.1) is invariant under such rotations, ρ_t will be invariant for all t . In other words, under ρ_t , (a, \mathbf{w}) is uniformly random, given $(a, \langle \mathbf{w}_*, \mathbf{w} \rangle, \|\mathbf{P}_\perp \mathbf{w}\|_2)$, where \mathbf{P}_\perp is the projection orthogonal to \mathbf{w}_* . We can therefore write a PDE for the joint distribution of these three quantities which we denote by $\bar{\rho}_t(da, ds, dr)$. Writing $\mathbf{z} = (a, s, r)$, $\bar{\rho}_t$ is obtained by solving the PDE

$$\partial_t \bar{\rho}_t = \partial_a (\bar{\rho}_t \partial_a \Psi(\mathbf{z}; \bar{\rho}_t)) + \partial_s (\bar{\rho}_t \partial_s \Psi(\mathbf{z}; \bar{\rho}_t)) + \frac{1}{r} \partial_r (r \bar{\rho}_t \partial_r \Psi(\mathbf{z}; \bar{\rho}_t)). \quad (6.3.2)$$

The initialization (6.3.1) translates into

$$\bar{\rho}_{\text{init}} = P_A \otimes \mathbf{N}(0, \gamma^2/d) \otimes \mathbf{Q}_{d-1, \gamma}, \quad (6.3.3)$$

⁴ $\mathbf{R}_\# \rho$ is the law of $(a, \mathbf{R}\mathbf{w})$ when $(a, \mathbf{w}) \sim \rho$.

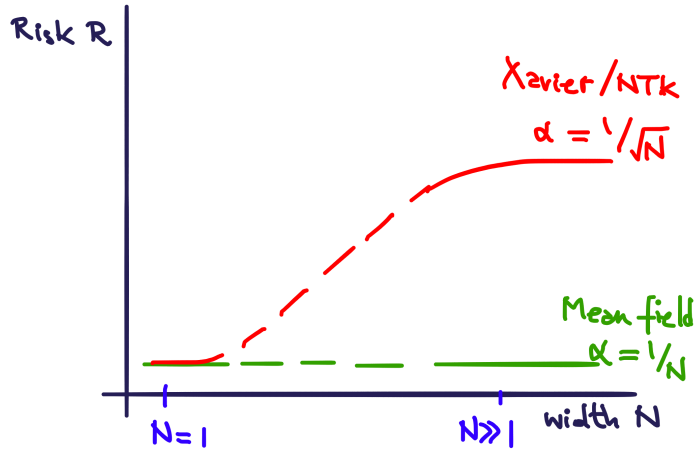


Figure 6.1: Conjectured behavior of the prediction error as a function of width for different scalings of the network.

where $Q_{k,\gamma}$ is the law of the square root of a chi-squared with k degrees of freedom, rescaled by $\gamma/\sqrt{k+1}$.

Let $R(\bar{\rho}_t)$ be the prediction risk of Eq. (6.2.15), written in terms of $\bar{\rho}_t$ (using the fact that \mathbf{w} is uniformly random conditional on a, s, r). It follows from the general theory that $t \mapsto R(\bar{\rho}_t)$ is non-decreasing. If it can be shown that

$$\lim_{t \rightarrow \infty} R(\bar{\rho}_t) \leq \varepsilon_0, \quad (6.3.4)$$

then this directly implies that, for any $\varepsilon > 0$, SGD achieves test error upper bounded (with high probability) by $\varepsilon_0 + \varepsilon$, from $O(d)$ samples. Indeed, the PDE (6.2.13) can be shown to approximate SGD provided the stepsize satisfies $\eta \leq 1/(C_0 d)$ and therefore a time of order one translates in a number of iterations (or samples) of order d (assuming online SGD).

Indeed we expect Eq. (6.3.4) to hold with $\varepsilon_0 = 0$ in many cases of interest, e.g. under the assumptions on $\sigma = \varphi$ in Proposition 6.1.2. A proof of this fact would be too long for these notes. Partial evidence is provided by the following facts:

1. The reduced PDE (6.3.2) is 3-dimensional and can be solved numerically, confirming Eq. (6.3.4).
2. For $P_A = \delta_1$ and γ small, if we change SGD and do not update the second layer weights, the resulting learning dynamics is similar to the one of a single neuron, covered by Proposition 6.1.2, and therefore an approximation argument can be employed to bound the risk.

We conclude that, under the setting of Proposition 6.1.2, very wide neural networks with mean-field scaling learn ridge functions from $O(d)$ samples.

While we do not have a precise characterization of the risk for intermediate values of N , we sketch in Figure 6.1 the (partly conjectural) behavior of the prediction errors for different scalings of the network, when learning a ridge function with $\varphi = \sigma$.

Appendix A

Summary of notations

For a positive integer, we denote by $[n]$ the set $\{1, 2, \dots, n\}$. For vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, we denote $\langle \mathbf{u}, \mathbf{v} \rangle = u_1 v_1 + \dots + u_d v_d$ their scalar product, and $\|\mathbf{u}\|_2 = \langle \mathbf{u}, \mathbf{u} \rangle^{1/2}$ the ℓ_2 norm. We denote by $\mathbb{S}^{d-1}(r) = \{\mathbf{u} \in \mathbb{R}^d : \|\mathbf{u}\|_2 = r\}$ the sphere of radius r in d dimensions (we will simply write for the unit ball $\mathbb{S}^{d-1} := \mathbb{S}^{d-1}(1)$).

Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, we denote $\|\mathbf{A}\|_{\text{op}} = \max_{\|\mathbf{u}\|_2=1} \|\mathbf{A}\mathbf{u}\|_2$ its operator norm and by $\|\mathbf{A}\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$ its Frobenius norm. If $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a square matrix, the trace of \mathbf{A} is denoted by $\text{Tr}(\mathbf{A}) = \sum_{i \in [n]} A_{ii}$. Further, given a positive semidefinite matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$ and for vectors $\mathbf{u} \in \mathbb{R}^d$, we denote $\|\mathbf{u}\|_{\mathbf{B}} = \|\mathbf{B}^{1/2}\mathbf{u}\|_2 = \langle \mathbf{u}, \mathbf{B}\mathbf{u} \rangle^{1/2}$ the weighted ℓ_2 norm.

Given a probability space (\mathcal{X}, ν) , we denote $L^2(\mathcal{X}) := L^2(\mathcal{X}, \nu)$ the space of square integrable functions on (\mathcal{X}, ν) , and $\langle f, g \rangle_{L^2(\mathcal{X})} = \mathbb{E}_{\mathbf{x} \sim \nu} \{f(\mathbf{x})g(\mathbf{x})\}$ and $\|f\|_{L^2(\mathcal{X})} = \langle f, f \rangle_{L^2(\mathcal{X})}^{1/2}$ respectively the scalar product and norm on $L^2(\mathcal{X})$ (we will sometimes write $\langle \cdot, \cdot \rangle_{L^2}$ when (\mathcal{X}, ν) is clear from context). The space $L^2(\mathbb{R}, \gamma)$ plays an important role throughout these notes, where $\gamma(dx) = e^{-x^2/2} dx / \sqrt{2\pi}$ is the standard Gaussian measure. We will often prefer the notation $G \sim \mathbf{N}(0, 1)$ to denote the standard normal distribution, and $L^2(\mathbf{N}(0, 1)) := L^2(\mathbb{R}, \gamma)$. The set of Hermite polynomials $\{\text{He}_k\}_{k \geq 0}$ forms an orthogonal basis of $L^2(\mathbb{R}, \gamma)$, where He_k has degree k . We follow the classical normalization:

$$\langle \text{He}_k, \text{He}_j \rangle_{L^2(\mathbf{N}(0,1))} = \mathbb{E}_{G \sim \mathbf{N}(0,1)} \{\text{He}_k(G)\text{He}_j(G)\} = k! \mathbb{1}_{j=k}.$$

In particular, for any function $g \in L^2(\mathbf{N}(0, 1))$, we have the decomposition

$$g(x) = \sum_{k=0}^{\infty} \frac{\mu_k(g)}{k!} \text{He}_k(x), \quad \mu_k(g) := \mathbb{E}_{G \sim \mathbf{N}(0,1)} \{g(G)\text{He}_k(G)\}, \quad \|g\|_{L^2}^2 = \sum_{k=0}^{\infty} \frac{(\mu_k(g))^2}{k!},$$

where $\mu_k(g)$ is sometimes referred to as the k -th Hermite coefficient of g .

We use $O_d(\cdot)$ (resp. $o_d(\cdot)$) for the standard big-O (resp. little-o) relations, where the subscript d emphasizes the asymptotic variable. Furthermore, we write $f = \Omega_d(g)$ if $g(d) = O_d(f(d))$, and $f = \omega_d(g)$ if $g(d) = o_d(f(d))$. Finally, $f = \Theta_d(g)$ if we have both $f = O_d(g)$ and $f = \Omega_d(g)$.

We use $O_{d,\mathbb{P}}(\cdot)$ (resp. $o_{d,\mathbb{P}}(\cdot)$) the big-O (resp. little-o) in probability relations. Namely, for $h_1(d)$ and $h_2(d)$ two sequences of random variables, $h_1(d) = O_{d,\mathbb{P}}(h_2(d))$ if for any $\varepsilon > 0$, there exists $C_\varepsilon > 0$ and $d_\varepsilon \in \mathcal{Z}_{>0}$, such that

$$\mathbb{P}(|h_1(d)/h_2(d)| > C_\varepsilon) \leq \varepsilon, \quad \forall d \geq d_\varepsilon,$$

and respectively: $h_1(d) = o_{d,\mathbb{P}}(h_2(d))$, if $h_1(d)/h_2(d)$ converges to 0 in probability. Similarly, we will denote $h_1(d) = \Omega_{d,\mathbb{P}}(h_2(d))$ if $h_2(d) = O_{d,\mathbb{P}}(h_1(d))$, and $h_1(d) = \omega_{d,\mathbb{P}}(h_2(d))$ if $h_2(d) = o_{d,\mathbb{P}}(h_1(d))$. Finally, $h_1(d) = \Theta_{d,\mathbb{P}}(h_2(d))$ if we have both $h_1(d) = O_{d,\mathbb{P}}(h_2(d))$ and $h_1(d) = \Omega_{d,\mathbb{P}}(h_2(d))$.

Appendix B

Literature survey

Chapter 1 (The linear regime)

The connection between neural networks trained by gradient descent and kernel methods was first¹ elucidated in [JGH18]. They showed that, under a specific scaling of weights at initialization and learning rate, neural networks trained by gradient flow converge to a kernel regression solution in the infinite-width limit ($N \rightarrow \infty$). The specific kernel of this solution was termed the *neural tangent kernel* (NTK). A striking implication of this finding is that gradient flow converges to zero training error—a global minimizer—despite the non-convexity of the full optimization landscape. Following this intuition, [LL18, DZPS18] proved—under the same scaling of parameters—global convergence of gradient descent for two-layer neural networks with ReLU activation and sufficiently large but finite number of neurons. Subsequent studies extended the proof of global convergence to deep neural networks and more general architectures [DLL⁺19, AZLS19b, AZLS19a, LXS⁺19, ZCZG20]. However, these results require stringent overparametrization conditions on the network width. For example, [DZPS18] requires two-layer neural networks with a number of neurons N of at least $\tilde{\Omega}(n^6/\lambda_0^4)$, where n is the training sample size and λ_0 is the minimum eigenvalue of the empirical neural tangent kernel matrix. With the goal of establishing global convergence for networks with realistic widths, a large body of work has steadily pushed down the overparametrization condition on N [JT19, KH19, SY19, ZG19, OS19, OS20, ZCZG20]. In particular, [OS20] showed that under mild assumptions on the data distribution, quadratic overparametrization $Nd = \Omega(n^2)$ is sufficient for global convergence of two-layer neural networks. This condition can be further improved to near optimal overparametrization $Nd = \Omega(n \log(n))$ by modifying the parameter scaling and choosing a suitable initialization, as shown in [BMR21]. In addition to global convergence of the training error, a number of studies have considered the generalization properties of neural networks in this regime and proved that low test error can be achieved under various data distribution assumptions [LL18, ADH⁺19a, CFW⁺21, CG19, NCS19, JT19].

As shown in [COB19], the connection to kernel methods is not restricted to neural networks, but extends to more general non-linear models trained by gradient descent. Indeed, it originates from a particular choice in the scaling of the model parameters. With this specific scaling, a small change in the weights produces a large change in the output of the non-linear model but only a small change in its Hessian (see also discussion in [LZB20]). As a consequence, neural networks

¹Several connections between neural networks and kernel methods have been discussed earlier in the literature, e.g. see [Nea95, CS09, DFS16, LBN⁺18, NXB⁺18] and references therein. However, in these cases, the correspondence with kernel models only holds for random neural networks, i.e. at initialization of the SGD dynamics, while [JGH18] holds, under certain conditions, for the entire training trajectory.

in this optimization regime have their weights barely moving, while they converge to 0 training loss. This led [COB19] to dub this regime the *lazy training regime*. Notably, the network can be approximated throughout training by its linearization around its random weights at initialization, which is the starting point for these notes.

For overparametrized neural networks, i.e. underconstrained optimization problems, there exist many global minimizers with zero training loss. These interpolators achieve very different generalization errors. Despite being trained with no explicit regularization that would promote ‘good’ models, the solutions found in practice generalize well to test data [ZBH⁺21]. A popular explanation for this performance postulates an ‘implicit regularization’ from the training algorithm itself [NTS14, NBMS17]. In words, generalization is implicitly controlled by the dynamics of the optimization algorithm, which selects a good minimizer, and not by adding an explicit regularization to the risk. For example, in the linear regime, gradient descent selects the minimizer closest from the initialization in weighted ℓ_2 distance (see Remark 1.4.1). Major research activity has been devoted in recent years to characterizing implicit regularization for various algorithms, including gradient descent in matrix factorization [GWB⁺17, LLL20], mirror descent [GLSS18a], gradient descent on separable data [SHN⁺18, JT18b], linear neural networks [GLSS18b, JT18a, YKM20] and neural networks with homogeneous activation functions [LL19, JT20]. Despite these efforts, the precise complexity measure that is implicitly controlled by SGD in neural networks remains poorly understood, except in restricted settings.

Chapter 2 (Linear regression under feature concentration)

In the case of Gaussian features, this model is also known as the ‘Gaussian design model’ [DW18]. The asymptotic risk of ridge regression with random features was computed by [Dic16] for isotropic Gaussian features $\mathbf{x}_i \sim \mathcal{N}(0, \mathbf{I}_p)$ in the proportional asymptotics $p, n \rightarrow \infty$ with $p/n \rightarrow \gamma \in (0, \infty)$. These results were generalized by [DW18] to anisotropic features $\mathbf{x}_i = \Sigma^{1/2} \mathbf{z}_i$, where \mathbf{z}_i has independent entries with bounded 12th moment. [HMRT22] later derived similar results under weaker assumptions on (β_*, Σ) .

The fact that ridge regression with isotropic random features displayed a ‘double descent’ pattern was pointed out in [AS17]. However, reproducing the full phenomenology observed empirically, with the minimum risk achieved at large overparametrization, require coefficient vectors β_* aligned with Σ . [RMR21, WX20] computed the asymptotic risk in such settings. [HMRT22] derived similar results under weaker assumptions on (β_*, Σ) . The results in [HMRT22] hold *non-asymptotically*, with explicit error bounds, using the anisotropic local law proved in [KY17].

The works listed above only considered the proportional parametrization scaling, i.e., $C^{-1} \leq p/n \leq C$, which prohibits features coming from highly overparametrized ($p \gg n$) or kernel ($p = \infty$) models. A general scaling (including $p = \infty$) was recently studied in [CM22]. Using a novel martingale argument to prove deterministic equivalents, they derived non-asymptotic predictions for the test error, with nearly optimal rates for the error bounds on the bias and variance terms, namely $\tilde{O}(n^{-1/2})$ and $\tilde{O}(n^{-1})$ which match the fluctuations of respectively the local law and average law for the resolvent.

Despite the simplicity of this first model, it is connected to more complex models via universality. For example, it was shown in some settings that kernel regression [Mis22] and random feature regression [MM19] have the same asymptotic test errors as some equivalent Gaussian models, where the non-linear feature maps are replaced by Gaussian vectors with matching first two moments.

The recent interest in interpolators was prompted by the experimental results in [ZBH⁺21,

[BMM18] which showed that deep neural networks and kernel methods can generalize well even when they interpolate *noisy data*. This surprising phenomenon —often referred to as *benign overfitting* following [BLLT20]— is at odds with the classical picture in statistics, where we expect interpolating noisy training data to lead to unreasonable models with large test errors. Recent work has instead proved that several standard learning models can indeed interpolate benignly under certain conditions, including the Nadaraya-Watson estimator [BRT19], kernel ridgeless regression [LR20, GMMM21] and max-margin linear classifiers [MRSY19, Sha22]. As a testbed to understand interpolation learning, much focus has been devoted to studying interpolation in linear regression with (sub-)Gaussian features [HMRT22, BLLT20, TB20, MVSS20, KZSS21]. In particular, [BLLT20] showed sufficient conditions for consistency of the minimum ℓ_2 -norm interpolating solution in terms of an effective rank, that were later refined in [CM22]. Further a number of studies have argued that benign overfitting in (kernel) ridge regression is a high dimensional phenomenon [RZ19, BBP22].

Chapter 3 (Kernel ridge regression)

The test error of kernel ridge regression (KRR) was investigated in a number of papers in the past [BBM05, CDV07, Wai19]. In particular, [CDV07] showed that KRR achieves minimax optimal rates over classes of functions under certain capacity condition on the kernel operator and source condition on the target function. However these earlier results have limitations. First, they require a strictly positive ridge regularization and therefore do not apply to interpolators. Second, they often only characterize tightly the decay rate of the error, i.e. as $n \rightarrow \infty$ for fixed dimension d . In these notes, we are instead interested in deriving the precise test error of KRR in high dimension where both n and d grow simultaneously. Finally, in contrasts to earlier work, the results discussed in these chapters hold for a given target function f_* (instead of worst case over a class of function).

The study of kernel models in high dimension was initiated by the seminal work [EK10]. El Karoui analysed empirical kernel matrices of the form $\mathbf{K} = (h(\langle \mathbf{x}_i, \mathbf{x}_j \rangle / d))_{i,j \leq n} \in \mathbb{R}^{n \times n}$ in the proportional asymptotics $n \propto d$, with $\mathbf{x}_i = \Sigma^{1/2} \mathbf{z}_i$ where $\mathbf{z}_i \in \mathbb{R}^d$ has independent entries with bounded $4 + \varepsilon$ moment. [EK10] showed that \mathbf{K} is well approximated by its linearization (here in the isotropic case $\Sigma = \mathbf{I}_d$)

$$\mathbf{K} \approx (h(0) + h''(0)/(2d))\mathbf{1}\mathbf{1}^\top + h'(0)\mathbf{G} + (h(1) - h(0) - h'(0))\mathbf{I}_n, \quad (\text{B.0.1})$$

where $\mathbf{G} = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle / d)_{i,j \leq n}$ is the Gram matrix. This result was later used to bound the asymptotic prediction error of KRR in the proportional regime [LR20, LLS21, BMR21]. In particular, KRR can learn at most a linear approximation to the target function in this regime.

In order to study a more realistic scenario, where $n \gg d$ with n, d large, several works have considered a more general *polynomial asymptotic scaling*, with $n \asymp d^\kappa$ for fixed $\kappa \in \mathbb{R}_{>0}$ as $d, n \rightarrow \infty$ [GMMM21, MMM22, LRZ20, GMMM20, MMM21, MM21, Xia22]. In the case of covariates uniformly distributed on the d -dimensional sphere, [GMMM21] generalized Eq. (B.0.1) and showed that the empirical kernel matrix can be approximated by its degree- $\lfloor \kappa \rfloor$ polynomial approximation. Using this decomposition, [GMMM21] showed that for $\kappa \notin \mathbb{N}$, KRR fits the best degree- $\lfloor \kappa \rfloor$ polynomial approximation (see Section 3.4 in Chapter 3). [MMM22] extended these results to a more general KRR setting under abstract conditions on the kernel operator, namely hypercontractivity of the top eigenfunctions and a spectral gap property (the number of eigenvalues λ_i such that $d^\delta/n \geq \lambda_i \geq d^{-\delta}/n$ is smaller than $n^{1-\delta}$ for some $\delta > 0$). In this setting, [MMM22] showed that KRR effectively acts as a shrinkage operator with some effective regularization $\lambda^{\text{eff}} > \lambda$. Specifically, denoting $(\psi_{d,j})_{j \geq 1}$ the eigenfunctions and $(\lambda_{d,j})_{j \geq 1}$ the eigenvalues in nonincreasing order of

the kernel operator \mathbb{K}_d , the KRR solution with target function f_* and regularization λ is given by

$$f_*(\mathbf{x}) = \sum_{j=1}^{\infty} c_j \psi_{d,j}(\mathbf{x}) \quad \Rightarrow \quad \hat{f}_\lambda = \sum_{j=1}^{\infty} \frac{\lambda_{d,j}}{\lambda_{d,j} + \frac{\lambda^{\text{eff}}}{n}} c_j \psi_{d,j}(\mathbf{x}) + \Delta, \quad (\text{B.0.2})$$

where $\|\Delta\|_{L^2} = o_{d,\mathbb{P}}(1)$. Hence, KRR essentially fits the projection of the target function on the top $O(n)$ eigenfunctions with $\lambda_{d,j} \gg \lambda^{\text{eff}}/n$, and none of the components of f_* with $\lambda_{d,j} \ll \lambda^{\text{eff}}/n$. In the setting of inner product kernels on the sphere [GMMM21], the spectral decay property is satisfied for $n \asymp d^\kappa$, $\kappa \notin \mathbb{N}$. In this case, $\lambda^{\text{eff}} = \Theta(1)$ and indeed all spherical harmonics of degree less or equal to $\lfloor \kappa \rfloor$ have $\lambda_{d,j} \gg \lambda^{\text{eff}}/n$, while higher degrees spherical harmonics have $\lambda_{d,j} \ll \lambda^{\text{eff}}/n$, and we recover that KRR fits the projection of f_* onto degree- $\lfloor \kappa \rfloor$ polynomials.

If the spectral decay property is not satisfied, the estimator (B.0.2) needs to be rescaled by a constant coming from random matrix theory effects, as shown in Chapter 2. The case $n \asymp d^\ell$, $\ell \in \mathbb{N}$ for data uniformly distributed on the sphere was studied recently in three concurrent papers [Mis22, HL22a, XHM⁺22]. They showed that the contribution of degree- ℓ spherical harmonics to the empirical kernel matrix behaves as a Wishart matrix. In particular, as n approaches $d^\ell/\ell!$, the condition number of the Wishart matrix diverges and a peak can appear in the risk curve.

Finally, note that asymptotics for the KRR test error were also heuristically derived either by conjecturing an equivalence with Gaussian feature models [JSS⁺20] or using statistical physics heuristics [CBP21, CLKZ21, CLKZ22].

Chapter 4 (Random Feature model)

The random feature (RF) model was initially introduced as a finite-rank randomized approximation to kernel methods [BBV06, RR07]. The connection between neural networks and RF models was originally pointed out by [Nea95, Wil96] and has recently attracted renewed attention through the NTK and Gaussian process descriptions of wide neural networks [NXB⁺18, DMHR⁺18, LBN⁺18]. [RR07] showed that the empirical random feature kernel converges pointwise to the asymptotic kernel ($N = \infty$). Note that this pointwise convergence does not provide any control on the performance of RF when both N, n are allowed to grow together. Subsequent works [RR08, Bac17b, RR17] derived bounds on the approximation and generalization errors of RF models. [RR08] proved a minimax upper bound $O(1/\sqrt{n} + 1/\sqrt{N})$ on the generalization error, but their results are limited to Lipschitz losses and require a stringent condition on $\|\mathbf{a}\|_\infty$. More recently, [RR17] studied the case of the square loss and proved that for f_* in the RKHS, $N = C\sqrt{n} \log n$ random features are sufficient to achieve the same error rate $O(n^{-1/2})$ as the limiting KRR. This is in contrast to the requirement $N = O(n^{1+\delta})$ described in these notes, the core difference lying in the assumption on the target function f_* (see discussion in [MMM22]). Furthermore, results in [RR17] required positive ridge regularization and only characterized the minimax error rate as $n \rightarrow \infty$ for a fixed RKHS (fixed d). In these notes, we will consider instead studying the RF model in high dimension when N, n, d all grow together.

A number of works [MM19, LCM20, AP20, ALP22, MMM22] have studied the asymptotic risk of ridge regression with random features in high dimension. In particular, [MM19] derives the complete analytical predictions for the test error in the proportional regime $n \asymp d$ and $N \asymp d$, in the case of covariates and weights both uniformly distributed on the d -dimensional sphere. These asymptotics capture in detail the double descent phenomenon in this model. In particular, [MM19] provides precise conditions for the highly overparametrized ($N/n \rightarrow \infty$) and interpolating ($\lambda \rightarrow 0+$) solution to be optimal among random feature models. The polynomial scaling for

random feature models was first considered in [GMMM21] which focused on the approximation error ($n = \infty$) with data and weights uniformly distributed on the sphere. They show that RF models with $d^{\ell+\delta} \leq N \leq d^{\ell+1-\delta}$ random features can only approximate the projection of the target function on degree- ℓ polynomials. The generalization error was studied in a subsequent work [MMM22]. They consider abstract conditions on the eigendecomposition of the activation function, namely hypercontractivity of the top eigenfunctions, an eigenvalue gap (i.e. there exists $m \leq d^{-\delta}s$, where $s = \min(N, n)$, such that $\lambda_{d,m} \geq d^\delta/s$ and $\lambda_{d,m+1} \leq d^{-\delta}/s$ for some constant $\delta > 0$) and $\max(N/n, n/N) \geq d^\delta$. Under these assumptions, [MMM22] shows that RF ridge regression fits exactly the projection of the target function on the top $m \leq \min(n, N)$ eigenfunctions. Applying these results to covariates and weights uniformly distributed on the sphere, this shows that for $d^{\ell_1+\delta} \leq N \leq d^{\ell_1-\delta}$, $d^{\ell_2+\delta} \leq n \leq d^{\ell_2-\delta}$ and $\max(N/n, n/N) \geq d^\delta$, RF ridge regression fits the degree- $\min(\ell_1, \ell_2)$ polynomial approximation to f_* .

Finally, in certain cases (for instance, ridge regression), the non-linear random feature model is connected through universality to a simpler linear Gaussian covariate model [GLK⁺20, GLR⁺22, HL22b, MS22].

Chapter 5 (Neural Tangent model)

In the linear regime, the optimization and generalization properties of neural networks depend crucially on the empirical neural tangent kernel matrix and in particular, its smallest eigenvalue. For example, [DZPS18] showed that for two layer neural networks with width $N \geq Cn^6/\lambda_{\min}(\mathbf{K}_n)^4$, where \mathbf{K}_n is the infinite-width empirical NTK matrix ($N = \infty$), the empirical kernel is well conditioned throughout training and neural networks converge to zero training error. A number of works have studied the empirical kernel matrices arising from the neural tangent model [FW20, LZB20, OS20, ZCZG20].

The approximation error of the neural tangent model associated to two-layer neural networks was studied in [GMMM21] for covariates and first layer weights uniformly distributed on the d -dimensional sphere. They prove that for $d^{\ell-1+\delta} \leq N \leq d^{\ell-\delta}$, the neural tangent model exactly fits the best degree- ℓ polynomial approximation to the target function. Namely, the neural tangent model has the same approximation power as the random feature model, provided we match the number of parameters $p = Nd$ for NT model and $p = N$ for RF model. Note that this equivalence does not hold when data is anisotropic, e.g. see [GMMM20]. The generalization error of the neural tangent model was studied in a subsequent paper [MZ22]. They first show that as long as $Nd/\log(Nd)^C \geq n$, the empirical neural tangent kernel has eigenvalues bounded away from zero, and the neural tangent model can exactly interpolate arbitrary labels. In the same regime, [MZ22] prove that ridge regression with the neural tangent model is well approximated by kernel ridge regression with the infinite width kernel.

Chapter 6 (Beyond the linear regime)

Several studies have empirically investigated the relevance of the linear regime to describe neural networks used in practice. They show that, in some settings, the neural tangent model captures the behavior of SGD-trained neural networks, at least at the beginning of the dynamics [LXS⁺19, GSJW20, WGL⁺20]. Furthermore, it was shown that in some settings NTK can achieve superior performance compared to standard kernels [ADH⁺19b, LWY⁺19, SFG⁺20] and even state-of-the-art performance on some datasets [ADL⁺19]. However, neural networks in the linear regime and kernel methods typically fall short in comparison to state-of-the-art neural networks. The

scenario in which weights move significantly away from their random initialization is referred as *rich* or *feature learning regime*.

A major theoretical achievement of the nineties [DHM89, Hor91, Bar93, Pin99] was to prove the approximation theoretic advantage of non-linear neural networks compared to fixed basis methods. In particular, Barron showed in his celebrated work [Bar93] that two-layer neural networks can approximate, with few neurons, functions with fast decay of their high frequency components, while linear methods need a number of parameters exponential in the dimension to approximate the same class of functions (in worst case). More recent works have established tighter lower bounds—that are cursed by dimensionality—on the performance of kernel and random feature models [Bac17a, YS19, VW19, GMMM21]. For example, [YS19] proves that a super-polynomial in d number of random features is required to approximate a single neuron.

While approximation theory deals with ideal representations—which might not be tractable to find in practice—, a recent line of work have sought to display theoretical settings where neural networks *trained by gradient descent* provably outperform neural tangent and kernel models [Bac17a, WGL⁺19, AZL19, AZL20, GMMM19, GMMM20, FDZ19, CB20, DM20]. In particular, much attention has been devoted to the setting of learning ridge functions, i.e. functions that only depend on a small number of (unknown) relevant directions $f_*(\mathbf{x}) = \psi(\mathbf{U}_*^\top \mathbf{x})$ with $\mathbf{U}_*^\top \mathbf{U}_* = \mathbf{I}_k$ [CBL⁺20, RGKZ21, AAM22, DLS22, MHPG⁺22, BBSS22, AAM23, BMZ23]. These functions are also known in the literature as single-index models in the case $k = 1$ and multiple-index models for $k > 1$. The reason for this interest is that learning ridge functions offer a simple setting where a clear separation exists between neural networks trained non-linearly and kernel methods. Indeed, linear methods are oblivious to latent linear structures [Bac17a, GMMM21, AAM22], while gradient descent has the possibility to align the network weights with the sparse support. The picture that emerges from these works is as follows. The complexity of learning single index models by gradient descent is driven by the *information exponent* [AGJ21], which measures the strength of the correlation between a single neuron and the ridge function at initialization. This notion of information exponent was generalized to multi-index models via the ‘leap complexity’ [AAM22, AAM23].

Hence, while the lazy regime offers a setting where both the optimization and generalization properties of neural networks can be understood precisely, it does not capture the full power of deep learning. A number of approaches have been suggested to go beyond the linear regime. Several groups have proposed higher-order or finite-width corrections to the limiting neural tangent model [HY20, BL19, HN19, DL20]. Another approach examines other infinite-width limits for gradient-trained neural networks. A systematic analysis of the different infinite-widths limits was conducted in [YH20], which characterized all non-trivial limits following an ‘abc-parametrization’ of the scaling of learning rate, model parameters and initialization. In particular, they put forward a new *maximal update parametrization* (μP) which maximizes the change in the network weights after one SGD step among all infinite-width limits.

A popular alternative to the linear regime corresponds to two-layer neural networks trained by stochastic gradient descent in the mean-field regime [CB18, MMN18, RVE18, SS20, MMM19]. This limit coincides with the μP limit for two-layer networks. Unlike the neural tangent approach, the evolution of network weights is now non-linear and described in terms of a Wasserstein gradient flow over the neurons’ weight distribution. However, analyzing the training dynamics requires to track the evolution of a distribution and remains challenging, except in highly-symmetric settings where the PDE is effectively low-dimensional [MMN18, AAM23, BMZ23]. Finally, several papers have proposed extensions of the mean-field limit to multilayer neural networks [NP20, LML⁺20].

Bibliography

- [AAM22] Emmanuel Abbe, Enric Boix Adsera, and Theodor Misiakiewicz, *The merged-staircase property: a necessary and nearly sufficient condition for sgd learning of sparse functions on two-layer neural networks*, Conference on Learning Theory, PMLR, 2022, pp. 4782–4887.
- [AAM23] Emmanuel Abbe, Enric Boix Adserà, and Theodor Misiakiewicz, *Sgd learning on neural networks: leap complexity and saddle-to-saddle dynamics*, The Thirty Sixth Annual Conference on Learning Theory, PMLR, 2023, pp. 2552–2623.
- [ADH⁺19a] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang, *Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks*, International Conference on Machine Learning, PMLR, 2019, pp. 322–332.
- [ADH⁺19b] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang, *On exact computation with an infinitely wide neural net*, Advances in Neural Information Processing Systems **32** (2019).
- [ADL⁺19] Sanjeev Arora, Simon S Du, Zhiyuan Li, Ruslan Salakhutdinov, Ruosong Wang, and Dingli Yu, *Harnessing the power of infinitely wide deep nets on small-data tasks*, International Conference on Learning Representations, 2019.
- [AGJ21] Gerard Ben Arous, Reza Gheissari, and Aukosh Jagannath, *Online stochastic gradient descent on non-convex losses from high-dimensional inference*, The Journal of Machine Learning Research **22** (2021), no. 1, 4788–4838.
- [ALP22] Ben Adlam, Jake A Levinson, and Jeffrey Pennington, *A random matrix perspective on mixtures of nonlinearities in high dimensions*, International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 3434–3457.
- [AP20] Ben Adlam and Jeffrey Pennington, *The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization*, International Conference on Machine Learning, PMLR, 2020, pp. 74–84.
- [AS17] Madhu S Advani and Andrew M Saxe, *High-dimensional dynamics of generalization error in neural networks*, arXiv preprint arXiv:1710.03667 (2017).
- [AZL19] Zeyuan Allen-Zhu and Yuanzhi Li, *What can resnet learn efficiently, going beyond kernels?*, Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 9017–9028.

- [AZL20] ———, *Backward feature correction: How deep learning performs deep learning*, arXiv:2001.04413 (2020).
- [AZLS19a] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song, *A convergence theory for deep learning via over-parameterization*, International Conference on Machine Learning, PMLR, 2019, pp. 242–252.
- [AZLS19b] ———, *On the convergence rate of training recurrent neural networks*, Advances in Neural Information Processing Systems **32** (2019), 6676–6688.
- [Bac17a] Francis Bach, *Breaking the curse of dimensionality with convex neural networks*, The Journal of Machine Learning Research **18** (2017), no. 1, 629–681.
- [Bac17b] ———, *On the equivalence between kernel quadrature rules and random feature expansions*, The Journal of Machine Learning Research **18** (2017), no. 1, 714–751.
- [Bar93] Andrew R Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Transactions on Information theory **39** (1993), no. 3, 930–945.
- [BBM05] Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson, *Local rademacher complexities*.
- [BBP22] Daniel Beaglehole, Mikhail Belkin, and Parthe Pandit, *Kernel ridgeless regression is inconsistent for low dimensions*, arXiv preprint arXiv:2205.13525 (2022).
- [BBSS22] Alberto Bietti, Joan Bruna, Clayton Sanford, and Min Jae Song, *Learning single-index models with shallow neural networks*, Advances in Neural Information Processing Systems **35** (2022), 9768–9783.
- [BBV06] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala, *Kernels as features: On kernels, margins, and low-dimensional mappings*, Machine Learning **65** (2006), 79–94.
- [BHMM19] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal, *Reconciling modern machine-learning practice and the classical bias–variance trade-off*, Proceedings of the National Academy of Sciences **116** (2019), no. 32, 15849–15854.
- [BL19] Yu Bai and Jason D Lee, *Beyond linearization: On quadratic and higher-order approximation of wide neural networks*, International Conference on Learning Representations, 2019.
- [BLLT20] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler, *Benign overfitting in linear regression*, Proceedings of the National Academy of Sciences **117** (2020), no. 48, 30063–30070.
- [BMM18] Mikhail Belkin, Siyuan Ma, and Soumik Mandal, *To understand deep learning we need to understand kernel learning*, International Conference on Machine Learning, PMLR, 2018, pp. 541–549.
- [BMR21] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin, *Deep learning: a statistical viewpoint*, Acta numerica **30** (2021), 87–201.

- [BMZ23] Raphaël Berthier, Andrea Montanari, and Kangjie Zhou, *Learning time-scales in two-layers neural networks*, arXiv:2303.00055 (2023).
- [BRT19] Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov, *Does data interpolation contradict statistical optimality?*, The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1611–1619.
- [BTA11] Alain Berlinet and Christine Thomas-Agnan, *Reproducing kernel hilbert spaces in probability and statistics*, Springer Science & Business Media, 2011.
- [CB18] Lénaïc Chizat and Francis Bach, *On the global convergence of gradient descent for over-parameterized models using optimal transport*, Advances in Neural Information Processing Systems **31** (2018), 3036–3046.
- [CB20] Lenaic Chizat and Francis Bach, *Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss*, Conference on Learning Theory, PMLR, 2020, pp. 1305–1338.
- [CBL⁺20] Minshuo Chen, Yu Bai, Jason D Lee, Tuo Zhao, Huan Wang, Caiming Xiong, and Richard Socher, *Towards understanding hierarchical learning: Benefits of neural representations*, arXiv:2006.13436 (2020).
- [CBP21] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan, *Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks*, Nature communications **12** (2021), no. 1, 2914.
- [CDV07] Andrea Caponnetto and Ernesto De Vito, *Optimal rates for the regularized least-squares algorithm*, Foundations of Computational Mathematics **7** (2007), no. 3, 331–368.
- [CFW⁺21] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu, *Towards understanding the spectral bias of deep learning*, Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, 2021, pp. 2205–2211.
- [CG19] Yuan Cao and Quanquan Gu, *Generalization bounds of stochastic gradient descent for wide and deep neural networks*, Advances in Neural Information Processing Systems **32** (2019), 10836–10846.
- [Chi11] Theodore S Chihara, *An introduction to orthogonal polynomials*, Courier Corporation, 2011.
- [CLKZ21] Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová, *Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime*, Advances in Neural Information Processing Systems **34** (2021), 10131–10143.
- [CLKZ22] ———, *Error rates for kernel classification under source and capacity conditions*, arXiv:2201.12655 (2022).
- [CM22] Chen Cheng and Andrea Montanari, *Dimension free ridge regression*, arXiv:2210.08571 (2022).

- [COB19] Lenaic Chizat, Edouard Oyallon, and Francis Bach, *On lazy training in differentiable programming*, NeurIPS 2019-33rd Conference on Neural Information Processing Systems, 2019, pp. 2937–2947.
- [CS09] Youngmin Cho and Lawrence Saul, *Kernel methods for deep learning*, Advances in neural information processing systems **22** (2009).
- [DFS16] Amit Daniely, Roy Frostig, and Yoram Singer, *Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity*, Advances in neural information processing systems **29** (2016).
- [DHM89] Ronald A DeVore, Ralph Howard, and Charles Micchelli, *Optimal nonlinear approximation*, Manuscripta mathematica **63** (1989), 469–478.
- [Dic16] Lee H Dicker, *Ridge regression and asymptotic minimax estimation over spheres of growing dimension*.
- [DL20] Xialiang Dou and Tengyuan Liang, *Training neural networks as learning data-adaptive kernels: Provable representation and approximation benefits*, Journal of the American Statistical Association (2020), 1–14.
- [DLL⁺19] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai, *Gradient descent finds global minima of deep neural networks*, International Conference on Machine Learning, PMLR, 2019, pp. 1675–1685.
- [DLS22] Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi, *Neural networks can learn representations with gradient descent*, Conference on Learning Theory, PMLR, 2022, pp. 5413–5452.
- [DM20] Amit Daniely and Eran Malach, *Learning parities with neural networks*, Advances in Neural Information Processing Systems **33** (2020).
- [DMHR⁺18] AGG De Matthews, J Hron, M Rowland, RE Turner, and Z Ghahramani, *Gaussian process behaviour in wide deep neural networks*, 6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings, 2018.
- [DW18] Edgar Dobriban and Stefan Wager, *High-dimensional asymptotics of prediction: Ridge regression and classification*, The Annals of Statistics **46** (2018), no. 1, 247–279.
- [DZPS18] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh, *Gradient descent provably optimizes over-parameterized neural networks*, International Conference on Learning Representations, 2018.
- [EF14] Costas Efthimiou and Christopher Frye, *Spherical harmonics in p dimensions*, World Scientific, 2014.
- [EK10] Noureddine El Karoui, *The spectrum of kernel random matrices*, The Annals of Statistics **38** (2010), no. 1, 1–50.
- [FDZ19] Cong Fang, Hanze Dong, and Tong Zhang, *Over parameterized two-level neural networks can learn near optimal feature representations*, arXiv:1910.11508 (2019).

- [FW20] Zhou Fan and Zhichao Wang, *Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks*, Advances in neural information processing systems **33** (2020), 7710–7721.
- [GLK⁺20] Federica Gerace, Bruno Loureiro, Florent Krzakala, Marc Mézard, and Lenka Zdeborová, *Generalisation error in learning with random features and the hidden manifold model*, International Conference on Machine Learning, PMLR, 2020, pp. 3452–3462.
- [GLR⁺22] Sebastian Goldt, Bruno Loureiro, Galen Reeves, Florent Krzakala, Marc Mézard, and Lenka Zdeborová, *The gaussian equivalence of generative models for learning with shallow neural networks*, Mathematical and Scientific Machine Learning, PMLR, 2022, pp. 426–471.
- [GLSS18a] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro, *Characterizing implicit bias in terms of optimization geometry*, International Conference on Machine Learning, PMLR, 2018, pp. 1832–1841.
- [GLSS18b] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro, *Implicit bias of gradient descent on linear convolutional networks*, Advances in neural information processing systems **31** (2018).
- [GMMM19] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari, *Limitations of lazy training of two-layers neural networks*, Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 9111–9121.
- [GMMM20] ———, *When do neural networks outperform kernel methods?*, Advances in Neural Information Processing Systems **33** (2020), 14820–14830.
- [GMMM21] ———, *Linearized two-layers neural networks in high dimension*, The Annals of Statistics **49** (2021), no. 2, 1029–1054.
- [GSJW20] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart, *Disentangling feature and lazy training in deep neural networks*, Journal of Statistical Mechanics: Theory and Experiment **2020** (2020), no. 11, 113301.
- [GWB⁺17] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro, *Implicit regularization in matrix factorization*, Advances in neural information processing systems **30** (2017).
- [HL22a] Hong Hu and Yue M Lu, *Sharp asymptotics of kernel ridge regression beyond the linear regime*, arXiv:2205.06798 (2022).
- [HL22b] ———, *Universality laws for high-dimensional learning with random features*, IEEE Transactions on Information Theory (2022).
- [HMRT22] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani, *Surprises in high-dimensional ridgeless least squares interpolation*, The Annals of Statistics **50** (2022), no. 2, 949–986.
- [HN19] Boris Hanin and Mihai Nica, *Finite depth and width corrections to the neural tangent kernel*, arXiv:1909.05989 (2019).

- [Hor91] Kurt Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural networks **4** (1991), no. 2, 251–257.
- [HY20] Jiaoyang Huang and Horng-Tzer Yau, *Dynamics of deep neural networks and neural tangent hierarchy*, International Conference on Machine Learning, PMLR, 2020, pp. 4542–4551.
- [JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, Advances in neural information processing systems, 2018, pp. 8571–8580.
- [JSS⁺20] Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clément Hongler, and Franck Gabriel, *Kernel alignment risk estimator: Risk prediction from training data*, Advances in Neural Information Processing Systems **33** (2020), 15568–15578.
- [JT18a] Ziwei Ji and Matus Telgarsky, *Gradient descent aligns the layers of deep linear networks*, arXiv:1810.02032 (2018).
- [JT18b] ———, *Risk and parameter convergence of logistic regression*, arXiv:1803.07300 (2018).
- [JT19] ———, *Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks*, International Conference on Learning Representations, 2019.
- [JT20] ———, *Directional convergence and alignment in deep learning*, Advances in Neural Information Processing Systems **33** (2020), 17176–17186.
- [Kac56] Mark Kac, *Foundations of kinetic theory*, Proceedings of The third Berkeley symposium on mathematical statistics and probability, vol. 3, 1956, pp. 171–197.
- [KH19] Kenji Kawaguchi and Jiaoyang Huang, *Gradient descent finds global minima for generalizable deep neural networks of practical sizes*, 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2019, pp. 92–99.
- [KY17] Antti Knowles and Jun Yin, *Anisotropic local laws for random matrices*, Probability Theory and Related Fields **169** (2017), 257–352.
- [KZSS21] Frederic Koehler, Lijia Zhou, Danica J Sutherland, and Nathan Srebro, *Uniform convergence of interpolators: Gaussian width, norm bounds and benign overfitting*, Advances in Neural Information Processing Systems **34** (2021), 20657–20668.
- [LBN⁺18] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein, *Deep neural networks as gaussian processes*, International Conference on Learning Representations, 2018.
- [LCM20] Zhenyu Liao, Romain Couillet, and Michael W Mahoney, *A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent*, Advances in Neural Information Processing Systems **33** (2020), 13939–13950.

- [LL18] Yuanzhi Li and Yingyu Liang, *Learning overparameterized neural networks via stochastic gradient descent on structured data*, Advances in Neural Information Processing Systems, 2018, pp. 8157–8166.
- [LL19] Kaifeng Lyu and Jian Li, *Gradient descent maximizes the margin of homogeneous neural networks*, arXiv:1906.05890 (2019).
- [LLL20] Zhiyuan Li, Yuping Luo, and Kaifeng Lyu, *Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning*, arXiv:2012.09839 (2020).
- [LLS21] Fanghui Liu, Zhenyu Liao, and Johan Suykens, *Kernel regression in high dimensions: Refined analysis beyond double descent*, International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 649–657.
- [LML⁺20] Yiping Lu, Chao Ma, Yulong Lu, Jianfeng Lu, and Lexing Ying, *A mean field analysis of deep resnet and beyond: Towards provably optimization via overparameterization from depth*, International Conference on Machine Learning, PMLR, 2020, pp. 6426–6436.
- [LR20] Tengyuan Liang and Alexander Rakhlin, *Just interpolate: Kernel “ridgeless” regression can generalize*, The Annals of Statistics **48** (2020), no. 3, 1329–1347.
- [LRZ20] Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai, *On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels*, Conference on Learning Theory, PMLR, 2020, pp. 2683–2711.
- [LS75] BF Logan and Larry A Shepp, *Optimal reconstruction of a function from its projections*, Duke Math. J. **42** (1975), no. 1, 645–659.
- [LWY⁺19] Zhiyuan Li, Ruosong Wang, Dingli Yu, Simon S Du, Wei Hu, Ruslan Salakhutdinov, and Sanjeev Arora, *Enhanced convolutional neural tangent kernels*, arXiv:1911.00809 (2019).
- [LXS⁺19] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington, *Wide neural networks of any depth evolve as linear models under gradient descent*, Advances in neural information processing systems **32** (2019), 8572–8583.
- [LZB20] Chaoyue Liu, Libin Zhu, and Mikhail Belkin, *On the linearity of large non-linear models: when and why the tangent kernel is constant*, Advances in Neural Information Processing Systems **33** (2020).
- [MBM18] Song Mei, Yu Bai, and Andrea Montanari, *The landscape of empirical risk for non-convex losses*, The Annals of Statistics **46** (2018), no. 6A, 2747–2774.
- [MHPG⁺22] Alireza Mousavi-Hosseini, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A Erdogdu, *Neural networks efficiently learn low-dimensional representations with sgd*, arXiv:2209.14863 (2022).

- [Mis22] Theodor Misiakiewicz, *Spectrum of inner-product kernel matrices in the polynomial regime and multiple descent phenomenon in kernel ridge regression*, arXiv:2204.10425 (2022).
- [MJ66] Henry P McKean Jr, *A class of markov processes associated with nonlinear parabolic equations*, Proceedings of the National Academy of Sciences **56** (1966), no. 6, 1907–1911.
- [MM19] Song Mei and Andrea Montanari, *The generalization error of random features regression: Precise asymptotics and the double descent curve*, Communications on Pure and Applied Mathematics (2019).
- [MM21] Theodor Misiakiewicz and Song Mei, *Learning with convolution and pooling operations in kernel methods*, arXiv:2111.08308 (2021).
- [MMM19] Song Mei, Theodor Misiakiewicz, and Andrea Montanari, *Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit*, Conference on Learning Theory, PMLR, 2019, pp. 2388–2464.
- [MMM21] ———, *Learning with invariances in random features and kernel models*, Conference on Learning Theory, PMLR, 2021, pp. 3351–3418.
- [MMM22] ———, *Generalization error of random feature and kernel methods: hypercontractivity and kernel matrix concentration*, Applied and Computational Harmonic Analysis **59** (2022), 3–84.
- [MMN18] Song Mei, Andrea Montanari, and Phan-Minh Nguyen, *A mean field view of the landscape of two-layer neural networks*, Proceedings of the National Academy of Sciences **115** (2018), no. 33, E7665–E7671.
- [MRSY19] Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan, *The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the over-parametrized regime*, arXiv:1911.01544 (2019).
- [MS22] Andrea Montanari and Basil N Saeed, *Universality of empirical risk minimization*, Conference on Learning Theory, PMLR, 2022, pp. 4310–4312.
- [MVSS20] Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai, *Harmless interpolation of noisy data in regression*, IEEE Journal on Selected Areas in Information Theory **1** (2020), no. 1, 67–83.
- [MZ22] Andrea Montanari and Yiqiao Zhong, *The interpolation phase transition in neural networks: Memorization and generalization under lazy training*, The Annals of Statistics **50** (2022), no. 5, 2816–2847.
- [NBMS17] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro, *Exploring generalization in deep learning*, Advances in neural information processing systems **30** (2017).

- [NCS19] Atsushi Nitanda, Geoffrey Chinot, and Taiji Suzuki, *Gradient descent can learn less over-parameterized two-layer neural networks on classification problems*, arXiv:1905.09870 (2019).
- [Nea95] Radford M Neal, *Bayesian learning for neural networks*, Ph.D. thesis, Citeseer, 1995.
- [NP20] Phan-Minh Nguyen and Huy Tuan Pham, *A rigorous framework for the mean field limit of multilayer neural networks*, arXiv:2001.11443 (2020).
- [NTS14] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro, *In search of the real inductive bias: On the role of implicit regularization in deep learning*, arXiv:1412.6614 (2014).
- [NXB⁺18] Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-dickstein, *Bayesian deep convolutional networks with many channels are gaussian processes*, International Conference on Learning Representations, 2018.
- [OS19] Samet Oymak and Mahdi Soltanolkotabi, *Overparameterized nonlinear learning: Gradient descent takes the shortest path?*, International Conference on Machine Learning, PMLR, 2019, pp. 4951–4960.
- [OS20] ———, *Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks*, IEEE Journal on Selected Areas in Information Theory **1** (2020), no. 1, 84–105.
- [Pin99] Allan Pinkus, *Approximation theory of the mlp model in neural networks*, Acta numerica **8** (1999), 143–195.
- [RGKZ21] Maria Refinetti, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborová, *Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed*, arXiv:2102.11742 (2021).
- [RMR21] Dominic Richards, Jaouad Mourtada, and Lorenzo Rosasco, *Asymptotics of ridge (less) regression under general source condition*, International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 3889–3897.
- [RR07] Ali Rahimi and Benjamin Recht, *Random features for large-scale kernel machines*, Advances in neural information processing systems **20** (2007).
- [RR08] ———, *Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning*, Advances in neural information processing systems **21** (2008).
- [RR17] Alessandro Rudi and Lorenzo Rosasco, *Generalization properties of learning with random features.*, NIPS, 2017, pp. 3215–3225.
- [RVE18] Grant M Rotskoff and Eric Vanden-Eijnden, *Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error*, stat **1050** (2018), 22.

- [RZ19] Alexander Rakhlin and Xiyu Zhai, *Consistency of interpolation with laplace kernels is a high-dimensional phenomenon*, Conference on Learning Theory, PMLR, 2019, pp. 2595–2623.
- [SFG⁺20] Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Jonathan Ragan-Kelley, Ludwig Schmidt, and Benjamin Recht, *Neural kernels without tangents*, International Conference on Machine Learning, PMLR, 2020, pp. 8614–8623.
- [Sha22] Ohad Shamir, *The implicit bias of benign overfitting*, Conference on Learning Theory, PMLR, 2022, pp. 448–478.
- [SHN⁺18] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro, *The implicit bias of gradient descent on separable data*, The Journal of Machine Learning Research **19** (2018), no. 1, 2822–2878.
- [SS20] Justin Sirignano and Konstantinos Spiliopoulos, *Mean field analysis of neural networks: A central limit theorem*, Stochastic Processes and their Applications **130** (2020), no. 3, 1820–1852.
- [SY19] Zhao Song and Xin Yang, *Quadratic suffices for over-parametrization via matrix chernoff bound*, arXiv:1906.03593 (2019).
- [Sze39] Gabor Szeg, *Orthogonal polynomials*, vol. 23, American Mathematical Soc., 1939.
- [TB20] Alexander Tsigler and Peter L Bartlett, *Benign overfitting in ridge regression*, arXiv:2009.14286 (2020).
- [Ver18] Roman Vershynin, *High-dimensional probability: An introduction with applications in data science*, vol. 47, Cambridge university press, 2018.
- [VW19] Santosh Vempala and John Wilmes, *Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds*, Conference on Learning Theory, PMLR, 2019, pp. 3115–3117.
- [Wai19] Martin J Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48, Cambridge university press, 2019.
- [WGL⁺19] Blake Woodworth, Suriya Genesekar, Jason Lee, Daniel Soudry, and Nathan Srebro, *Kernel and deep regimes in overparametrized models*, Conference on Learning Theory (COLT), 2019.
- [WGL⁺20] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro, *Kernel and rich regimes in overparametrized models*, Conference on Learning Theory, PMLR, 2020, pp. 3635–3673.
- [Wil96] Christopher Williams, *Computing with infinite networks*, Advances in neural information processing systems **9** (1996).
- [WX20] Denny Wu and Ji Xu, *On the optimal weighted ℓ_2 regularization in overparameterized linear regression*, Advances in Neural Information Processing Systems **33** (2020), 10112–10123.

- [XHM⁺22] Lechao Xiao, Hong Hu, Theodor Misiakiewicz, Yue Lu, and Jeffrey Pennington, *Precise learning curves and higher-order scalings for dot-product kernel regression*, Advances in Neural Information Processing Systems **35** (2022), 4558–4570.
- [Xia22] Lechao Xiao, *Eigenspace restructuring: a principle of space and frequency in neural networks*, Conference on Learning Theory, PMLR, 2022, pp. 4888–4944.
- [YH20] Greg Yang and Edward J Hu, *Feature learning in infinite-width neural networks*, arXiv:2011.14522 (2020).
- [YKM20] Chulhee Yun, Shankar Krishnan, and Hossein Mobahi, *A unifying view on implicit bias in training linear neural networks*, arXiv:2010.02501 (2020).
- [YS19] Gilad Yehudai and Ohad Shamir, *On the power and limitations of random features for understanding neural networks*, Advances in Neural Information Processing Systems **32** (2019), 6598–6608.
- [ZBH⁺21] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, *Understanding deep learning (still) requires rethinking generalization*, Communications of the ACM **64** (2021), no. 3, 107–115.
- [ZCZG20] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu, *Gradient descent optimizes over-parameterized deep relu networks*, Machine Learning **109** (2020), no. 3, 467–492.
- [ZG19] Difan Zou and Quanquan Gu, *An improved analysis of training over-parameterized deep neural networks*, arXiv:1906.04688 (2019).