Topics on the mathematics of deep learning

Bruno Loureiro
Département d'Informatique, École Normale Supérieure - PSL & CNRS, France
bruno.loureiro@di.ens.fr

Lecture notes for the course "*Mathematics of deep learning*" taught at the IASD/MASH master program of the Université Paris Sciences-Lettres (PSL) between January 10 - February 28, 2025.

Contents

Notation

We denote the cardinality of a set A as $|A| \coloneqq \mathrm{card}(A)$, and denote finite discrete sets as $[n] \coloneqq \{1, \dots, n\}$. We denote vectors by bold lower letters $\boldsymbol{v} \in \mathbb{R}$ and matrices by bold capital letters $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, and their elements by v_i and A_{ij}. The spectrum of a symmetric square matrix $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ as $\mathrm{spec}(\boldsymbol{M}) \coloneqq \{\lambda_1, \dots, \lambda_d\} \subset \mathbb{R}$ which we always assume is ordered $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. We denote by $||\boldsymbol{A}||_{\mathtt{op}}$, $||\boldsymbol{A}||_*$ and $||\boldsymbol{A}||_{\mathtt{F}}$ the operator, nuclear and Frobenius norm, respectively.

Chapter 1

Introduction to empirical risk minimisation

Introduction & Motivation

AI-based technology is increasingly present in day-a-day life. Tasks such as talking to your phone or asking a LLM to translate a text were unimaginable a decade ago. The backbone of these developments is *machine learning*, the field concerned with how computers learn from data. In particular, one class of machine learning models, *deep neural networks*, has been a driving force behind these developments.

Despite the major engineering breakthroughs achieved by neural networks, it is fair to say our current mathematical understanding of their working remains, to say the least, poor. Indeed, current machine learning practice defies several aspects of the intuition built from classical statistical analysis, presenting us with several interesting mathematical challenges. While this "gap with the practice" is frustrating to some, the mathematically minded student should see these challenges as a fertile ground with plenty of opportunity. Open problems are the fuel of research.

Curiously, an important part of the progress made in the past few years came from the understanding that the "exotic" properties of neural networks observed by practitioners are shared by simpler architectures that are amenable to a mathematical analysis. This observation will be the guiding thread of these lectures. Our goal will be to motivate and investigate these research questions in the simplest context where we can make sense of them.

⚠ A word of caution: as any research-level course, there is no consensus of what "Mathematics of deep learning" actually means. Therefore, it should go without saying that this course is an obviously biased selection of topics. It is not intended to be exhaustive, and probably a more appropriate name for this course would be "Some mathematics of deep learning".

1.1 Supervised learning

Machine Learning is the subject concerned with the problem of how statistical models learn patterns from data. In practice, this can mean different things depending on the nature of the data and the nature of the information one aims to extract from it. Three of the most common tasks in machine learning are *supervised learning*, *unsupervised learning* and *reinforcement learning*. In the following, we will be focusing in supervised learning.

In supervised learning, data is given in pairs $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i \in [n]\}$ (often refereed to as the *training data*), where $x_i \in \mathcal{X}$ are known as the *covariates* or *input data* and $y_i \in \mathcal{Y}$ are known as the *response* or *labels*. In particular, when $\mathcal{Y} = \mathbb{R}$, we say we have a *regression task*, while if $\mathcal{Y}$ is a discrete set, e.g. $\mathcal{Y} = \{1, \cdots, K\}$ we say we have a *classification task*. The main goal of supervised learning is to come up with a function $f : \mathcal{X} \to \mathcal{Y}$ that given a covariate, predicts its label "as well as

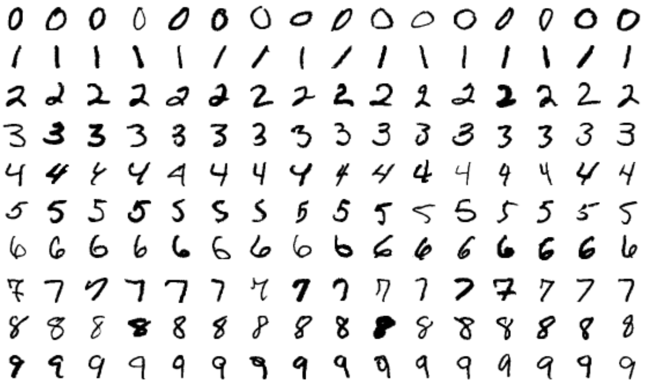

Figure 1.1: Example of different samples $x_i \in \mathbb{R}^{28\times 28}$ from the MNIST data set.

possible". A few comments are in order.

- Often, we will assume the covariates are vectorised $\mathcal{X} \subset \mathbb{R}^d$. In "real life", data is encoded in a data structure, for example an Excel table, a `pandas.DataFrame` or an image given by a `numpy.ndarray` with pixels and RGB values). Nevertheless, it is common practice to *vectorise* or *flatten* in a big vector. Studying how the encoding might affect prediction is an interesting topic, although outside of the scope of these lectures.

- In a K-class classification task, the image of the predictor $f : \mathcal{X} \to [K]$ is a discrete set. This is not amenable to doing optimisation (for instance defining a derivative). Therefore, in practice it is common to decompose $f = d \circ \boldsymbol{g}$ with $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^K$, known as the *score*, and $d : \mathbb{R}^K \to [k]$, known as the *decoder*.

- When $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}$, it is common to define the *covariate matrix* $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ and the *response vector* $\boldsymbol{y} \in \mathbb{R}$ by stacking the covariates x_i and the responses y_i row-wise.

Example 1 (Supervised learning tasks)**.** A few practical examples of supervised learning are:

- **Image classification:** The covariates x_i represent images and the labels y_i encode the respective classes. A very popular example is the MNIST data set, composed of $n = 70000$ images of black and white hand written digits. In this case, the covariates are given in terms of a matrix $\boldsymbol{x}_i \in \mathbb{R}^{28\times 28}$ with entries containing the grayscale level of each pixels, and $y_i \in \{0, \dots, 9\}$ label the respective digit. See Figure 1.1 for an illustration.

- **Price estimation:** Consider the problem of estimating the value of a house. In this case, the covariates can be a table with different relevant information (surface, number of rooms, location, electric efficiency, date of construction, etc.) and $y_i \in \mathbb{R}$ its price.

- **Text generation:** In this task, we are given a sentence (i.e. a sequence of words) and the goal is to predict the next word ("text completion"). In this case, the covariates are given by a vector of words, and the responses are the next word in the sentence. This task is at the heart of *Large Language Models* (LLMs) such as GPT, and is also known as *self-supervised learning* since the supervised data is typically obtained by taking full sentences and masking words.

The key word in the definition given above is "to learn as well as possible". There are two subtle aspects in this sentence: what does "learning" and "as well as possible" means.

Learning vs. memorisation — To see this, consider the following predictor:

$$f(x) = \begin{cases} y_i & \text{if } x = x_i \text{ for some } (x_i, y_i) \in \mathcal{D} \\ 0 & \text{otherwise} \end{cases} \tag{1.1.1}$$

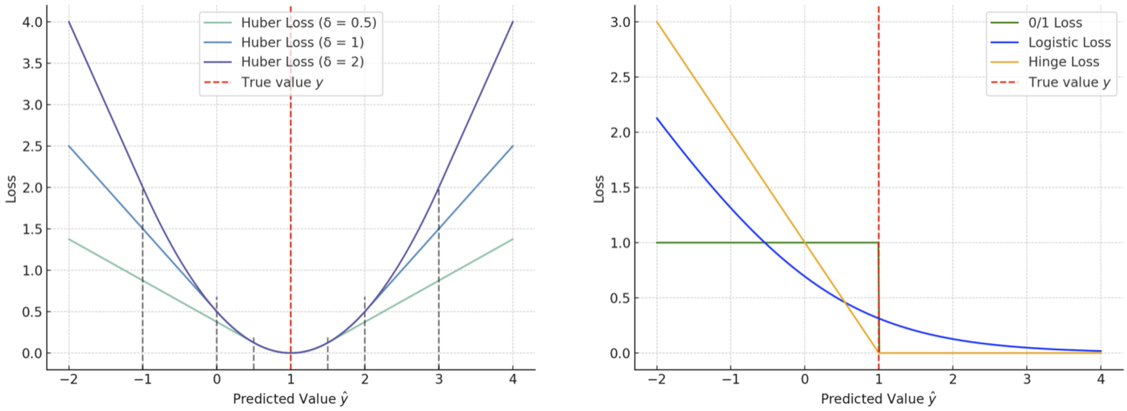

Figure 1.2: **(Left)** Huber loss. **(Right)** Classification losses.

It is clear that this function perfectly assigns the correct response to all covariates which are in the training set. However, it will very likely predict the wrong response for covariates which are not in the training data. Indeed, any reasonable definition of "learning" will imply that this predictor has not learned anything from the data, but rather *memorised* the training data.

To make the notion of "learning" precise, it is common to adopt a probabilistic point of view. More precisely, we assume that the training data $(x_i, y_i)_{i\in[n]}$ has been independently[1] drawn from a joint probability distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$. With this assumption, we say that a predictor $f : \mathcal{X} \to \mathcal{Y}$ has *learned* is it is able to "predict well" the response of covariates which are outside the training data, i.e. that for a typical new sample $(x_{\text{new}}, y_{\text{new}}) \sim p$, $f(x_{\text{new}})$ is "close" to y_{new}. Now, it remains making precise what "close" means.

1.1.1 Loss and risk: or how to measure "good" learning

To quantify how "good" a predictor is, we introduce a notion of "distance"[2] between the predicted response $f(x)$ and the true response y, known as a *loss function* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$. From that, it is natural to define the following two notions:

Definition 1 (Risks)**.** Let $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ denote training data drawn from p, and let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ denote a loss function. For any predictor $f : \mathcal{X} \to \mathcal{Y}$ we define the *empirical risk*, also known as the *training error*:

$$\hat{R}(f; \mathcal{D}) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)). \tag{1.1.2}$$

Similarly, we define the *population risk*, also known as *test* or *generalisation error*:

$$R(f) \coloneqq \mathbb{E}\left[\ell(y, f(x))\right]. \tag{1.1.3}$$

⚠ Note that while $\hat{R}(f; \mathcal{D})$ is a random function of f (since x_i, y_i are random variables), $R(f)$ is a deterministic function of f.

Example 2 (Loss functions for regression)**.** Consider a regression task $\mathcal{Y} = \mathbb{R}$ with vectorised data $\mathcal{X} = \mathbb{R}^d$.

[1] Although sample-wise independence is a common assumption, it is good to keep in mind it is not always the case in practice, where there might be sampling biases.

[2] Note the loss function is not a distance in the mathematical sense.

- **Square loss:** The widely used loss function for regression problems is the square loss:

$$\ell(y, f(\boldsymbol{x})) = \frac{1}{2}(y - f(\boldsymbol{x}))^2 \tag{1.1.4}$$

- **Absolute deviation:** The square loss is particularly sensitive to outliers in the data. An alternative loss function for regression which is less sensitive to outliers is the absolute deviation loss:

$$\ell(y, f(\boldsymbol{x})) = |y - f(\boldsymbol{x})| \tag{1.1.5}$$

- **Huber loss:** The Huber loss combines the squared and absolute deviation losses:

$$\ell_\delta(y, f(\boldsymbol{x})) = \begin{cases} \frac{1}{2}(y - f(\boldsymbol{x}))^2 & \text{for } |y - f(\boldsymbol{x})| \leq \delta \\ \delta\left(|y - f(\boldsymbol{x})| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases} \tag{1.1.6}$$

  Note that the Huber loss has an hyperparameter $\delta \geq 0$ that needs to be fixed. Figure 1.2 (left) shows the Huber loss for different values of location parameter δ.

Example 3 (Loss functions for classification)**.** Consider a K-class classification task $\mathcal{Y} = [K]$ with vectorised data $\mathcal{X} = \mathbb{R}^d$. Ideally, we would like to minimise the number of misclassified examples in expectation. This corresponds to minimising the so called **0/1 loss**:

$$\ell(y, \boldsymbol{f}(\boldsymbol{x})) = \mathbf{1}\left(y \neq f(\boldsymbol{x})\right) \tag{1.1.7}$$

where $f : \mathcal{X} \to [K]$ is a classifier. The predictor f is discrete, and the 0/1 loss function is non-convex and nowhere differentiable. As discussed in section 1.1, the solution to the first observation is to define $f = d \circ \boldsymbol{g}$ with $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^K$ and d a decoder. The solution to the second consists of defining a **surrogate loss** function on g which is convex. The most popular example is the **cross-entropy loss**:

$$\ell(\boldsymbol{y}, \boldsymbol{g}(\boldsymbol{x})) = -\sum_{k=1}^{K} y_k \log \frac{e^{y_k g(\boldsymbol{x})_k}}{\sum_{k'=1}^{K} e^{y_{k'} g(\boldsymbol{x})_{k'}}} \tag{1.1.8}$$

where the labels are one-hot encoded.[3] A particular case which is often studied in the theoretical literature is binary classification ($K = 2$). In this case, it is common to encode the labels as Radamacher variables $y \in \{-1, +1\}$ (and hence $f(\boldsymbol{x}) = \text{sign}(g(\boldsymbol{x}))$). The most popular loss functions for binary classification are the **logistic loss**:

$$\ell(y, g(\boldsymbol{x})) = \log\left(1 + e^{-yg(\boldsymbol{x})}\right), \tag{1.1.9}$$

which is just a particular case of the cross-entropy loss, and the **hinge loss**:

$$\ell(y, g(\boldsymbol{x})) = \max(0, 1 - yg(\boldsymbol{x})). \tag{1.1.10}$$

Note that these two examples are only a function of the *margin* $t = yg(\boldsymbol{x})$. Figure 1.2 (right) compares the logistic and hinge losses with the 0/1 loss defined in eq. (1.1.7), which when written in terms of the score function $g(\boldsymbol{x})$ in binary classification is given by a Heaviside step function $\ell(y, g(\boldsymbol{x})) = \Theta(-yg(\boldsymbol{x}))$.

[3] The one-hot encoding consists of embedding the labels $y \in \{1, \ldots, K\}$ into vectors $\boldsymbol{y} \in \{0, 1\}^K$ with components $y_k = 1$ if $y = k$ and $y_k = 0$ otherwise.

1.1.2 Bayes risk

Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, what is the best achievable risk? Note that the population risk can be decomposed as:

$$R(f) = \mathbb{E}[\ell(Y, f(X))] = \mathbb{E}_x[\mathbb{E}[\ell(Y, f(x))|X = x]] \tag{1.1.11}$$

where we used the probabilist notation of Y, X to denote the random variables, as opposed to the values they can take. This defines the conditional risk:

$$r(z|x) = \mathbb{E}[\ell(Y, z)|X = x] \tag{1.1.12}$$

which is a deterministic function of both $x \in \mathcal{X}$ and $z \in \mathcal{Y}$.

Proposition 1 (Bayes predictor)**.** The population risk is minimised by the following *Bayes predictor* $f_\star : \mathcal{X} \to \mathcal{Y}$:

$$f_\star(x) \in \underset{z\in\mathcal{Y}}{\arg\min}\ \mathbb{E}[\ell(Y, z)|X = x] = \underset{z\in\mathcal{Y}}{\arg\min}\ r(z|x) \tag{1.1.13}$$

Note that although the global minimiser might not be unique, the risk associated at the Bayes predictor, known as the *Bayes risk*, is unique:

$$R_\star = \mathbb{E}_x\left[\inf_{z\in\mathcal{Y}} \mathbb{E}_Y[\ell(Y, z)|X = x]\right] \tag{1.1.14}$$

Given a loss function and a data distribution, the best predictor is the Bayes predictor $f_\star$, also known as *target function*. Typically, the Bayes risk is non-zero due to the noise in the labels. Of course, computing the Bayes predictor in practice is intractable, as typically we don't have access to the data distribution. Nevertheless, it provides the theoretical "gold standard" for assessing the methods we will study next.

Example 4 (Bayes predictor for the square loss)**.** Consider a regression problem $\mathcal{Y} = \mathbb{R}$ with the square loss $\ell(y, z) = (y - z)^2$. By definition, the Bayes predictor is given by:

$$f_\star(x) \in \underset{z\in\mathbb{R}}{\arg\min}\ r(z|x) \tag{1.1.15}$$

with $r(z|x) = \mathbb{E}[(Y - z)^2|X = x]$. Since this is a differentiable function of $z \in \mathbb{R}$, minimisers are critical points:

$$\partial_z r(z|x) = -2\mathbb{E}[Y - z|X = x] \overset{!}{=} 0 \tag{1.1.16}$$

This has a unique solution, given by the conditional expectation over the likelihood $p(y|x)$:

$$f_\star(x) = \mathbb{E}[Y|X = x] \tag{1.1.17}$$

Example 5 (Bayes predictor for the 0/1 loss)**.** Consider a binary classification problem $\mathcal{Y} = \{-1, +1\}$ with the 0/1 loss $\ell(y, z) = \mathbf{1}(y \neq z)$. By definition, the Bayes predictor is given by:

$$f_\star(x) \in \underset{z\in\{-1,+1\}}{\arg\min}\ r(z|x) \tag{1.1.18}$$

with $r(z|x) = \mathbb{E}[\mathbf{1}(Y \neq z)|X = x] = \mathbb{P}(Y \neq z|X = x)$. Denoting $g(x) = \mathbb{P}(Y = 1|X = x)$, we have:

$$f_\star(x) = \text{sign}(2g(x) - 1) = \begin{cases} +1 & \text{if } g(x) \geq \frac{1}{2} \\ -1 & \text{if } g(x) < \frac{1}{2} \end{cases} \tag{1.1.19}$$

Note that what happens exactly at $g(x) = 1/2$ is irrelevant as the Bayes risk is the same:

$$R_\star = \mathbb{E}_X[\min\{g(X), 1 - g(X)\}] \tag{1.1.20}$$

1.2 Empirical risk minimisation

Now that we made sense of what learning is, and we have introduced a way of assessing how good a predictor is, it remains to establish a systematic procedure for how to find good predictors. In other words, an *algorithm* for learning.[4] While different supervised learning algorithms exist - such as decision trees and local averaging methods - our focus in the following will be on *empirical risk minimisation*, which is arguably one of the most popular supervised learning frameworks.

Consider a supervised learning problem with training data $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i \in [n]\}$ independently from a distribution p. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ denote a loss function. As we discussed above, ideally we would like to find a predictor which has small population risk. A natural idea for finding a predictor would be to minimise the population risk. However, in supervised learning we don't have access to the data distribution, only to a finite number of samples from it - the training data $\mathcal{D}$. Therefore, we cannot minimise the population risk directly. The main idea of empirical risk minimisation is minimise instead the empirical risk:

$$\min_{f \in \mathcal{H}} \hat{R}(f; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)). \tag{1.2.1}$$

Note that minimising over the set of all functions $f : \mathcal{X} \to \mathcal{Y}$ is an intractable computational problem, and therefore one should restrict the minimisation problem in eq. (1.2.1) to a subset $f \in \mathcal{H}$, also known as the *hypothesis class*. Most commonly, we further restrict the minimisation to parametric function classes, i.e. hypotheses $\mathcal{H} = \{f(\cdot; \theta) : \mathcal{X} \to \mathcal{Y} : \theta \in \Theta\}$. In this case, the empirical risk can be thought as a function of the parameters $\theta \in \mathcal{D}$ instead: $\hat{R}(f; \mathcal{D}) = \hat{R}(\theta; \mathcal{D})$[5].

⚠ The choice of a hypothesis class $\mathcal{H}$ (parametric or not) introduces a bias in learning, often refereed as the *inductive bias*. Indeed, the choice of $\mathcal{H}$ reflects a belief of the statistician on how the data likelihood $p(y|x)$ looks like. For example, a choice of linear function reflects a belief that the response is linearly correlated to the covariates.

Below, we give a few examples of popular parametric hypotheses.

Example 6 (Hypotheses classes)**.** To illustrate, consider a regression task with $\mathcal{Y} = \mathbb{R}$.

- **Linear functions:** Let $\boldsymbol{\varphi} : \mathcal{X} \to \mathbb{R}^p$ denote a vector-valued function, also known as a *feature map*. The hypothesis of linear functions is defined by:

$$\mathcal{H} = \{f(x; \boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \boldsymbol{\varphi}(x) \rangle : \boldsymbol{\theta} \in \mathbb{R}^p\} \tag{1.2.2}$$

- **Generalised linear models:** Let $\boldsymbol{\varphi} : \mathcal{X} \to \mathbb{R}^p$ denote a feature map and $g : \mathbb{R} \to \mathbb{R}$ a real-valued function. The hypothesis of generalised linear models is defined by

$$\mathcal{H} = \{f(x; \boldsymbol{\theta}) = g\left(\langle \boldsymbol{\theta}, \boldsymbol{\varphi}(x) \rangle\right) : \boldsymbol{\theta} \in \mathbb{R}^p\} \tag{1.2.3}$$

- **Two-layer neural networks:** Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a real-valued function and consider vectorised covariates $\mathcal{X} = \mathbb{R}^d$. The hypothesis of fully-connected two-layer neural networks is defined by:

$$\mathcal{H} = \left\{ f(x; \boldsymbol{a}, \boldsymbol{W}) = \sum_{k=1}^{p} a_k \sigma\left(\langle \boldsymbol{w}_k, \boldsymbol{x} \rangle\right) : \boldsymbol{a} \in \mathbb{R}^p, \boldsymbol{W} \in \mathbb{R}^{p \times d} \right\} \tag{1.2.4}$$

  Note that that on a high-level, this can be seen as a linear model with a parametric feature map $\boldsymbol{\varphi}(\boldsymbol{x}; \boldsymbol{W}) = \sigma(\boldsymbol{W}\boldsymbol{x})$ which itself adapts to the data via empirical risk minimisation.

[4] Technically, and algorithm $\mathcal{A}$ can be defined as a map that takes the data $\mathcal{D}$ and returns a hypothesis $\hat{f}(\mathcal{D}) \in \mathcal{H}$ in the chosen hypothesis class.

[5] Examples of non-parametric function classes include kernel methods, which we will see later.

Therefore, empirical risk minimisation maps the problem of learning to an optimisation problem over a (random) objective function $F(\theta) := \hat{R}(\theta;\mathcal{D})$. Except for a few examples, such as linear functions, this objective function is not convex, and therefore it might have several minimisers $\hat{\theta}(\mathcal{D}) \in \arg\min\ \hat{R}(\theta;\mathcal{D})$.

⚠ The minimiser $\hat{\theta}(\mathcal{D}) \in \arg\min\ \hat{R}(\theta;\mathcal{D})$ is itself random, since it is a function of the training data. Therefore, although the test error $R(\hat{\theta})$ is a deterministic function conditioned on the minimiser $\hat{\theta}(\mathcal{D})$, it is a random function of the training data $\mathcal{D}$.

A good choice of hypothesis? — At this point, one might ask why not considering a very expressive hypothesis class $\mathcal{H}$. The reason is two-fold. First, there is a computational reason: the more parameters there are, the more parameters we have to optimise - making the optimisation computationally more demanding. Second, a statistical reason: the training data is typically noisy, meaning that the richer our hypothesis, the more likely we will fit the noise in the data (also known as *overfitting*), which might hurt the test error. Traditional wisdom suggests that a good choice for $\mathcal{H}$ is a class of functions which is neither too simple nor too complicated. It is important to keep in mind, however, that our current practice of deep learning, where neural networks with billions of parameters are both successfully optimised and achieve good test performances defy this traditional wisdom.

1.3 Optimising the risk: descent algorithms

Empirical risk minimisation maps the learning problem to an optimisation problem:

$$\min_{\theta\in\Theta} \hat{R}(\theta;\mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i;\theta)). \tag{1.3.1}$$

Since the empirical risk above is generically a non-convex function, the choice of optimisation algorithm is important. In particular, two different algorithms can lead to two different minimisers which might have different generalisation properties. We now discuss some of the most popular optimisation algorithms used for machine learning, known as *descent-based methods*.

1.3.1 Gradient descent

Perhaps the most natural algorithm for optimisation is *gradient descent* (GD):

$$\begin{aligned} \theta_{k+1} &= \theta_k - \eta_k \nabla_\theta \hat{R}(\theta_k;\mathcal{D}) \\ &= \theta_k - \eta_k \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \ell(y_i, f(x_i;\theta_k)) \end{aligned} \tag{GD}$$

which consists of simply updating the weights in the steepest descent direction, with each step scaled by the *learning rate* $\eta_k > 0$. Note that GD naturally stops at a point in which $\nabla_\theta \hat{R} = 0$, which can be both a local or global minima. Defining a continuous function $\theta(\eta_k k) = \theta_k$ by piecewise affine interpolation, when the step size is small $\eta_k \to 0^+$, GD is well approximated by a continous *gradient flow*:

$$\dot{\theta}(t) = -\nabla_\theta \hat{R}(\theta(t);\mathcal{D}). \tag{1.3.2}$$

where $\dot{\theta} := {}^{\mathrm{d}\theta}/_{\mathrm{d}t}$. Or, seeing things in the opposite way, GD can be seen as the Euler discretisation of gradient flow with $t = k\eta_k$.

1.3.2 Stochastic gradient descent

A drawback of GD is that at every step k, one needs to compute the full gradient over the empirical risk. This means running over the full training set at every time - which can be slow if n is large. A simple way to avoid this computational bottleneck is to estimate the gradient at each step k only on a subset $b_k \subset [n]$ (known as a *mini-batch*) of the training data, which gives *stochastic gradient descent* (SGD):

$$\begin{aligned} \theta_{k+1} &= \theta_k - \eta_k \nabla_\theta \hat{R}(\theta; b_k). \\ &= \theta_k - \eta_k \frac{1}{|b_k|} \sum_{i \in b_k} \nabla_\theta \ell(y_i, f(x_i; \theta_k)). \end{aligned} \tag{1.3.3}$$

Together with its variants, SGD is one of the most used algorithm in modern machine learning.

⚠ Note that the choice of mini-batch $b_k \subset [n]$ plays a crucial role in the algorithm. Popular choices are: (a) sampling the mini-batches uniformly from $[n]$ with replacement or (b) partitioning $[n]$ over K disjoint subsets and going through the data in order. In this case, each full-pass over the data is known as an *epoch*.

Besides being computationally more efficient than GD, one advantage of SGD is that when the mini-batches b_k are chosen independently and without replacement[6], a single epoch of SGD it can be seen as an approximation for gradient flow on the population risk:

$$\dot{\theta}(t) = -\nabla_\theta R(\theta(t)). \tag{1.3.4}$$

Indeed, at each $k > 0$ the gradient $\nabla_\theta \hat{R}(\theta_k; b_k)$ is an unbiased estimate of the population gradient $\nabla_\theta R(\theta_k)$ (Robbins and Monro, 1951). This limit is known as *one-pass SGD*[7], and is mostly often studied in the particular case of $|b_k| = 1$. Note that this is not the case for GD or SGD with replacement, since seing the same data point (x_i, y_i) implies that the gradients will be biased.

Although the one-pass setting might seem unrealistic on a first sight, it is worth noting that it is a good approximation to certain scenarios, such as Large Language Models (LLM) like ChatGPT-3 which are trained of billions of tokens, see e.g. Table 2.2 in Brown et al. (2020).

Remark 1. In one-pass SGD with $|b_k| = 1$, each step corresponds to seeing one sample (x_i, y_i), and therefore the amount of data required to achieve a given error is equal to the number of SGD steps - or in other words: *convergence rates are equivalent to the sample complexity.*

With the observation above in mind, an useful way of thinking about one-pass SGD is as a noisy version of gradient descent on the population risk:

$$\theta_{k+1} = \theta_k - \eta_k \nabla_\theta R(\theta_k) + \eta_k \varepsilon_k \tag{1.3.5}$$

where we defined the effective noise:

$$\begin{aligned} \varepsilon_k &:= \frac{1}{|b_k|} \sum_{i \in b_k} \nabla_\theta \ell\left(y_i, f(x_i; \theta_k)\right) - \nabla_\theta R(\theta_k) \\ &= \frac{1}{|b_k|} \sum_{i \in b_k} \nabla_\theta \ell\left(y_i, f(x_i; \theta_k)\right) - \mathbb{E}\left[\nabla_\theta \ell(y, f(x; \theta_k))\right] \end{aligned} \tag{1.3.6}$$

which has zero mean since the estimation is unbiased[8]. This observation can be surprising at first sight: on average one-pass SGD optimises the true population risk even though this is an unknown

[6] When enough data is available $n \gg |b_k|$, this can be achieved by partitioning the training data in disjoint batches and running a single epoch of SGD

[7] Sometimes also refereed to online SGD, specially in the Statistical Physics literature.

[8] Note that since a fresh batch b_k is drawn at every step k, θ_k is independent of (x_i, y_i) for $i \in b_k$.

function! Therefore, understanding one-pass SGD essentially amounts to understanding two things: (a) gradient descent on the population risk, and (b) the properties of the effective noise ε_k. It is important to stress, however, that ε_k is a not a simple Gaussian noise, and therefore as a stochastic process *SGD can be very different from Brownian motion.*

1.4 Risk decompositions

We now have introduced the three key components involved in supervised learning:

- **The data**: The training data $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i \in [n]\}$, assumed to be drawn i.i.d. from a distribution $p(x, y)$.
- **The architecture**: The (typically parametric) class of functions $\mathcal{H} = \{f(\cdot; \theta) : \mathcal{X} \to \mathcal{Y} : \theta \in \Theta\}$ we choose to fit the data, also known as hypothesis class.
- **The algorithm**: The practical procedure we use to choose a particular function/hypothesis $\hat{f} \in \mathcal{H}$ using the training data $\mathcal{D}$, often by minimising the empirical risk (ERM) using a descent-based algorithm such as SGD.

Successful learning crucially depends on these three factors, and therefore it is envisageable to understand how each one of them impact the generalisation error. This motivates us to decompose the excess as a function of these different components. For concreteness, consider a supervised learning problem with training data $\mathcal{D}$, and consider empirical risk minimisation with a given algorithm (e.g. SGD). Let $\hat{\theta}(\mathcal{D})$ denote the output of the training algorithm. As argued in Section 1.1, understanding generalisation amounts to understanding the excess risk $R(\hat{\theta}) - R_\star$.[9]

A first coarse decomposition consists of separating the excess risk in an *estimation* and an *approximation* component:

$$R(\hat{\theta}) - R_\star = \underbrace{\left\{R(\hat{\theta}) - \inf_{\theta \in \Theta} R(\theta)\right\}}_{\text{estimation}} + \underbrace{\left\{\inf_{\theta \in \Theta} R(\theta) - R_\star\right\}}_{\text{approximation}}. \tag{1.4.1}$$

The estimation error quantifies how far the empirical risk minimiser is from the best possible predictor in the class of functions chosen by the statistician.[10] It depends on the training data, architecture and algorithm, and typically decreases with the quantity of data available. Instead, the approximation error quantifies how far the best predictor in the class is from the Bayes predictor, and hence is a deterministic quantity that only depends on the choice of architecture and on the underlying data distribution. It is independent of the amount of data available, and quantifies a fundamental limitation of approximating the Bayes predictor with a given choice of hypothesis.

The estimation error mixes both statistical and algorithmic aspects. Letting $\bar{\theta} \in \arg\min_{\theta \in \Theta} R(\theta)$, it can be useful to define a finer decomposition:

$$R(\hat{\theta}) - R(\bar{\theta}) = \underbrace{\left\{R(\hat{\theta}) - \hat{R}(\hat{\theta}; \mathcal{D})\right\} - \left\{R(\bar{\theta}) - \hat{R}(\bar{\theta}; \mathcal{D})\right\}}_{\text{statistical}} + \underbrace{\left\{\hat{R}(\hat{\theta}; \mathcal{D}) - \hat{R}(\bar{\theta}; \mathcal{D})\right\}}_{\text{algorithmic}} \tag{1.4.2}$$

The first two terms are statistical terms: they quantify how far minimising the empirical risk is from minimising the population risk. In other words, how much generalisation we loose by empirically estimating the risk, both for the $\hat{\theta}$ and $\bar{\theta}$. Sometimes these terms are also called the the *generalisation gap.* The third terms quantifies how successful the optimisation procedure, by comparing the empirical risk of the algorithm to the empirical risk of the best predictor in the class.

[9] Note this is a random quantity, so one often considers its expectation value over the draw of $\mathcal{D}$ or aims at a high-probability statement.

[10] In general, this can be outside of the class, hence the infimum.

1.5 Central challenges of supervised learning

In the previous sections, we introduced the main ingredients that define a supervised learning problem: the data distribution p, the hypothesis class $\mathcal{H}$ and the most popular training algorithms: GD and SGD. It is important stressing these three ingredients - data, hypothesis and algorithm - are indissociable. For instance, the hypothesis translates a belief of the statistician about the nature of the data likelihood $p(y|x)$, and has a major impact over the behaviour of the training algorithm since it defines the optimisation landscape. The more complex the hypothesis, potentially the more complex is the landscape. On the other hand, imposing convexity at the optimisation landscape will necessarily lead to a simple linear hypothesis which can only express linear relationships between covariates and response.

Below, we list some of the main theoretical questions we would like to answer for a supervised learning task.

- **Sample complexity:** how many samples n from the data distribution p we need to achieve low excess risk with a given hypothesis class $\mathcal{H}$? What type of functions are "hard" - i.e. require a lot of data - to approximate with a given architecture?

- **Architecture design:** how to choose the hypothesis class $\mathcal{H}$ when we only have access to a finite number of samples? How rich it needs to be? For example: how wide and how deep a network needs to be to approximate a given target function?

- **Algorithmic bias:** how does the choice of algorithm will impact the generalisation? Are there architectures which are "easier" to optimise than others? How the data distribution impacts the optimisation landscape?

These questions are far from new. Indeed, a whole mathematical field, known as *Computational Learning Theory*, was developed in the 80's around formalising and addressing these questions (Valiant, 1984; Vapnik, 2013). On a high-level, the central idea in subject is introduce a suitable complexity measure over the class $\mathcal{H}$ allowing to bound the generalisation gap uniformly over any hypothesis in $\mathcal{H}$ with high-probability over the training data. Therefore, these classical uniform bounds contain at their core the intuition of a trade-off between complexity (both statistical and computational) and generalisation.

However, as we already touched Section 1.2, the current trend in deep learning where every new generation of state-of-the-art neural network is deeper and wider clearly defies this intuition. This discrepancy begs new ideas, and is at the core of current research in machine learning theory. The goal of these lectures is to touch some of the recent mathematical developments in this direction.

Chapter 2

Curses and Blessings of dimensionality

2.1 The curse of dimensionality

Some of the key challenges in machine learning theory arise from the fact that the data and the functions we want to learn live in high-dimensional spaces, such as $\mathbb{R}^d$ with $d \gg 1$. Think for instance of one of the simplest classification problems, MNIST, given by $n = 60000$i mages of black and white digits with 28×28 pixels. When vectorised, this gives a data matrix in $\mathbb{R}^{60000\times 728}$!

The term *curse of dimensionality* was introduced by the Richard E. Bellman, pioneer of dynamical programming, to refer to the hindrance of doing computer science in high-dimensional spaces. We now discuss some examples that illustrate this, and why high-dimensonality does not always make things harder, but sometimes also easier.

2.1.1 k-Nearest neighbours

Consider two vectors sampled uniformly from the hypercube $\boldsymbol{x}, \boldsymbol{x} \sim \text{Unif}([0,1]^d)$. What is their expected Euclidean distance?

$$\begin{aligned}\mathbb{E}[||\boldsymbol{x} - \boldsymbol{x}'||_2^2] &= \sum_{k=1}^{d} \mathbb{E}[(x_k - x_k')^2] = d\ \mathbb{E}[(U - U')^2] \\ &= 2s\Big(\underbrace{\mathbb{E}[U^2]}_{1/3} - \underbrace{\mathbb{E}[U]^2}_{1/2^2}\Big) = \frac{d}{6} \end{aligned} \tag{2.1.1}$$

On the other hand, the variance of the distance is given by:

$$\text{Var}[||\boldsymbol{x} - \boldsymbol{x}'||_2^2] = d\ \text{Var}[(U - U')^2] = \frac{7d}{180} \tag{2.1.2}$$

Therefore, the ratio between the typical fluctuation of the distance (given by the standard deviation) and its expected value is given by:

$$\frac{\text{Std}[||\boldsymbol{x} - \boldsymbol{x}'||_2^2]}{\mathbb{E}[||\boldsymbol{x} - \boldsymbol{x}'||_2^2]} = O\left(\frac{1}{\sqrt{d}}\right) \tag{2.1.3}$$

This means that in the high-dimensional limit $d \to \infty$, the distance between two uniformly sampled points grows much faster than its fluctuations. Why is this a potential a problem?

Let's consider a supervised regression task with training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^{d+1} : i \in [n]\}$ drawn i.i.d. from a model:

$$y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i, \qquad \boldsymbol{x}_i \sim \text{Unif}([0,1]^d) \tag{2.1.4}$$

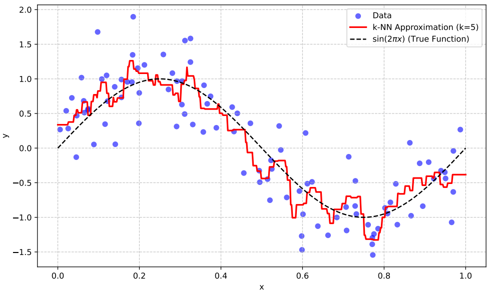

Figure 2.1: Approximating $f_{\star}(x) = \cos(2\pi x)$ with the 5-NN algorithm. Here, we draw $n = 100$ points from $y_i = f_{\star}(x) + \varepsilon$ with $x_i \sim \text{Unif}([0,1])$ and $\varepsilon_i \sim \mathcal{N}(0, 0.2)$.

where $f_{\star} : [0,1]^d \to \mathbb{R}$ is a regular function (e.g. Lipschitz) and ϵ_i independent from $\boldsymbol{x}_i$ and have zero mean. Consider one the first algorithms we learn in the machine learning course: the k-Nearest Neighbours (kNN) algorithm, consisting of estimating the label of a point $\boldsymbol{x}$ by taking an average over the labels of all its k closest neighbours in Euclidean norm:

$$f(\boldsymbol{x}) = \frac{1}{k} \sum_{i:\ k \text{ smallest } ||\boldsymbol{x}-\boldsymbol{x}_i||_2^2} y_i \tag{2.1.5}$$

For regular target functions $f_{\star}$ in $d = 1$, this works very well, see fig. 2.3 for an example. What about for general $d > 1$? In order for eq. (2.1.5) to be meaningful, for every $\boldsymbol{x} \in [0,1]^d$ we need to have at least one training sample $\boldsymbol{x}_i$ nearby. However, in high-dimensions this is not simple: as we have seen in eq. (2.1.3) the distance between uniformly sampled points grows in d. How many training points n we then need in order to have at least one $\boldsymbol{x}_i$ at distance 1 to an arbitrary $\boldsymbol{x} \in [0,1]^d$?

Let any $\boldsymbol{x}_0 \in [0,1]^d$ consider the set of points at distance at least $r > 0$ from $\boldsymbol{x}_0$:

$$B_r(\boldsymbol{x}_0, r) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^d : ||\boldsymbol{x} - \boldsymbol{x}_0||_2 \leq r\} \tag{2.1.6}$$

Asking for having at least one training point $\boldsymbol{x}_i$ distance 1 from any point in the hypercube $\boldsymbol{x} \in [0,1]^d$ is equivalent:

$$[0,1]^d \subset \bigcup_{i=1}^{n} B_d(\boldsymbol{x}_i, 1) \tag{2.1.7}$$

Taking the volume on both sides, we must have:

$$\text{Vol}([0,1]^d) \leq \text{Vol}\left(\bigcup_{i=1}^{n} B_d(\boldsymbol{x}_i, 1)\right) \leq n\text{Vol}\left(B_d(0,1)\right) \tag{2.1.8}$$

However, since:

$$\text{Vol}\left(B_d(0,r)\right) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} r^d \asymp (\frac{2\pi e r^2}{d})^{d/2} (d\pi)^{-1/2} = O(d^{-d/2-\frac{1}{2}}), \text{ as } d \to \infty \tag{2.1.9}$$

while $\text{Vol}([0,1]^d) = 1$, we have:

$$1 \leq n \frac{\pi^{d/2}}{\Gamma(d/2+1)} r^d \qquad \Leftrightarrow \qquad n \geq O(d^{d/2+1/2}) \text{ as } d \to \infty \tag{2.1.10}$$

In other words: to get a meaningful estimation with the kNN estimator, we need exponential data in d!

2.1.2 Grids in high-dimensions

As we will see in section 3.1, many of the classical approximation results in analysis involve approximating a function uniformly on a compact subset of $[0,1]^d \subset \mathbb{R}^d$. A very common proof scheme consists of partitioning $[0,1]^d$ in a uniform grid of constant size and approximating the target function by a piecewise constant function at each element of the grid. In dimension $d = 1$, it is easy to see we need $N = \lceil 1/\delta \rceil$ points to do it. In dimension 2, we need $N = \lceil 1/\delta \rceil^2$. More generally, in dimension d we will need $N = \lceil 1/\delta \rceil^d$ points. Therefore, the finer we want the partition to be, the more points we need, scaling exponentially in the dimension.

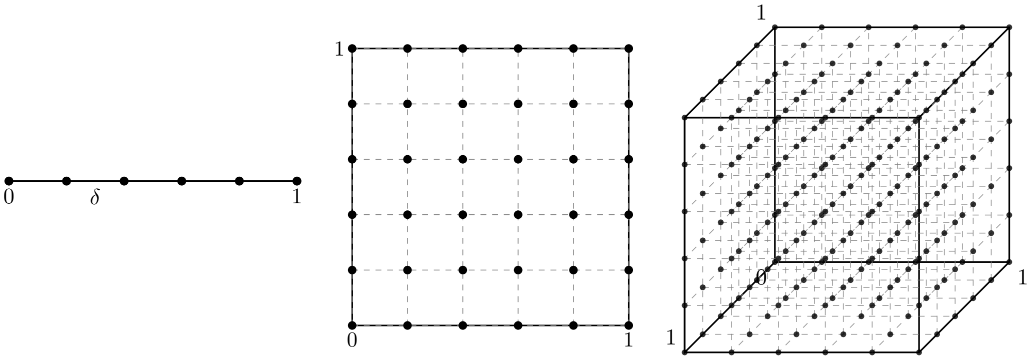

Figure 2.2: Partition of $[0,1]^d$ in a uniform grid of size $\delta = 0.2$ for $d \in \{1,2,3\}$

2.1.3 The empirical covariance is not reliable

The curse of dimensionality also plagues many of the statistical procedures we are used. For instance, consider the problem of estimating the covariance of data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^d$ samples i.i.d. from a distribution. The maximum likelihood estimator in this case is given by the empirical covariance matrix:

$$\hat{\boldsymbol{\Sigma}}_n := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top \tag{2.1.11}$$

where we assumed for simplicity data is centred. If $d = O(1)$ is fixed, by the law of large numbers we have:

$$\hat{\boldsymbol{\Sigma}}_n \xrightarrow{a.s.} \boldsymbol{\Sigma} \text{ as } n \to \infty \tag{2.1.12}$$

where $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is the population covariance of the data. But what happens when d is also large? For instance, when $n = \Theta(d)$? As we will see in Lecture 4, in this limit the matrix $\hat{\boldsymbol{\Sigma}}_n$ can have a very different behaviour. Let's consider a concrete example: take $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ i.i.d. and consider the question: how far are the eigenvalues of $\hat{\boldsymbol{\Sigma}}_n$ from 1 (the eigenvalues of $\boldsymbol{\Sigma} = \boldsymbol{I}_d$)? In eq. (3.2.3) we show a histogram of the eigenvalues of $\hat{\boldsymbol{\Sigma}}_n$ when $d = 500$ and $n = 1000$. Even though the expected eigenvalue is one, most of the eigenvalues are closer to zero. This has important consequences for learning. For instance, if we consider the PCA problem where the goal is to estimate the directions of the data with largest variance, a naive look at the the spectrum of $\hat{\boldsymbol{\Sigma}}_n$ will suggest that most of directions have small variance, which can lead to the misleading conclusion that data is effectively low-dimensional — while the true data distribution is actually isotropic in $\mathbb{R}^d$

2.1.4 Blessings of dimensionality

While learning in high-dimensional spaces certainly come with its challenges, it also comes with benefits, known as the *blessings of dimensionality* — a terminology coined by statistician David

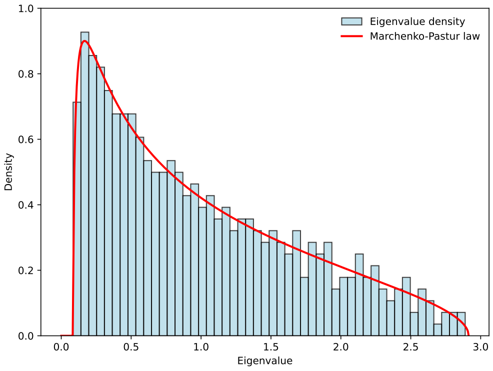

Figure 2.3: Histogram of eigenvalues of the empirical covariance matrix $\hat{\mathbf{\Sigma}}_n$ of i.i.d. covariates $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ with $d = 500$ and $n = 1000$.

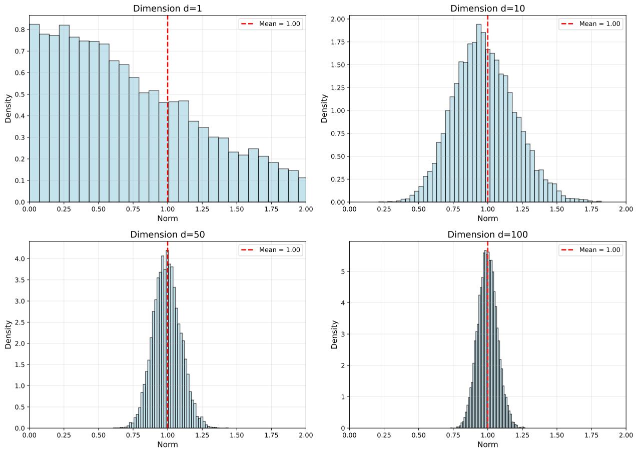

Figure 2.4: Histogram of the norm $||\boldsymbol{x}_i||_2^2$ of $i \in [10^3]$ Gaussian vectors $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, {}^1\!/\!{}_d\boldsymbol{I}_d)$ for increasing $d \in \{1, 10, 50, 100\}$.

Donoho in his AMS math challenges lecture (Donoho et al., 2000). The most notable one is the phenomenon of *concentration of measure*, where the statistical fluctuations of some random quantities of interest get suppressed in high-dimensions.

The most well known example of this is the *thin-shell phenomena* for random Gaussian vector: the property that in dimension $\mathbb{R}^d$, random Gaussian points tightly cluster around the hypersphere.

Proposition 2 (Thin-shell)**.** Let $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ denote a random Gaussian vector. Then, for every $\epsilon > 0$, there exists $C > 0$ such that:

$$\mathbb{P}\left((1-\epsilon)\sqrt{d} \leq ||\boldsymbol{x}||_2 \leq (1+\epsilon)\sqrt{d}\right) \geq 1 - e^{-C(\epsilon)d} \tag{2.1.13}$$

Or in words: with high-probability $\boldsymbol{x}$ is close to $\mathbb{S}^{d-1}(\sqrt{d})$.

The proof of this result follows from Berstein's inequality applied to $||x||_2 - d$, a sum of zero mean sub-exponential random variables. See fig. 2.4 for an illustration. This can have important consequences for computer science.

Theorem 1 (Johnson-Lenderstrauss lemma, 1984)**.** Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$. Then, for any $i, j \in [n]$ and any desired precision $\epsilon > 0$, there exists a linear map $T : \mathbb{R}^d \to \mathbb{R}^k$ with $k \leq C \log n$ such that:

$$(1-\epsilon)||\boldsymbol{x}_i - \boldsymbol{x}_j||_2 \leq ||T(\boldsymbol{x}_i) - T(\boldsymbol{x}_j)||_2 \leq (1+\epsilon)||\boldsymbol{x}_i - \boldsymbol{x}_j||_2 \tag{2.1.14}$$

In other words: T is an approximate isometry.

Although we will not discuss it here, the proof of this theorem is given by taking a Gaussian random mapping and using Gaussian concentration of the norm to show the isotropy property. We refer the interested reader to (Vershynin, 2018) for a proof. This result is striking, since it tell us that any learning algorithm with $d \gg n$ that depends only on the pairwise distances of the data (a.k.a. data *Gram matrix*) can be substantially reduced in computational cost. For instance, if $n = 10^3$ and $d = 10^4$, we have $k \sim 10$!

Chapter 3

Approximation theory

3.1 Universal approximation of neural networks

3.1.1 Motivation & Introduction

Last lecture we have seen that we can decompose the excess risk in two terms:

$$R(\hat{\theta}) - R_{\star} = \underbrace{\left\{ R(\hat{\theta}) - \inf_{\theta\in\Theta} R(\theta) \right\}}_{\text{estimation}} + \underbrace{\left\{ \inf_{\theta\in\Theta} R(\theta) - R_{\star} \right\}}_{\text{approximation}}. \tag{3.1.1}$$

Our focus in today's lecture is about the second term, the *approximation error*, which quantifies the capacity of the hypothesis $\mathcal{H} = \{f_{\theta} : \mathbb{R}^d \to \mathcal{Y} : \theta \in \Theta\}$ in expressing the Bayes predictor. This term is only a function of the choice of architecture and the underlying data distribution, and is independent of the training data. It can also be thought as the irreducible error in the best case scenario where we have infinite amount of data and are able to perfectly optimise the risk.

3.1.2 Introduction to approximation theory

An approximation problem is composed of two ingredients:

1. A target class of functions one wishes to approximate.
2. A metric that defines how good the approximation is.

In eq. (3.1.1), we wish to approximate the Bayes predictor $f_{\star}$. However, the Bayes predictor is implicitly a function of the data distribution and the loss function, and its properties will be problem dependent. Instead of focusing of a particular case, classical approximation theory take as a gold standard the class of continuous functions $\mathcal{C}$, which is often rich enough for machine learning.

The second point depends on the loss function. For instance, a regression problem with the squared-loss will have low excess risk if:

$$||f_{\star} - f_{\theta}||_{L^2(\mu)} := \mathbb{E}_{\boldsymbol{x}}[(f_{\star}(\boldsymbol{x}) - f_{\theta}(\boldsymbol{x}))^2] \tag{3.1.2}$$

is small, where we denoted by μ is the marginal distribution over the covariates $\boldsymbol{x} \in \mathbb{R}^d$. Therefore, the $L^2(\mu)$ error a good enough measure when one wants to benchmark the generalisation error of a network trained by ERM with the square loss, for example. In other cases, other norms might be more adapted. Consider for instance the problem of binary classification with a L-Lipschitz margin-based loss function $\ell(yf_{\theta}(x))$ (e.g. logistic or hinge). Then:

$$\mathbb{E}[\ell(yf_{\theta}(\boldsymbol{x})) - \ell(yf_{\star}(\boldsymbol{x}))] \leq L\ \mathbb{E}[|f_{\theta}(x) - f_{\star}(x)|] := ||f_{\theta} - f_{\star}||_{L_1(\mu)} \tag{3.1.3}$$

which is the $L_1(\mu)$ norm. This is a weaker notion that before, since $L^2(\mu) \subset L^1(\mu)$. More generally, we have that $||\cdot||_p$ is an increasing function of $p \in [1, \infty]$ (Exercise 1), and therefore the norm giving the strongest guarantees is $L^\infty(\mu)$, also known in this context as the *uniform norm.* More frequently, results are proven in terms of the uniform norm over a compact set $K \subset \mathbb{R}^d$:

$$||f||_{L^\infty(K)} = \sup_{\boldsymbol{x} \in K} |f(\boldsymbol{x})| \tag{3.1.4}$$

for example, $K = [0,1]^d$ — and this will be the focus of our discussion in this lecture.

⚠ When $\text{supp}(\mu) \subset K$, we have $||f||_{L^\infty(\mu)} \geq ||f||_{L^\infty(K)}$, but in general these are different notions (Exercise 2). Overall, focusing on $||\cdot||_{L^\infty(K)}$ allow to have results which are independent of the covariate distribution, but which are not adapted to a problem of interest. Indeed, as we will going to see later, these guarantees tend to be pessimistic.

Example 7 (Uniform vs. non-uniform)**.** To get an intuition for uniform vs. non-uniform guarantees, let's consider a simple but instructive example on $[0,1] \subset \mathbb{R}$. Let $g(x) = 0$ for all $x \in [0,1]$ and define:

$$f_n(x) = \begin{cases} 1 & x \in [1, {}^1\!/_n] \\ 0 & \text{otherwise} \end{cases} \tag{3.1.5}$$

Then, we have:

$$||g - f_n||_{L^1([0,1])} = \int_0^1 |f_n(x)| \mathrm{d}x = \frac{1}{n} \tag{3.1.6}$$

and hence, for any $\epsilon > 0$, taking $n > \lceil {}^1\!/_\epsilon \rceil$ we can approximate g in L^1 to precision ϵ. More generally, for any $p \geq 1$ we have:

$$||g - f_n||_{L^p([0,1])} = \left(\int_0^1 |f_n(x)|^p \mathrm{d}x \right)^{{}^1\!/_p} = \frac{1}{n^{{}^1\!/_p}} \tag{3.1.7}$$

and therefore getting a L^p guarantee to precision ϵ requires larger $n > O(\epsilon^{-p})$. Indeed, in uniform norm:

$$||g - f_n||_{L^\infty([0,1])} = 1 \tag{3.1.8}$$

for all $n \in \mathbb{N}$.

This motivates the following definition:

Definition 2 (Universal approximator)**.** Let $K \subset \mathbb{R}^d$ denote a compact subset. We say a class of functions $\mathcal{H}$ is a *universal approximator* over K if for any continuous function $g \in \mathcal{C}(K)$ and any $\epsilon > 0$, there exists $f \in \mathcal{H}$ such that:

$$|f(\boldsymbol{x}) - g(\boldsymbol{x})| \leq \epsilon, \qquad \forall \boldsymbol{x} \in K \tag{3.1.9}$$

Remark 2. Two comments are in order:

- Equation (3.1.9) can be equivalently rewritten to to:

$$\sup_{g \in \mathcal{C}(K)} \inf_{f \in \mathcal{H}} ||f - g||_{L^\infty(K)} < \epsilon \tag{3.1.10}$$

- Most of the proofs that follow will consider $K = [0,1]^d$. This is without loss of generality since we can always cover a compact by a cube and then apply a rescaling.

- Restricting to a compact set K is crucial though. See exercise 4 for one example.

You might remember from your analysis course the following result:

Theorem 2 (Weierstrass theorem, 1885)**.** Polynomials with unbounded degree are universal approximators, i.e. for every continuous real-valued function $f : [a, b] \to \mathbb{R}$ and for every $\epsilon > 0$, there exists a polynomial p such that:

$$\sup_{x\in[a,b]} |f(x) - p(x)| \leq \epsilon \tag{3.1.11}$$

Remark 3. Sometimes, this is also stated as “polynomials are dense in $\mathcal{C}(K)$.

Actually, polynomials are just a particular case of one of the central results in approximation theory:

Theorem 3 (Stone-Weierstrass)**.** Let $K \subset \mathbb{R}^d$ denote a compact subset and $\mathcal{H}$ a class of functions $f : K \to \mathbb{R}$ satisfying the following properties:

- **Continuity:** Every $f \in \mathcal{H}$ is continuous, i.e. $\mathcal{H} \subset \mathcal{C}(K)$.
- **Non-zero element:** For all $\boldsymbol{x} \in K$, there exists $f \in \mathcal{H}$ such that $f(\boldsymbol{x}) \neq 0$. This is sometimes also stated as $\mathcal{H}$ containing the constant functions.
- **Separability:** For all $\boldsymbol{x}, \boldsymbol{x}' \in K$, there exists a $f \in \mathcal{H}$ such that $f(\boldsymbol{x}) \neq f(\boldsymbol{x}')$.
- **Closure:** $\mathcal{H}$ is a sub-algebra of $\mathcal{C}(K)$.[1]

Then, $\mathcal{H}$ is a universal approximator over K.

Remark 4 (Curse of dimensionality)**.** We will not prove this result in the lectures. However, it is worth highlighting that a constructive proof of this result, due to Sergei Berstein in 1912 , involves approximating g by a suitable choice of polynomials (known as Bernstein polynomials) on a grid. Therefore, as a consequence of the discussion in Section 2.1.2, results derived as a consequence of the Stone-Weierstrass theorem 3 typically suffer from the curse of dimensionality, i.e. the number of grid points needed to cover $[0, 1]^d$ with intervals of size ϵ scale with $O(\epsilon^{-d})$.

It is easy to derive theorem 2 from the Stone-Weierstrass theorem by checking that the algebra generated by the coordinates $x_1, \cdots, x_d$ plus contants separates points. More generally, this provides a powerful way of proving that a given class of functions are universal approximators.

3.2 Neural networks

We now move to the main subject of this lecture: neural networks. A L-layer fully connected neural network is a parametric function of the type:

$$\boldsymbol{f}_\theta(\boldsymbol{x}) = \sigma_L\left(\boldsymbol{W}_L\sigma_{L-1}\left(\cdots \boldsymbol{W}_2\sigma_1\left(\boldsymbol{W}_1 x + \boldsymbol{b}_1\right) + \boldsymbol{b}_2 \cdots\right) + \boldsymbol{b}_L\right) \tag{3.2.1}$$

where:

- The parameters $\boldsymbol{W}_\ell \in \mathbb{R}^{p_{\ell+1}\times p_\ell}$, $\ell \in [L]$ are known as the *weight matrices* and $\boldsymbol{b}_\ell \in \mathbb{R}^{p_\ell}$ as the *biases*,[2] and are the trainable parameters of the model $\theta = (\boldsymbol{W}_\ell, \boldsymbol{b}_\ell)_{\ell\in[L]}$.

[1] In other words, it is closed under vector space operations and point-wise multiplication.

[2] The name “bias” is quite misleading, and should in no way be mistaken for the standard notion of bias in statistics.

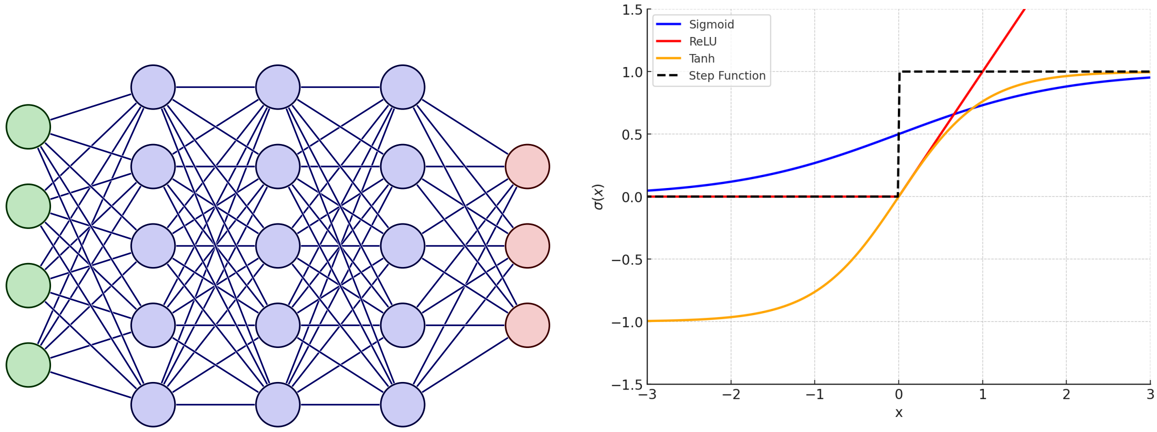

Figure 3.1: (**Left**) Fully connected neural network of depth $L = 4$, hidden-layer widths $p_\ell = 5$ for $\ell \in [4]$ and $p_5 = 4$ in $d = 4$. (**Right**) Popular activation functions used in neural networks.

- The non-linear functions $\sigma_\ell : \mathbb{R} \to \mathbb{R}$ are known as the activation functions. Some of the most common examples, which act component-wise on vectors, are illustrated in fig. 3.1 (right). More general examples with are vector valued are the softmax and the max-pooling activations.

- L is known as the *depth* of the network, and $p_{\ell+1}$ as the *width* of the layer $\ell \in [L]$.

- The first and last layers are known as the *input* and *output* layers, while the middle layers are known as the *hidden layers.*

See fig. 3.1 (left) for an illustration.

⚠ Fully connected networks are one of many classes of neural network architectures, such as convolutional networks, U-networks, transformers, etc. In this lecture, we focus on the fully connected case for simplicity.

A particular case of interest in these lectures will be *two-layer neural networks* ($L = 2$) with

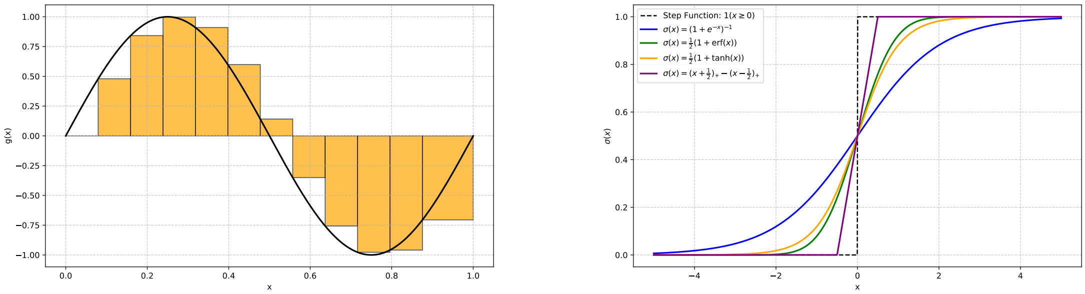

Figure 3.2: (**Left**) Approximating $g(x) = \sin(2\pi x)$ with a two-layer neural network with step-function activation to precision $\epsilon = 0.5$. The function g is 2π-Lipschitz in $[0, 1]$, and hence we have $p = 13$. (**Right**) Different activation functions which are close to the step-function $\mathbf{1}(x \geq 0)$.

real-valued outputs:

$$f_\theta(\boldsymbol{x}) = \langle \boldsymbol{a}, \sigma(\boldsymbol{W}\boldsymbol{x})\rangle = \sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} + b_j\rangle). \tag{3.2.2}$$

where for convenience we relabelled $\boldsymbol{a} := \boldsymbol{W}_2 \in \mathbb{R}^p$ to stress that the last-layer is a vector. Since it will play an important role in what follows, we will denote the class of two-layer neural networks over $\mathbb{R}^d$ with activation function σ as:

$$\mathcal{F}_{\sigma,d,p} := \left\{ f_\theta : \mathbb{R}^d \to \mathbb{R} : f_\theta(x) = \sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, x\rangle + b_j) \right\} \tag{3.2.3}$$

$$\mathcal{F}_{\sigma,d} := \bigcup_{p=0}^{\infty} \mathcal{F}_{\sigma,d,p} \tag{3.2.4}$$

Our goal in what follows is to study the approximation properties of fully-connected networks. On a high-level, we will show that two-layer neural networks with unbounded width are universal approximators in the sense of definition 2. Ideally, we would like to show not only that it is possible to approximate a given class of functions with a neural networks, but also to estimate how many neurons are needed. However, such *quantitative* results are harder to show.

3.2.1 Uniform results on $\mathbb{R}$

We start our discussion by considering $d = 1$.

Proposition 3. Any L-Lipschitz function on $[0, 1]$ can be ϵ-approximated by a two-layer neural network of width $\lceil L/\epsilon \rceil$ and step-function activation $\sigma(x) = \mathbf{1}_{x \geq 0}(x)$.

Proof. First, note that we can create a top-hat function of height 1 in an interval $[a, b]$ by combining two step-functions:

$$\mathbf{1}(x \geq a) - \mathbf{1}(x \geq b) = \mathbf{1}(x \in [a, b]) \tag{3.2.5}$$

Pictorially, it is easy to see that we can approximate any regular g with top-hat functions, see fig. 3.2 (left). The proof essentially follows this intuition. The idea is to partition $[0, 1]$ in p intervals of equal size δx and approximate g in each of these intervals by a top-hat function with height given by how much the function varies. The only tricky part is figuring out the good interval size to obtain the desired precision, which should depend on the regularity of the function.

Consider a L-Lipschitz function $g : [0,1] \to \mathbb{R}$. Given a precision $\epsilon > 0$, let $p = \lceil L/\epsilon \rceil$, and define the two-layer neural network of width p:

$$f_\theta(x) = \sum_{j=0}^{p-1} a_j \mathbf{1}(x - b_j \geq 0) \tag{3.2.6}$$

where:

$$b_j = \frac{j\epsilon}{L}, \qquad j \in \{0, 1, \ldots, p-1\} \tag{3.2.7}$$

$$a_j = \begin{cases} g(0) & j = 1 \\ g(b_j) - g(b_{j-1}) & j > 1. \end{cases} \tag{3.2.8}$$

From eq. (3.2.5), this is equivalent to:

$$f_\theta(x) = \sum_{j=0}^{p-1} g(b_j) \mathbf{1}(x \in [b_j, b_{j+1}]) \tag{3.2.9}$$

It is easy to check that this does the job. Indeed, for any $x \in [0,1]$, define:

$$k = \max\{j \in \{0, 1, \cdots, p-1\} : b_j \leq x\} \tag{3.2.10}$$

such that $x \in [b_k, b_{k+1}]$ and $f(x) = f(b_k)$ is constant in this interval. Therefore:

$$\begin{aligned} |g(x) - f(x)| &= |g(x) - g(b_k) + g(b_k) - f_\theta(b_k)| \\ &\leq |g(x) - g(b_k)| + |g(b_k) - f_\theta(x_k)| + |f_\theta(b_k) - f_\theta(x)| \\ &\leq L|x - b_k| + \left| g(b_k) - \sum_{j=0}^{k} a_j \right| + 0 \\ &\leq L\frac{\epsilon}{L} + \left| g(b_k) - g(b_0) - \sum_{j=1}^{k} (g(b_j) - g(b_{j-1})) \right| \\ &\leq \epsilon \end{aligned} \tag{3.2.11}$$

Since the above was for arbitrary $x \in [0,1]$, it implies the uniform bound. □

Remark 5. The construction in the proof of proposition 3 is suboptimal, since the grid size is adapted only to the global regularity of g in $[0,1]$, not to its local regularity. Indeed, we pay a high price for very regular regions of g, such as flat regions. See exercise 3 for a more detailed discussion of this point.

Since sigmoid-like activation functions such as $\sigma(x) = (1 + e^{-x})$, $\sigma(x) = 1/2(1 + \text{erf}(x))$, $\sigma(x) = 1/2(1 + \tanh(x))$ or even the sum of relu activations $\sigma(x) = \text{relu}(x + \delta) - \text{relu}(x - \delta)$ (see fig. 3.2 (right) for an illustration), it is intuitive that eq. (3.2.5) can be generalised to those cases.

Proposition 4. Every L-Lipschitz function $g : \mathbb{R} \to \mathbb{R}$ can be ϵ-approximated by a two layer neural network of width $p = O(L/\epsilon)$ with sigmoid-like activation σ:

$$\lim_{x \to -\infty} \sigma(x) = 0, \qquad \lim_{x \to \infty} \sigma(x) = 1 \tag{3.2.12}$$

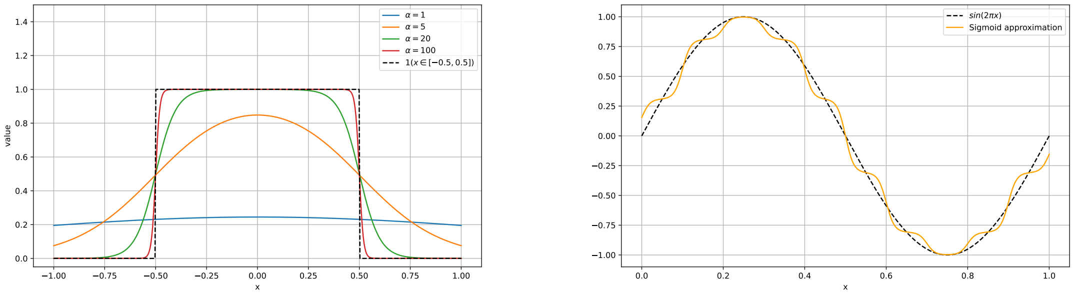

Figure 3.3: (**Left**) Approximating the box function $1(x \in [-1/2, 1/2])$ with the difference of two scaled sigmoids. (**Right**) Approximating the sign function $\sin(2\pi x)$ with a sum of $p = 20$ sigmoid functions.

Sketch of the proof. Note that to show this result, from eq. (3.2.5) it suffices to show that we can approximate the step function $\mathbf{1}(x \geq t)$ with sigmoid-like functions in L^∞ over an interval. The tricky part is that a single sigmoid will not do the job. Indeed, away from the discontinuity $x \neq t$ we can approximate $\mathbf{1}(x \geq t)$ by rescaling $\sigma_\alpha(x) = \sigma(\alpha(t - x))$ with sufficiently large α. However, at the discontinuity $x = t$, the sup norm will be constant. To approximate the jump itself, we have to take two sigmoids with a small shift $\delta > 0$:

$$\phi_{\alpha,\delta}(x) = \sigma(\alpha(t - (x - \delta/2))) - \sigma(\alpha(x - (t + \delta/2))) \tag{3.2.13}$$

For $x \notin [t - \delta/2, t + \delta/2]$ and large α, the two terms are equal so the difference is zero, while for $x \in [t - \delta/2, t + \delta/2]$ the difference is close to 1 at large α. This is an approximation to the box function in an interval, and allow us to provide localised increments, see fig. 3.3 for an illustration.

□

3.2.2 Uniform results on $\mathbb{R}^d$

We now move to $\mathbb{R}^d$ with $d > 1$. Note that the proof of proposition 3 can be decomposed in two steps: first, we showed that two-layer neural neural networks with step-function activation can express the top-hat function in an interval, i.e. eq. (3.2.5). Then, we showed that we can approximate any continuous function by a combination of piece-wise constant functions, which is precisely a linear combination of top-hats. The second part of this construction is easy to generalise to $d > 1$.

Proposition 5. Let $g : \mathbb{R}^d \to \mathbb{R}$ denote a continuous function and $\epsilon > 0$ a desired precision. Then, there exists a partition $\mathcal{P} = (R_1, \cdots, R_N)$ of $[0,1]^d$ into $N = \lceil \delta^{-d} \rceil$ rectangles with side length $\delta > 0$ and real numbers $a_j \in \mathbb{R}$, $j \in [N]$ such that:

$$\sup_{\boldsymbol{x} \in [0,1]^d} \left| g(x) - \sum_{j=1}^{N} a_j \mathbf{1}(\boldsymbol{x} \in R_j) \right| < \epsilon \tag{3.2.14}$$

In other words: linear combinations of indicator functions are universal approximators over $[0,1]^d$.

Proof. Let $\mathcal{P} = (R_1, \cdots, R_N)$ of $[0,1]^d$ denote a partition of $[0,1]^d$ into N rectangles with side length smaller than $\delta > 0$. Note that since each rectangle has volume δ^d, this requires $N = \lceil \delta^{-d} \rceil$ rectangles. What we need to show is that we can choose δ and a_j such that eq. (3.2.14) holds. For every rectangle R_j, take an arbitrary point $\boldsymbol{x}_j \in R_j$ and let:

$$a_j = g(\boldsymbol{x}_j) \tag{3.2.15}$$

be the corresponding height of the top-hat function. Then:

$$\begin{aligned} \sup_{\boldsymbol{x}\in[0,1]^d} |g(\boldsymbol{x}) - f(\boldsymbol{x})| &= \sup_{j\in[N]} \sup_{\boldsymbol{x}\in R_j} |f(\boldsymbol{x}) - g(\boldsymbol{x})| && (3.2.16)\\ &\leq \sup_{j\in[N]} \sup_{\boldsymbol{x}\in R_j} \left\{ |g(\boldsymbol{x}) - g(\boldsymbol{x}_i)| + \underbrace{|g(\boldsymbol{x}_i) - f(\boldsymbol{x})|}_{=0} \right\} && (3.2.17)\\ &= \sup_{j\in[N]} \sup_{\boldsymbol{x}\in R_j} |g(\boldsymbol{x}) - g(\boldsymbol{x}_i)| && (3.2.18) \end{aligned}$$

Since g is continuous on $\mathbb{R}^d$, it is uniformly continuous on the compact set $[0,1]^d$, and therefore for every $\epsilon > 0$, there exists $\delta > 0$ such that:

$$||\boldsymbol{x} - \boldsymbol{x}'||_\infty \leq \delta \qquad \Rightarrow \qquad |g(\boldsymbol{x}) - g(\boldsymbol{x}')| \leq \epsilon \tag{3.2.19}$$

Therefore, choosing the size of our rectangle to be such δ, we have:

$$\sup_{\boldsymbol{x}\in[0,1]^d} |g(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \epsilon. \tag{3.2.20}$$

□

Remark 6 (Curse of dimensionality, again)**.** As we had already seen in Section 2.1.2, the number of indicator function required to approximate g scale exponentially in the dimension $N \sim \delta^{-d}$, which is an instance of the curse of dimensionality.

To establish a similar result to proposition 3, we now need to approximate the indicator function over the rectangles $\mathbf{1}(R_j)$ with a neural network. The first proof of this result in the two-layer case is due to Cybenko (1989) with sigmoid functions and a slightly different argument. Here, we will present the proof by Hornik et al. (1989) which is based on the Stone-Weierstrass theorem 3.

Theorem 4 (Hornik, Stinchcombe, White 1989)**.** The class of two-layer neural networks with unbounded width and sigmoid-like activation function σ:

- σ is increasing.
- $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$.

Are universal approximators over $[0,1]^d$. In other words, for any continuous function $g : \mathbb{R}^d \to \mathbb{R}$, there exists a two-layer neural network $f \in \mathcal{F}_{\sigma,d}$ such that for any $\epsilon > 0$:

$$\sup_{\boldsymbol{x}\in[0,1]^d} |g(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \epsilon \tag{3.2.21}$$

The proof of theorem 4 proceeds in two steps: first, we will use the Stone-Weierstrass theorem to prove that two-layer networks with $\sigma(x) = \cos(x)$ are universal approximators. Then, we will prove that we can approximate the cosine with any activation satisfying the properties above.

Lemma 1. Two layer neural networks of unbounded width with cosine activation are universal approximators.

Proof. In order to apply the Stone-Weierstrass theorem, we need to check that the class of two-layer networks:

$$\mathcal{F}_{\cos,d} = \bigcup_{p\geq 1} \left\{ f_\theta(x) = \sum_{j=1}^{p} a_j \cos(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_j) \right\} \tag{3.2.22}$$

satisfy the conditions of theorem 3:

- **Contituity:** Elements of $\mathcal{F}_{\cos,d,p}$ are continuous since they are linear combinations of continuous functions $(\cos(x))$.
- **Non-zero element:** For all $\boldsymbol{x} \in [0,1]^d$, $1 \in \mathcal{F}_{\cos,d}$ since $\cos(\langle \mathbf{0}, \boldsymbol{x} \rangle) = 0$.
- **Separability:** For any $\boldsymbol{x}, \boldsymbol{x}' \in [0,1]^d$, there exists a $\boldsymbol{w} \in \mathbb{R}^d$ such that $\cos(\langle \boldsymbol{w}, \boldsymbol{x} \rangle) \neq \cos(\langle \boldsymbol{w}, \boldsymbol{x}' \rangle)$. Take for instance:

$$f(\boldsymbol{z}) = \cos\left(\frac{\langle (\boldsymbol{z} - \boldsymbol{x}'), (\boldsymbol{x} - \boldsymbol{x}') \rangle}{||\boldsymbol{x} - \boldsymbol{x}'||_2^2}\right) \tag{3.2.23}$$

  which satisfies $f(\boldsymbol{x}) = 1$ and $f(\boldsymbol{x}') = 0$.

- **Closure:** $\mathcal{F}_{\cos,d}$ is clearly closed under addition and multiplication by a real. It is also closed under multiplication since:

$$\cos(x)\cos(y) = \frac{1}{2}(\cos(x+y) + \cos(x-y)) \tag{3.2.24}$$

  Therefore, for any $f, h \in \mathcal{F}_{\cos,d}$:

$$\begin{aligned} f(\boldsymbol{x})h(\boldsymbol{x}) &= \left(\sum_{j=1}^{p} a_j \cos(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle + b_j)\right)\left(\sum_{k=1}^{p'} c_k \cos(\langle \boldsymbol{u}_k, \boldsymbol{x} \rangle + d_k)\right) \\ &= \sum_{j=1}^{p}\sum_{k=1}^{p'} a_j c_k \cos(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle + b_j)\cos(\langle \boldsymbol{u}_k, \boldsymbol{x} \rangle + d_k) \\ &= \frac{1}{2}\sum_{j=1}^{p}\sum_{k=1}^{p'} a_j c_k \left[\cos\left(\langle \boldsymbol{w}_j + \boldsymbol{u}_k, \boldsymbol{x} \rangle + b_k + c_k\right) + \cos\left(\langle \boldsymbol{w}_j - \boldsymbol{u}_k, \boldsymbol{x} \rangle + b_k - c_k\right)\right] \end{aligned} \tag{3.2.25}$$

  which implies $fh \in \mathcal{F}_{\cos,d}$.

Therefore, by the Stone-Weierstrass theorem 3, two-layer neural networks with cosine activation are universal approximators over $[0,1]^d$. □

Remark 7. As you will show in exercise 5, the activation $\sigma(x) = \exp(x)$ also satisfy the conditions of the Stone-Weierstrass theorem.

We now move to the proof of the general case.

Sketch of the proof. Thanks to theorem 4, for any continuous function $g : \mathbb{R}^d \to \mathbb{R}$, there exists a two-layer neural network of width p with cosine activation:

$$f(\boldsymbol{x}) = \sum_{j=1}^{p} \tilde{a}_j \cos\Big(\langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x} \rangle + \tilde{b}_j\Big) \tag{3.2.26}$$

such that:

$$\sup_{\boldsymbol{x} \in [0,1]^d} |g(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \frac{\epsilon}{2} \tag{3.2.27}$$

Therefore, our goal is to approximate $f(x)$ by a two-layer neural network with sigmoid-like activation. First, note that for any $x \in [0,1]^d$, the pre-activations are bounded:

$$t_j = \langle \tilde{\boldsymbol{w}}_j, \boldsymbol{x} \rangle + \tilde{b}_j \in [|\tilde{b}_j|, |\langle \tilde{\boldsymbol{w}}_j, \mathbf{1} \rangle| + |\tilde{b}_j|] \tag{3.2.28}$$

Therefore, we can focus on the univariate problem of approximating $\cos(x)$ on a compact interval with a sigmoid two-layer neural network, which we saw can be done with proposition 4. □

Remark 8. The most general universal approximation result for two-layer neural networks, proven by Leshno et al. (1993), shows that $\mathcal{F}_{\sigma,d}$ with σ a locally bounded piece-wise continuous activation is a universal approximator if and only if σ is not a polynomial.

3.3 Quantitative L^2 results

So far, we have discussed only uniform approximation results. Although they are general and strong, they do not provide quantitative rates, such as an estimate of how fast the error goes down with the width. Moreover, as we saw these results suffer from the curse of dimensionality. This is a fundamental trade-off due to the fact that we consider a very broad class of functions (continuous functions) and require a strong guarantee (L^∞, uniform norm).

Getting quantitative approximation rates require making more precise assumptions. In this section, we discuss a celebrated result from Andrew Barron in this direction, where the key idea is to explore further the regularity of the functions (Barron, 1993). A trivial but instructive example of how regularity can help is given by differentiable functions $f : \mathbb{R} \to \mathbb{R}$ with $f(0) = 0$. By the fundamental theorem of calculus, we can write:

$$f(x) = f(0) + \int_0^x f'(t)\mathrm{d}t = \int_0^1 \mathbf{1}(x \geq t)g'(t)\mathrm{d}t \tag{3.3.1}$$

When discretised, this is precisely a two-layer neural network with step-function activation that we discussed in proposition 3! Moreover, the discretisation error scales as $O(1/\sqrt{p})$ in $L^2([0,1])$ norm. Barron's construction can be seen as a sophisticated version of this simple example.

3.3.1 Infinite width limit of two-layer networks

Consider the two-layer neural network eq. (3.2.2):

$$f_\theta(\boldsymbol{x}) = \frac{1}{p}\sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_j) \tag{3.3.2}$$

where, without loss of generality, we have rescaled $a_j \mapsto a_j/p$. Introducing the following empirical measure over the network parameters:

$$\hat{\mu}_p = \frac{1}{p}\sum_{j=1}^{p} \delta_{\theta_j} = \frac{1}{p}\sum_{j=1}^{p} \delta_{(a_j, \boldsymbol{w}_j, b_j)} \tag{3.3.3}$$

we can rewrite eq. (3.3.2) as an integral over $\hat{\mu}_p$:

$$f_\theta(\boldsymbol{x}) = \int_\Omega a\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b)\hat{\mu}_p(\mathrm{d}a, \mathrm{d}\boldsymbol{w}, \mathrm{d}b) \tag{3.3.4}$$

with $\Omega = \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}$.

Remark 9. This construction only requires an empirical sum structure, and therefore we could write a similar expression for any map $\varphi(\boldsymbol{x}, \theta)$, with the ridge function $\varphi(\boldsymbol{x}, \theta) = a\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b)$ being a particular example with $\theta = (a, \boldsymbol{w}, b) \in \Omega$:

$$f_\theta(\boldsymbol{x}) = \int_\Omega \varphi(\boldsymbol{x}; \theta)\hat{\mu}_p(\mathrm{d}\theta) \tag{3.3.5}$$

Informally, we expect that when the width is large $p \gg 1$, $\hat{\mu}_p$ to be close (in a weak sense) to a limiting measure μ over Ω. In other words: a finite width network can be seen as a discretisation of an infinite width network:

$$f_\theta(\boldsymbol{x}) = \int_\Omega \varphi(\boldsymbol{x}; \theta)\mu(\mathrm{d}\theta) \tag{3.3.6}$$

The key idea of Barron's construction is to show that a class of sufficiently regular functions (to be defined next) can be written in the form eq. (3.3.6), and therefore they can be approximated by two-layer neural networks up to a discretisation error.

3.3.2 Barron's space

As motivated above, Barron's construction exploits the regularity of the target function, and relies on its Fourier decomposition. Recall that any integrable function $g \in L^1(\mathbb{R}^d)$ admits a Fourier decomposition:

$$g(\boldsymbol{x}) = \int_{\mathbb{R}^d} \mathrm{d}\boldsymbol{\xi}\ \hat{g}(\boldsymbol{\xi}) e^{i\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle} \tag{3.3.7}$$

where $\hat{g} \in L^1(\mathbb{R}^d)$ are the Fourier coefficients. Since this is invertible, we can also express $\hat{g}$ in terms of g:

$$\hat{g}(\boldsymbol{\xi}) = \int_{\mathbb{R}^d} \frac{\mathrm{d}\boldsymbol{x}}{(2\pi)^d}\ g(\boldsymbol{x}) e^{-i\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle} \tag{3.3.8}$$

One of the useful properties of the Fourier transform is that it linearise gradients. Assuming g is differentiable:

$$\nabla g(\boldsymbol{x}) = \int_{\mathbb{R}^d} \mathrm{d}\boldsymbol{\xi}\ \hat{g}(\boldsymbol{\xi}) e^{i\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle} i\boldsymbol{\xi} \qquad \Leftrightarrow \qquad \widehat{\nabla g}(\boldsymbol{\xi}) = i\boldsymbol{\xi}\ \hat{g}(\xi) \tag{3.3.9}$$

This property provides a useful way of measuring the regularity of g. Indeed, in order for this expression to be mathematically well-defined, we need that $\widehat{\nabla g} \in L^1(\mathbb{R}^d)$, which means that the coefficients $|\hat{g}(\boldsymbol{\xi})|$ must decay faster than $||\boldsymbol{\xi}||_2$. More generally, if all derivatives of g up to order $m \in \mathbb{N}$ are integrable, we must have:

$$|\hat{g}(\boldsymbol{\xi})| \leq C||\boldsymbol{\xi}||^{-m}, \text{ as } ||\boldsymbol{\xi}||_2 \to \infty \tag{3.3.10}$$

Therefore, from the point of view of the Fourier transform differentiable functions are functions for which the Fourier coefficients decay faster than polynomials. From the FTC argument in eq. (3.3.1), it will not be surprising that having a finite gradient norm is exactly what we need to write a function in terms of an infinite width two-layer neural network.

Definition 3 (Barron space)**.** Let $f \in L^1(\mathbb{R}^d)$ denote an integrable function. We define its *Barron norm* as:

$$||f||_B \coloneqq \int_{\mathbb{R}^d} ||\boldsymbol{\xi}|| \cdot |\hat{f}(\boldsymbol{\xi})| \mathrm{d}\boldsymbol{\xi} = \int_{\mathbb{R}^d} ||\widehat{\nabla f}(\boldsymbol{\xi})|| \mathrm{d}\boldsymbol{\xi} \tag{3.3.11}$$

The Barron space $\mathcal{F}_B \subset L^1(\mathbb{R}^d)$ is then defined as the space of integrable functions with finite Barron norm:

$$\mathcal{F}_B \coloneqq \{f \in L^1(\mathbb{R}^d) : ||f||_B < \infty\} \tag{3.3.12}$$

⚠ Note that since the Barron norm $||\cdot||_B$ is defined through the derivative of the function, it is strictly speaking not a norm but a semi-norm. Indeed, two distinct functions $f, g \in L^1(\mathbb{R}^d)$ can have the same Barron norm $||f||_B = ||g||_B$ without being identical $f(\boldsymbol{x}) \neq g(\boldsymbol{x})$ at all $\boldsymbol{x} \in \mathbb{R}^d$, as long as $f(\mathbf{0}) \neq g(\mathbf{0})$. For $\mathcal{F}_B$ to define a proper Banach space, we need to fix $f(\mathbf{0})$.

Example 8. To get some intuition, it is useful to look at the Barron norm of some well-known functions.

- **Gaussian:** For $g(\boldsymbol{x}) = e^{-1/2||\boldsymbol{x}||_2^2}$ with $\boldsymbol{x} \in \mathbb{R}^d$, we have $||g||_B = O(\sqrt{d})$. You will show this in exercise 6.

- **Single neuron:** Let $g(\boldsymbol{x}) = \sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b)$ with $\boldsymbol{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ and $\sigma : \mathbb{R} \to \mathbb{R}$. This is a single-neuron of a neural network, also known as *ridge function*. We have:

$$||g||_B \leq ||\boldsymbol{w}|| \int_{\mathbb{R}} |\xi\hat{\sigma}(\xi)|\mathrm{d}\xi \tag{3.3.13}$$

- See Section IX of (Barron, 1993) for a big list of examples.

⚠ The ReLU activation $\sigma(x) = x_+$ has infinite Barron norm - see exercise 6.

Remark 10. All the notions in this section can be similarly defined to functional spaces over a probability density μ over $\mathbb{R}^d$.

3.3.3 Barron's theorem

We are now ready to state Barron's theorem for two-layer neural networks.

Theorem 5 (Barron (1993))**.** Let p_x be a probability distribution with $\mathrm{supp}(p_x) \subset B(0, R)$. Then, for any integrable function $g \in L^1(p_x)$, there exists a two-layer neural network $f_\theta \in \mathcal{F}_{\sigma,d,p}$ with p neurons and sigmoid-like activation σ such that:

$$||f_\theta - f(\mathbf{0}) - g||_{L^2(p_x)} \leq \frac{2R||g||_B}{\sqrt{p}} \tag{3.3.14}$$

Furthermore, we can impose that the second-layer weights are bounded: $\sum_{j=1}^{p} |a_j| \leq 2R||g||_B$

Remark 11. Two comments on this result are in order.

- Barron's theorem can be alternatively stated as: for any $g \in L^1(p_x)$ and desired precision $\epsilon > 0$, there exists a two-layer neural network $f_\theta \in \mathcal{F}_{\sigma,d,p}$ with width:

$$p(\epsilon) = \left(\frac{2R||g||_B}{\epsilon}\right)^2 \tag{3.3.15}$$

  such that $||f_\theta - f(\mathbf{0}) - g||_{L^2(p_x)} \leq \epsilon$.

- Note that the width is quadratic in the precision $p = O(\epsilon^{-2})$, which is much better than the exponential dependence on the dimension of universal results based on grids, such as theorem 4. This is often stated as a "dimension-free" result that "avoids the curse of dimensionality". However, it is important to keep in mind that $||g||_B$ can potentially hide a dependence in d, such as in example 8 for the Gaussian density where $||g||_B = O(\sqrt{d})$.

- Barron's original theorem does not directly apply to the ReLU activation. A generalisation of these ideas to ReLU is discussed in (Bach, 2017).

The proof of Barron's theorem involve two steps, which we will discuss separately:

(a) Showing that any function $g \in \mathcal{F}_B$ can be written as an infinite width two-layer neural network.

(b) Showing that an infinite width network can be approximated by a finite width network to the desired precision $\epsilon > 0$ with $p(\epsilon)$ as in eq. (3.3.15).

We start by discussing the first step.

Proposition 6. Let $g \in \mathcal{F}_B$. Then, for every $\boldsymbol{x} \in B(0, R)$, there exists a probability measure μ with $\text{supp}(\mu) \subset \Omega := [-M, M] \times \mathbb{R}^{d+1}$ such that:

$$g(\boldsymbol{x}) - g(\mathbf{0}) = \int_{\Omega} \varphi(\boldsymbol{x}, \boldsymbol{\theta})\mu(\mathrm{d}\theta) \tag{3.3.16}$$

with $M := R||f||_B$ and $\varphi(\boldsymbol{x}, \boldsymbol{\theta}) = a\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b)$ a ridge function with sigmoid-like activation σ

Sketch of the proof. The key idea is to use the Fourier transform of g to write it in an integral form that ressambles the infinite width limit of a two-layer network. Since g is real-valued, we can write:

$$\begin{aligned} g(\boldsymbol{x}) - g(\mathbf{0}) &= \text{Re}\left[\int_{\mathbb{R}^d} \hat{g}(\boldsymbol{\xi})\left(e^{i\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle} - 1\right) \mathrm{d}\boldsymbol{\xi}\right] \\ &\overset{(a)}{=} \text{Re}\left[\int_{\mathbb{R}^d} |\hat{g}(\boldsymbol{\xi})| e^{i\alpha(\xi)}\left(e^{i\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle} - 1\right) \mathrm{d}\boldsymbol{\xi}\right] \\ &= \int_{\mathbb{R}^d} |\hat{g}(\boldsymbol{\xi})| \left[\cos\left(\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle + \alpha(\boldsymbol{\xi})\right) - \cos\left(\alpha(\boldsymbol{\xi})\right)\right] \mathrm{d}\boldsymbol{\xi} \end{aligned} \tag{3.3.17}$$

where in (a) we used the polar decomposition of $\hat{g}$. Note that this looks exactly like the infinite width of a neural network with cosine activation. However, this differs from the desired result in two important ways: the integral is over the Lebesgue measure and the second layer weights are unbounded. To fix these points, we rewrite:

$$\begin{aligned} g(\boldsymbol{x}) - g(\mathbf{0}) &= \int_{\mathbb{R}^d} \frac{||g||_B}{||\boldsymbol{\xi}||_2} \left[\cos\left(\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle + \alpha(\boldsymbol{\xi})\right) - \cos\left(\alpha(\boldsymbol{\xi})\right)\right] \frac{||\boldsymbol{\xi}||_2 \cdot |\hat{g}(\boldsymbol{\xi})|}{||g||_B} \mathrm{d}\boldsymbol{\xi} \\ &= \int_{\mathbb{R}^d} \varphi(\boldsymbol{x}, \boldsymbol{\xi})\mu(\mathrm{d}\boldsymbol{\xi}) \end{aligned} \tag{3.3.18}$$

where we have defined:

$$\varphi(\boldsymbol{x}, \boldsymbol{\xi}) = \frac{||g||_B}{||\boldsymbol{\xi}||_2} \left[\cos\left(\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle + \alpha(\boldsymbol{\xi})\right) - \cos\left(\alpha(\boldsymbol{\xi})\right)\right], \qquad \mu(\mathrm{d}\boldsymbol{\xi}) := \frac{||\boldsymbol{\xi}||_2 \cdot |\hat{g}(\boldsymbol{\xi})|}{||g||_B} \mathrm{d}\boldsymbol{\xi} \tag{3.3.19}$$

In particular, note that φ is bounded. Indeed, since cos is 1-Lipschitz:

$$|\cos\left(\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle + \alpha(\boldsymbol{\xi})\right) - \cos\left(\alpha(\boldsymbol{\xi})\right)| \leq |\langle \boldsymbol{\xi}, \boldsymbol{x}\rangle| \leq ||\boldsymbol{x}||_2 \cdot ||\boldsymbol{\xi}||_2 \tag{3.3.20}$$

where in the last inequality we used Cauchy-Schwarz. Hence:

$$|\varphi(\boldsymbol{x}, \boldsymbol{\xi})| \leq ||g||_B \cdot ||\boldsymbol{x}||_2 \leq R||g||_B \tag{3.3.21}$$

since $\boldsymbol{x} \in B(0, R)$. Moreover, μ is a probability measure over $\mathbb{R}^d$ since:

$$\int_{\mathbb{R}^d} \mu(\mathrm{d}\boldsymbol{\xi}) = \frac{1}{||g||_B} \int_{\mathbb{R}^d} ||\boldsymbol{\xi}||_2 \cdot |\hat{g}(\boldsymbol{\xi})| \mathrm{d}\boldsymbol{\xi} = \frac{||g||_B}{||g||_B} = 1 \tag{3.3.22}$$

This almost concludes the proof. It remains just to show that for each $\boldsymbol{\xi} \in \mathbb{R}^d$, $\varphi(\cdot, \boldsymbol{\xi})$ in eq. (3.3.19) can be approximated by a sum of sigmoid-like functions which is a consequence of proposition 4. □

Step (b) in the proof of theorem 5 uses a classical proof scheme in probability: to create an approximation for g by sampling a finite width network from the measure defined in proposition 6.

Proof of Barron's theorem. . Let $g \in \mathcal{F}_B$, and without loss of generality assume $g(\mathbf{0}) = 0$. Then, by proposition 6 there exists a probability measure μ with $\text{supp}(\mu) \subset \Omega$ such that:

$$g(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\theta} \sim \mu}[\varphi(\boldsymbol{x}, \boldsymbol{\theta})] \tag{3.3.23}$$

with $\varphi(\boldsymbol{x}, \boldsymbol{\theta}) = a\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b)$. We now consider p independent samples $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_p \sim \mu$, and let:

$$f(\boldsymbol{x}; \boldsymbol{\Theta}) = \frac{1}{p}\sum_{j=1}^{p} \varphi(\boldsymbol{x}, \boldsymbol{\theta}_j) \tag{3.3.24}$$

denote a random estimate of g. Note that by construction we have $f(\cdot; \boldsymbol{\Theta}) \in \mathcal{F}_{\sigma,d,p}$. In particular, since $\boldsymbol{\theta}_j$ are independent we have:

$$\mathbb{E}_{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_p\sim\mu}[f(\boldsymbol{x}; \boldsymbol{\Theta})] = g(\boldsymbol{x}) \tag{3.3.25}$$

and by the law of large numbers, $f(\cdot; \boldsymbol{\Theta}) \to g$ almost surely as $p \to \infty$. The desired approximation error therefore corresponds exactly to this estimation error:

$$\mathbb{E}_{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_p\sim\mu}||f(\cdot; \boldsymbol{\Theta}) - g||^2_{L^2(p_x)} = \mathbb{E}_{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_p\sim\mu}||f(\cdot; \boldsymbol{\Theta})||^2_{L^2(p_x)} - ||g||^2_{L^2(p_x)} \tag{3.3.26}$$

Focusing on the first terms, note that we have:

$$\begin{aligned} ||f(\cdot; \boldsymbol{\Theta})||^2_{L^2(p_x)} &= \left|\left|\frac{1}{p}\sum_{j=1}^{p}\varphi(\cdot; \boldsymbol{\theta}_j)\right|\right|^2_{L^2(p_x)} \\ &= \frac{1}{p^2}\sum_{j,k=1}^{p}\langle\varphi(\cdot; \boldsymbol{\theta}_j), \varphi(\cdot; \boldsymbol{\theta}_k)\rangle_{L^2(p_x)} \\ &= \frac{1}{p^2}\sum_{j=1}^{p}||\varphi(\cdot; \boldsymbol{\theta}_j)||^2_{L^2(p_x)} + \frac{1}{p^2}\sum_{j\neq k}\langle\varphi(\cdot; \boldsymbol{\theta}_j), \varphi(\cdot; \boldsymbol{\theta}_k)\rangle_{L^2(p_x)} \end{aligned} \tag{3.3.27}$$

Now, taking the expectation with respect to $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_p \sim \mu$ and using independence:

$$\mathbb{E}_{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_p\sim\mu}||f(\cdot; \boldsymbol{\Theta})||^2_{L^2(p_x)} = \frac{1}{p}\mathbb{E}_{\boldsymbol{\theta}\sim\mu}\left[||\varphi(\cdot; \boldsymbol{\theta})||^2_{L^2(p_x)}\right] + \left(1 - \frac{1}{p}\right)||g||_{L^2(p_x)} \tag{3.3.28}$$

Inserting this back into eq. (3.3.26):

$$\mathbb{E}_{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_p\sim\mu}||f(\cdot; \boldsymbol{\Theta}) - g||^2_{L^2(p_x)} = \frac{\mathbb{E}_{\boldsymbol{\theta}\sim\mu}||\varphi(\cdot, \boldsymbol{\theta})||^2_{L^2(p_x)} - ||g||^2_{L^2(p_x)}}{p} \leq \frac{\mathbb{E}_{\boldsymbol{\theta}\sim\mu}||\varphi(\cdot, \boldsymbol{\theta})||^2_{L^2(p_x)}}{p} \tag{3.3.29}$$

Which is exactly the desired rate in the width. To conclude the proof, we just need to remark that since $\operatorname{supp}(\mu) \subset \Omega$ and $\operatorname{supp}(p_x) \subset B(0, R)$:

$$\mathbb{E}_{\boldsymbol{\theta}\sim\mu}||\varphi(\cdot, \boldsymbol{\theta})||^2_{L^2(p_x)} \leq ||\varphi||^2_{L^2(B(0,R)\times\Omega)} \leq ||\varphi||^2_{L^\infty(B(0,R)\times\Omega)} \leq R^2||g||^2_B \cdot ||\sigma||^2_{L^\infty(\mathbb{R})} \tag{3.3.30}$$

where in the last inequality we used that:

$$|\varphi(\boldsymbol{x}, \boldsymbol{\theta})| = |a| \cdot |\sigma(\langle \boldsymbol{w}, \boldsymbol{x} + b)| \leq M||\sigma||_{L^\infty(\mathbb{R})} = R||g||_B \tag{3.3.31}$$

Since for sigmoid-like function we have $||\sigma||_{L^\infty(\mathbb{R})} = 1$. Therefore, since the expectation over $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p \sim \mu$ is finite, the probability of the event that $f(\cdot, \boldsymbol{\Theta})$ satisfies the estimation error:

$$||f(\cdot; \boldsymbol{\Theta}) - g||_{L^2(p_x)} \leq \frac{R||g||_B}{\sqrt{p}} \tag{3.3.32}$$

is finite, which concludes the proof. □

To go further

In these lectures, we have seen that sufficiently wide two-layer neural networks with sigmoid-like activation are able to approximate well-behaved function in a compact set. We have discussed both uniform for a broad function class (continuous function) - which are strong, but come at a cost of an exponential dependence of the network width on the data dimension - and L^2 results for regular functions (Barron functions) - which are weaker but for which the width adapts to the regularity of the function.

These results date back from the 90's, and there are modern extensions in many directions, some of which we have already referred to. One exciting set of more recent results concern *depth separation*. While two-layer networks are universal approximations, they are barely used outside the scope of theoretical works, and as the name emphasises, *deep learning* is really about deep networks. Therefore, provably showing that deep networks have better approximation properties than shallow networks is an important endeavour. To wrap up our discussion about approximation theory with neural networks, we mention of such result, hoping it will motivate the reader to go *deeper*

Theorem 6 (Telgarsky (2015, 2016))**.** Let $L > 1$. There exists $g : \mathbb{R} \to \mathbb{R}$ computed by a neural network with ReLU activations with $2L^2 + 4$ layers and $p = 3L^3 + 12$ nodes but such that any network f with $\leq L$ layers and $\leq 2^L$ nodes cannot approximate it:

$$||f - g||_{L^1([0,1])} \geq \frac{1}{32} \tag{3.3.33}$$

Remark 12. While positive results are strong in the strongest norm (L^∞, uniform), negative results such as this one are strongest in the weakest norm (L^1). This shows the network f badly fails to approximate g.

3.4 Exercises

Exercise 1. Let μ denote a probability measure over $\mathbb{R}^d$. Show that the $L^p(\mu)$ norm:

$$||f||_{L^p(\mu)} := \left(\int \mu(\mathrm{d}\boldsymbol{x})|f(\boldsymbol{x})|^p\right)^{1/p} \tag{3.4.1}$$

is an increasing function of $p \in [1, \infty]$:

$$||f||_{L^\infty(\mu)} \geq \cdots \geq ||f||_{L^2(\mu)} \geq ||f||_{L^1(\mu)} \tag{3.4.2}$$

where we recall $||f||_{L^\infty(\mu)} = \sup_{x \in \mathrm{supp}(\mu)} |f(\boldsymbol{x})|$. Conclude that we have the inclusion:

$$L^\infty(\mu) \subset \cdots \subset L^2(\mu) \subset L^1(\mu) \tag{3.4.3}$$

Exercise 2. Let μ denote a probability measure in $\mathbb{R}$ and $[a, b] \subset \mathbb{R}$ a compact subset of your choice. Give examples of functions $f, g : \mathbb{R} \to \mathbb{R}$ such that:

$$||f||_{L^\infty(\mu)} \leq ||f||_{L^\infty([a,b])}, \qquad \text{and } ||g||_{L^\infty(\mu)} \geq ||g||_{L^\infty([a,b])} \tag{3.4.4}$$

Note: In the above, we want $[a, b]$ and μ to be the same in both inequalities.

Exercise 3. Consider the following continuous function on $\mathbb{R}$:

$$g(x) = \begin{cases} 0 & x < -1/2 \\ x + 1/2 & x \in [-1/2, 1/2] \\ 1 & x > 1/2 \end{cases} \tag{3.4.5}$$

How many neurons p are needed to approximate g within a precision $\epsilon > 0$ on the compact set $[-1,1] \subset \mathbb{R}$ using the two-layer neural network with step-size activation from Proposition 2 in the lectures? Show that we could do as well by using fewer neurons if we adapt the partition to the function.

Exercise 4. Show that:

$$\inf_{f_\theta \in \mathcal{F}_{\text{relu},1}} \sup_{x \in \mathbb{R}} |f_\theta(x) - \sin(x)| \geq 1 \tag{3.4.6}$$

where $\mathcal{F}_{\text{relu},1}$, the class of two-layer neural networks over $\mathbb{R}$ with relu activation and unbounded width, is was defined in eq. (3.2.3). Conclude that the compactness assumption in definition 2 is crucial to define meaningful approximation results.

Exercise 5. By following the same steps as in the proof of lemma 1, prove that that $\mathcal{F}_{\text{exp},d}$ are universal approximators over $[0,1]^d$

Exercise 6. (a) Consider a Gaussian density with variance σ^2:

$$f(\boldsymbol{x}) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{1}{2\sigma^2}||\boldsymbol{x}||_2^2} \tag{3.4.7}$$

Show that its Barron norm is given by:

$$||f||_B = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{d+1}{2})}{\Gamma(\frac{d}{2})} \frac{1}{(2\pi\sigma^2)^{\frac{d+1}{2}}} \tag{3.4.8}$$

Conclude that for $\sigma^2 \geq 1$, we have $||f||_B = O(\sqrt{d})$ as $d \to \infty$. What happens for $\sigma^2 < 1$?

(b) Show that for the ridge function $f(\boldsymbol{x}) = \sigma(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b)$ the Barron norm is bounded by:

$$||f||_B \leq ||\boldsymbol{w}||_2 \int_{\mathbb{R}} |\xi \hat{\sigma}(\xi)| \mathrm{d}\xi. \tag{3.4.9}$$

Conclude that the ridge function with sigmoid-like activation are in $\mathcal{F}_B$, but not with ReLU activation $\sigma(x) = x_+$.

Chapter 4

Neural networks close to initialisation

4.1 Motivation

In the previous lectures, we have mostly focused our attention of the approximation question: given a hypothesis class, how well the the best predictor in the class can approximate the Bayes risk? The key take away from this discussion is that neural networks are pretty good in approximation. Even the simplest two-layer neural network is able to *uniformly* approximate an arbitrary *continuous function* on a *compact set* provided the *width is large enough*. In the worst case, "large enough" means exponentially in the input dimension. Nevertheless, we saw that this worst case result can be considerable improved if the target function is more regular, with the width scaling proportionally to a regularity measure (a.k.a. the Barron norm).

While this result is satisfying in many ways, it is only one piece of the puzzle. Sure, the best choice of first and second layer weights does the approximation job, but in the worst case finding these weights with an actual algorithm is a known NP-hard problem (Blum and Rivest, 1988). Nevertheless, training large neural networks with SGD on limited data seem to yield good generalisation performance. How come?

As we saw in Lecture 1, the approximation error is only one component of the generalisation gap:

$$R(\hat{\theta}) - R_\star = \underbrace{\left\{R(\hat{\theta}) - \inf_{\theta\in\Theta} R(\theta)\right\}}_{\text{estimation}} + \underbrace{\left\{\inf_{\theta\in\Theta} R(\theta) - R_\star\right\}}_{\text{approximation}}. \tag{4.1.1}$$

with the remaining part, the *estimation error*, accounting for how far our predictor is from the best predictor in the class. Understanding the estimation error involves both statistical and algorithmic aspects, and only recently we started developing a good understanding of this term for the case of two-layer neural trained with descent-based algorithms such as GD and SGD. In next few lectures, we will review some of this progress, starting from the influential result that in some regime, wide neural networks are equivalent to kernel methods.

4.2 The neural tangent kernel

Understanding the training dynamics of two-layer networks requires understanding how the weights move under a certain algorithm. Therefore, the starting point is initialisation.

	$\boldsymbol{a} \in \mathbb{R}^p$	$\boldsymbol{W} \in \mathbb{R}^{p \times d}$	$\boldsymbol{b} \in \mathbb{R}^p$
Kaiming (PYTORCH)	$a_j \underset{i.i.d.}{\sim} \text{Unif}\left(\left[-\sqrt{\frac{6}{p}}, \sqrt{\frac{6}{p}}\right]\right)$	$w_{jk} \underset{i.i.d.}{\sim} \text{Unif}\left(\left[-\sqrt{\frac{6}{d}}, \sqrt{\frac{6}{d}}\right]\right)$	$\boldsymbol{0}$
Xavier (TENSORFLOW)	$a_j \underset{i.i.d.}{\sim} \text{Unif}\left(\left[-\sqrt{\frac{6}{p+1}}, \sqrt{\frac{6}{p+1}}\right]\right)$	$w_{jk} \underset{i.i.d.}{\sim} \text{Unif}\left(\left[-\sqrt{\frac{6}{p+d}}, \sqrt{\frac{6}{p+d}}\right]\right)$	$\boldsymbol{0}$

Table 4.1: Default initialisation for PYTORCH and TENSORFLOW, also known the the Kaiming-Hu and the Xavier-Glorot initialisations, respectively.

4.2.1 Wide networks at initialisation

Consider a two-layer neural network with activation σ and width p:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_j) \tag{4.2.1}$$

In a standard PYTHON frameworks such as PYTORCH or TENSORFLOW, the weights are typically initialised at random from a uniform distribution at a interval depending on the width and the dimension, see table 4.1. It is curious to note that irrespective of your choice, the interval of the second layer weights shrinks as the width grows, in particular, the standard deviation vanishes as:

$$\text{std}(a_j^0) = O(1/\sqrt{p}) \text{ as } p \to \infty \tag{4.2.2}$$

The discussion above motivates the following rewriting at initialisation:

$$f(\boldsymbol{x};\boldsymbol{\theta}^0) = \frac{1}{\sqrt{p}} \sum_{j=1}^{p} \tilde{a}_j^0 \sigma(\langle \boldsymbol{w}_j^0, \boldsymbol{x}\rangle) \tag{4.2.3}$$

where $\tilde{a}_j = O(1)$. For reasons that we will see next, the $1/\sqrt{p}$ scaling is also known as the *NTK scaling*, and can be understood as the right scaling in order to make the $f(\boldsymbol{x};\boldsymbol{\theta}^0) = O(1)$ at initialisation. Indeed, $f(\boldsymbol{x};\boldsymbol{\theta}^0)$ is given by a sum of p independent random variables with $O(1)$ variance.

This is precisely the scaling of the central limit theorem! Indeed, assuming $\boldsymbol{x} \in \mathbb{R}^d$ is fixed, the above is a sum of independent random variables with standard deviation $1/\sqrt{p}$ - meaning that in the limit $p \to \infty$ the network at initialisation will converge to a Gaussian function with mean zero and variance given by:

$$K_{\text{GP}}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \mathbb{E}[f(\boldsymbol{x};\boldsymbol{\theta}^0) f(\boldsymbol{x}';\boldsymbol{\theta}^0)] = \mathbb{E}_{a,\boldsymbol{w}}\left[a^2 \sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\sigma(\langle \boldsymbol{w}, \boldsymbol{x}'\rangle)\right] \tag{4.2.4}$$

This is known as a *Gaussian process*, and was first derived in (Neal, 1996). In other words:

$$f(\cdot;\boldsymbol{\theta}^0) \xrightarrow{d} \mathcal{GP}\left(0, K_{\text{GP}}(\cdot,\cdot)\right) \tag{4.2.5}$$

⚠ Note that the $1/\sqrt{p}$ CLT scaling in eq. (4.2.3) is different from the $1/p$ scaling assumed in the discussion of Barron spaces in last lecture. As we will see later, this has important consequences for the space of functions that can be expressed by networks in this scaling.

4.2.2 Wide networks close to initialisation

At initialisation, a wide neural network with standard scaling is a Gaussian random function. What happens as we move away from initialisation? Before looking at particular training algorithms, let's look at how our network looks *close* to initialisation. More concretely, let $\boldsymbol{\theta} \in B(\boldsymbol{\theta}^0, 1)$. By Taylor's theorem:

$$f(\boldsymbol{x};\boldsymbol{\theta}) \approx f(\boldsymbol{x};\boldsymbol{\theta}^0) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta} - \boldsymbol{\theta}^0\rangle \tag{4.2.6}$$

Therefore, to first order the network $f(\boldsymbol{x};\boldsymbol{\theta})$ can be seen as linear method with features $\varphi(\boldsymbol{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta})$ — also known as the *neural tangent features*. But how close is the predictor to being linear? This is quantified by the next order or remainder term:

$$f(\boldsymbol{x};\boldsymbol{\theta}) \approx f(\boldsymbol{x};\boldsymbol{\theta}^0) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta}-\boldsymbol{\theta}^0\rangle + \frac{1}{2}\langle \boldsymbol{\theta}-\boldsymbol{\theta}^0, \boldsymbol{H}(\boldsymbol{\theta}-\boldsymbol{\theta}^0) \qquad (4.2.7)$$

where $\boldsymbol{H} = \nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}^0)$ is the Hessian matrix. Therefore, defining the linearised network:

$$f_{\text{lin.}}(\boldsymbol{x};\boldsymbol{\theta}) = f(\boldsymbol{x};\boldsymbol{\theta}^0) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta}-\boldsymbol{\theta}^0\rangle \qquad (4.2.8)$$

we can see that the deviation of f from $f_{\text{lin.}}$ in uniform norm is quantified by the operator norm of the Hessian:

$$\begin{aligned} \sup_{\boldsymbol{\theta}\in B(\boldsymbol{\theta}^0,1)} |f(\boldsymbol{x};\boldsymbol{\theta}) - f_{\text{lin.}}(\boldsymbol{x};\boldsymbol{\theta})| &= \frac{1}{2}\sup_{\boldsymbol{\theta}\in B(\boldsymbol{\theta}^0,1)} \langle \boldsymbol{\theta}-\boldsymbol{\theta}^0, \boldsymbol{H}(\boldsymbol{\theta}-\boldsymbol{\theta}^0)\rangle & (4.2.9)\\ &= \sup_{||\boldsymbol{v}||_2=1} \langle \boldsymbol{v}, \boldsymbol{H}\boldsymbol{v}\rangle := \frac{1}{2}||\boldsymbol{H}||_{\mathsf{op}} & (4.2.10)\end{aligned}$$

Where the last equality follows from the fact that the quadratic form $f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle$ with $\boldsymbol{A} \succeq 0$ attains its maximum on $B(\boldsymbol{0},1)$ on the boundary. Up to now, our discussion is valid for an arbitrary parametric function $f(\boldsymbol{x};\boldsymbol{\theta})$. Let's now look at the case of the two-layer neural network under standard initialisation scaling:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\sqrt{p}}\sum_{j=1}^{p} a_j\sigma(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle + b_j) \qquad (4.2.11)$$

To simplify the exposition, we will consider both a_j and b_j to be fixed, and let $b_j = 0$ to lighten the notation. As you will show in exercise 9, the argument that follows carries over to the general case. Assuming σ is twice differentiable, the gradient and Hessian of the network with respect to $\boldsymbol{w}_j$ reads:

$$\begin{aligned} \nabla_{\boldsymbol{w}_j} f(\boldsymbol{x};\boldsymbol{\theta}) &= \frac{1}{\sqrt{p}} a_j\sigma'(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle)\boldsymbol{x} & (4.2.12)\\ H_{jk}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{w}_j}\nabla_{\boldsymbol{w}_k} f(\boldsymbol{x};\boldsymbol{\theta}) &= \frac{1}{\sqrt{p}} a_j\sigma''(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle)\delta_{ij}\boldsymbol{x}\boldsymbol{x}^\top & (4.2.13)\end{aligned}$$

Therefore, the Hessian at initialisation $\boldsymbol{H} \in \mathbb{R}^{pd\times pd}$ is a block-diagonal matrix:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{H}_2 & \cdots & \boldsymbol{0}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{H}_p \end{bmatrix} \qquad (4.2.14)$$

with $\boldsymbol{H}_j \in \mathbb{R}^{d\times d}$ for $j \in [p]$ given by a rank-one matrix:

$$\boldsymbol{H}_j = \frac{1}{\sqrt{p}} a_j^0\sigma''(\langle \boldsymbol{w}_j^0, \boldsymbol{x}\rangle)\boldsymbol{x}\boldsymbol{x}^\top \qquad (4.2.15)$$

The operator norm of each block is easy to compute:

$$||\boldsymbol{H}_j||_{\mathsf{op}} = \frac{1}{\sqrt{p}}|a_j^0| \cdot |\sigma''(\langle \boldsymbol{w}_j^0, \boldsymbol{x}\rangle)| \cdot ||\boldsymbol{x}||_2^2 \qquad (4.2.16)$$

Hence, the operator norm of the full Hessian at initialisation is given by:

$$||\boldsymbol{H}||_{\mathsf{op}} = \sup_{j\in[p]} ||\boldsymbol{H}_j||_{\mathsf{op}} = \sup_{j\in[p]} \left[\frac{1}{\sqrt{p}}|a_j^0| \cdot |\sigma''(\langle \boldsymbol{w}_j^0, \boldsymbol{x}\rangle)| \cdot ||\boldsymbol{x}||_2^2\right] \tag{4.2.17}$$

As discussed in Section 4.2.1, $a^0 = O(1)$ is typically initialised uniformly in a bounded interval, say $[-1, 1]$. Moreover, σ has often bounded second derivative $\sigma''(z) < M$, e.g. for sigmoid-like functions or ReLU. Therefore:

$$||\boldsymbol{H}||_{\mathsf{op}} \leq \frac{M||\boldsymbol{x}||_2^2}{\sqrt{p}} = O({}^1\!/\!\sqrt{p}) \text{ as } p \to \infty \tag{4.2.18}$$

Putting this together with eq. (4.2.9) gives the following remarkable result:

Proposition 7. Let $f(\boldsymbol{x};\boldsymbol{\theta})$ denote a two-layer neural network of width p and activation function σ:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\sqrt{p}}\sum_{j=1}^{p} a_j\sigma(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle) \tag{4.2.19}$$

Assume σ has bounded second derivative. Then, under standard initialisation on the second-layer weights $a^0 \sim \mathrm{Unif}([-m, m])$:

$$\sup_{\boldsymbol{\theta}\in B(0,1)} |f(\boldsymbol{x};\boldsymbol{\theta}) - f_{\mathrm{lin.}}(\boldsymbol{x};\theta)| = O({}^1\!/\!\sqrt{p}) \text{ as } p \to \infty. \tag{4.2.20}$$

This is a quite remarkable result: it tell us that wide two-layer neural networks with standard scaling are close to linear functions!

⚠ Note that while the operator norm of the Hessian vanishes with $p \to \infty$, that's not the case for $f(\boldsymbol{x};\boldsymbol{\theta})$ (as we saw in section 4.2.1) and for the gradient $\nabla_{\boldsymbol{\theta}} f$. Indeed, for any $j \in [p]$ we have:

$$\sum_{j=1}^{p} ||\nabla_{\boldsymbol{w}_j} f(\boldsymbol{x};\boldsymbol{\theta})||_2^2 = \frac{1}{p}\sum_{j=1}^{p} a_j^2\sigma'(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle)^2||\boldsymbol{x}||_2^2 = O(1). \tag{4.2.21}$$

Hence the linear approximation itself $f_{\mathrm{lin.}}$ is not a constant function — meaning the linear behaviour is actually non-trivial.

Remark 13. Some remarks are in order.

- This result first appeared in Liu et al. (2020a), who also proved a more general result for deep networks.
- Note that in our derivation, we did not use any information about the initialisation of the first layer weights $\boldsymbol{w}_j^0$. Indeed, this is not necessary in the two-layer case. However, it does play an important role in the proof of the deep case.
- Note that this argument breaks down if we add a non-linearity at the last-layer $g(\boldsymbol{x};\boldsymbol{\theta}) = \phi(f(\boldsymbol{x};\boldsymbol{\theta}))$ — see exercise 8 for a discussion.

4.2.3 Wide networks away from initialisation

We now discuss what happens as we move away from initialisation. For that, we consider a supervised learning problem with training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathcal{Y} : i \in [n]\}$. For simplicity of exposition, we will consider the case where the network is trained under gradient flow on the square loss:

$$\begin{aligned}\dot{\boldsymbol{\theta}}(t) &= -\nabla\hat{R}(\boldsymbol{\theta};\mathcal{D})\\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i;\boldsymbol{\theta}(t)))\,\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta}(t))\end{aligned} \tag{4.2.22}$$

where we denote $\dot{u} := {}^{\mathrm{d}u}/_{\mathrm{d}t}$. As the weights $\boldsymbol{\theta}(t)$ change, how does the predictor change? For that, we use the chain rule to write:

$$\frac{\mathrm{d}f(\boldsymbol{x};\boldsymbol{\theta})}{\mathrm{d}t} = \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x};\boldsymbol{\theta}), \dot{\boldsymbol{\theta}} \rangle \tag{4.2.23}$$

Inserting eq. (4.2.22) in the above:

$$\frac{\mathrm{d}f(\boldsymbol{x};\boldsymbol{\theta})}{\mathrm{d}t} = \frac{1}{n}\sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i;\boldsymbol{\theta})) \langle \nabla f(\boldsymbol{x};\boldsymbol{\theta}), \nabla f(\boldsymbol{x}_i;\boldsymbol{\theta}) \rangle \tag{4.2.24}$$

Defining the *empirical neural tangent kernel*:

$$\hat{K}_t(\boldsymbol{x},\boldsymbol{x}') = \langle \nabla f(\boldsymbol{x};\boldsymbol{\theta}(t)), \nabla f(\boldsymbol{x}';\boldsymbol{\theta}(t)) \rangle \tag{4.2.25}$$

We can rewrite:

$$\frac{\mathrm{d}f(\boldsymbol{x};\boldsymbol{\theta})}{\mathrm{d}t} = \frac{1}{n}\sum_{i=1}^{n} \hat{K}_t(\boldsymbol{x}_i,\boldsymbol{x}) (y_i - f(\boldsymbol{x}_i;\boldsymbol{\theta})) \tag{4.2.26}$$

In particular, if we are interested to track the prediction function over the training data, we can define a vector: $\boldsymbol{f} \in \mathbb{R}^n$ with elements $f_i := f(\boldsymbol{x}_i;\boldsymbol{\theta})$ and write the above in a matrix form:

$$\dot{\boldsymbol{f}}(t) = \frac{1}{n}\hat{\boldsymbol{K}}_t(\boldsymbol{y} - \boldsymbol{f}(t)) \tag{4.2.27}$$

where we defined the kernel matrix $\hat{\boldsymbol{K}}_t \in \mathbb{R}^{n\times n}$ with elements $\hat{K}_{t,ij} = \hat{K}_t(\boldsymbol{x}_i,\boldsymbol{x}_j)$.

Remark 14. A few comments are in order.

- The above derivation is valid for any parametric function $f(\boldsymbol{x};\boldsymbol{\theta})$. In particular, it is valid for networks with depth $L > 2$.
- Let $p = |\Theta|$ denote the total number of parameters. Then, if $\hat{\boldsymbol{K}}_t$ has rank n (which is possible if $p > n$, for instance), stationary points of the flow consist of interpolators $\boldsymbol{f} = \boldsymbol{y}$.
- Moreover, if at some time $t > T$ the empirical NTK becomes constant $\hat{K}_t \approx \hat{K}$, eq. (4.2.27) simply describes an exponential relaxation to the interpolator $\boldsymbol{f} = \boldsymbol{y}$, with the relaxation rate given by the largest eigenvalue of $\boldsymbol{K}$.

Now let's go back to our favourite model: the two-layer neural network. Motivated by the discussion in section 4.2.1, we will consider the NTK initialisation:

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\sqrt{p}}\sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle) \tag{4.2.28}$$

with Kaiming-like initialisation $a_j(0) \simeq \mathrm{Unif}([-1,1])$, $\boldsymbol{w}_j(0) \sim \mathcal{N}(\boldsymbol{0}, {}^1/_d\boldsymbol{I}_d)$ and for simplicity consider $b_j = 0$. Then, since $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{a})$, the neural tangent kernel in eq. (4.2.25) explicitly reads:

$$\hat{K}_t(\boldsymbol{x},\boldsymbol{x}') = \frac{1}{p}\sum_{j=1}^{p} \sigma(\langle \boldsymbol{w}_j(t), \boldsymbol{x} \rangle)\sigma(\langle \boldsymbol{w}_j(t), \boldsymbol{x}' \rangle) + \frac{\langle \boldsymbol{x},\boldsymbol{x}' \rangle}{p}\sum_{j=1}^{p} a_j^2 \sigma'(\langle \boldsymbol{w}_j(t), \boldsymbol{x} \rangle)\sigma'(\langle \boldsymbol{w}_j(t), \boldsymbol{x}' \rangle) \tag{4.2.29}$$

Just as in the argument we made in section 4.2.1, at initialisation $t = 0$ the empirical NTK kernel $\hat{K}_0$ can be seen as a discretisation of a limiting infinite width kernel. Indeed, by the law of large numbers we have

$$\hat{K}_0(\boldsymbol{x},\boldsymbol{x}') \xrightarrow{a.s.} K_0(\boldsymbol{x},\boldsymbol{x}') := K_{\mathrm{RF}}(\boldsymbol{x},\boldsymbol{x}') + K_{\mathrm{NTK}}(\boldsymbol{x},\boldsymbol{x}'), \text{ as } p \to \infty \tag{4.2.30}$$

where:

$$K_{\mathrm{RF}}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \mathbb{E}_{\boldsymbol{w}}\left[\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\sigma(\langle \boldsymbol{w}, \boldsymbol{x}'\rangle)\right]$$
$$K_{\mathrm{NTK}}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \langle \boldsymbol{x}, \boldsymbol{x}'\rangle \mathbb{E}_{a,\boldsymbol{w}}\left[a^2\sigma'(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\sigma'(\langle \boldsymbol{w}, \boldsymbol{x}'\rangle)\right] \tag{4.2.31}$$

The kernel K_{RF} is known as the *random features* kernel. Some authors refer to the full K_0 as the *neural tangent kernel* (NTK), while others only the part proportional to the derivative of the activation K_{NTK}.

This argument only applies to initialisation $t = 0$ at this point, and the choice of initial scaling $1/\sqrt{p}$ is crucial. In particular, note that K_0 is quite different from the Gaussian process kernel defined by the wide limit of initialisation in eq. (4.2.4). We are now interested in understanding how much does $f(\boldsymbol{x}, \boldsymbol{\theta}(t))$ deviates from $f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t))$ along the flow eq. (4.2.26). For that, define:

$$E(t) = \sup_{\boldsymbol{x}} |f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t))| \tag{4.2.32}$$

and note that at $t = 0$, we have $E(0) = 0$ if $\boldsymbol{\theta}(0) = \boldsymbol{\theta}^0$. Then, we have:

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} \leq \sup_{\boldsymbol{x}} \left|\frac{\mathrm{d}}{\mathrm{d}t}\left[f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t))\right]\right| \tag{4.2.33}$$

Therefore, we need to control:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left[f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t))\right] &= \left\langle \nabla f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - \nabla f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t)), \dot{\boldsymbol{\theta}}(t)\right\rangle \\ &= \left\langle \nabla f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - \nabla f(\boldsymbol{x}, \boldsymbol{\theta}^0), \dot{\boldsymbol{\theta}}(t)\right\rangle\end{aligned} \tag{4.2.34}$$

since $\nabla f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t)) = \nabla f(\boldsymbol{x}, \boldsymbol{\theta}^0)$. Hence, by Cauchy-Schwarz:

$$\left|\frac{\mathrm{d}}{\mathrm{d}t}\left[f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t))\right]\right| \leq ||\nabla f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - \nabla f(\boldsymbol{x}, \boldsymbol{\theta}^0)||_2 \cdot ||\dot{\boldsymbol{\theta}}(t)||_2 \tag{4.2.35}$$

By the mean value theorem, for any $u \in [0, 1]$:

$$||\nabla f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - \nabla f(\boldsymbol{x}, \boldsymbol{\theta}^0)|| \leq ||\nabla^2 f(\boldsymbol{x}, u\boldsymbol{\theta}^0 + (1-u)\boldsymbol{\theta}(t))||_{\mathrm{op}} \cdot ||\boldsymbol{\theta}(t) - \boldsymbol{\theta}^0||_2 \tag{4.2.36}$$

Therefore, the main ingredient to control $E(t)$ is to ensure that the velocity is bounded $||\boldsymbol{\theta}(t)||_2 \leq V$. Indeed, assuming this is the case implies, by continuity, that $\boldsymbol{\theta}(t)$ cannot move away from $\boldsymbol{\theta}^0$ by a fixed distance for a given time horizon:

$$||\boldsymbol{\theta}(t) - \boldsymbol{\theta}^0||_2 = \left|\left|\int_0^t \dot{\boldsymbol{\theta}}(s)\mathrm{d}s\right|\right|_2 \leq \int_0^t ||\dot{\boldsymbol{\theta}}(s)||\mathrm{d}s = Vt \tag{4.2.37}$$

In other words, for any time horizon $t \in [0, R/V)$, we have $||\boldsymbol{\theta}(t) - \boldsymbol{\theta}^0||_2 \leq R$. Together with our bound on the Hessian eq. (4.2.18) this implies that in this time horizon:

$$\left|\frac{\mathrm{d}}{\mathrm{d}t}\left[f(\boldsymbol{x}, \boldsymbol{\theta}(t)) - f_{\mathrm{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t))\right]\right| \leq \frac{VM||\boldsymbol{x}||_2}{\sqrt{p}}||\boldsymbol{\theta}(t) - \boldsymbol{\theta}^0||_2 \tag{4.2.38}$$

Or in other words:

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} \leq \frac{Vc}{\sqrt{p}}||\boldsymbol{\theta}(t) - \boldsymbol{\theta}^0||_2 \leq \frac{RVc}{\sqrt{p}} \tag{4.2.39}$$

for some other constant $c > 0$. By Gronwall's inequality:

$$E(t) \leq \underbrace{E(0)}_{=0} + \frac{RVc}{\sqrt{p}} t \tag{4.2.40}$$

Therefore, as long as $t \in [0, {}^{R}\!/\!{}_{V}]$, we have $E(t) = O({}^{1}\!/\!{}_{\sqrt{p}})$ and $f(\boldsymbol{x}, \boldsymbol{\theta}(t)) \approx f_{\text{lin.}}(\boldsymbol{x}, \boldsymbol{\theta}(t)))$. It remains just to justify why the velocity is bounded along the flow. This is rather intuitive. Indeed, recall that under the flow:

$$\dot{\boldsymbol{f}}(t) = \frac{1}{n} \hat{\boldsymbol{K}}_t (\boldsymbol{y} - \boldsymbol{f}(t)) \tag{4.2.41}$$

where $\boldsymbol{K}_t \in \mathbb{R}^{n \times n}$ is the matrix with entries $\hat{K}_{t,ij} = \langle \nabla f(\boldsymbol{x}_i; \boldsymbol{\theta}(t)), \nabla f(\boldsymbol{x}_j; \boldsymbol{\theta}(t)) \rangle$. Since in the interval $t \in [0, {}^{R}\!/\!{}_{V}]$ we have $f \approx f_{\text{lin.}}$, then $\hat{K}_t \approx \hat{K}_0$ since the NTK for the linear model is constant. This means that eq. (4.2.27) is simply an exponential relaxation. In particular, the speed is controlled by the minimum eigenvalue of the NTK gram matrix:

$$||\boldsymbol{y} - \boldsymbol{f}(t)||_2 \leq e^{-\lambda_{\min}(\hat{\boldsymbol{K}}_0)t} ||\boldsymbol{y} - \boldsymbol{f}(0)||_2 \tag{4.2.42}$$

Therefore, as long as the minimum eigenvalue of $\hat{\boldsymbol{K}}_0$ is non-zero (i.e. $\hat{\boldsymbol{K}}_0 \succ 0$) — which can be proven for specific activations and distributions of initial conditions $\boldsymbol{\theta}^0$, see Exercise 7 — then gradient flow relaxes exponentially fast, meaning that the velocity $\dot{\boldsymbol{\theta}}$ remains bounded and that t never exceeds ${}^{R}\!/\!{}_{V}$, which closes the argument. This can be formalised in the following theorem, which appeared in Du et al. (2019):

Theorem 7 (Du et al. (2019))**.** Assume $\lambda_0 = \lambda_{\min}(\boldsymbol{K}_0) > 0$. Then, for $p \geq O({}^{n^6}\!/\!{}_{\lambda_0 \delta^3})$, with probability at least $1 - \delta$ we have:

$$||\boldsymbol{y} - \boldsymbol{f}(t)||_2 \leq e^{-\lambda_0 t} ||\boldsymbol{y} - \boldsymbol{f}(0)||_2 \tag{4.2.43}$$

along the gradient flow in eq. (4.2.41).

Remark 15 (Generalisations)**.** In this section, we discussed the NTK in the specific case of wide two-layer neural networks trained under gradient flow. However, the results discussed hold on a much more general scope. First, it can be generalise to deeper networks (Jacot et al., 2018; Lee et al., 2019) and other architectures, such as convolutional networks and transformers (Arora et al., 2019; Yang, 2020). Similarly, it can be generalised to other algorithms, such as gradient descent and stochastic gradient descent.

Epilogue

Together, the results in this section tell us that in the standard initialisation from Table 4.1 (a.k.a. *NTK scaling*), infinite width networks are equivalent to a particular class of kernel methods, where the so-called neural tangent kernels are functions of the architecture at initialisation. This result is striking, and suggests that to understand deep networks, it suffices to understand their NTK at initialisation. But don't be fooled: there is actually a reason why neural networks are used in practice, instead of kernels, and you should regard this result with a pinch of suspicion.

As we discussed in Lecture 1, one of the reasons behind the practical success of neural networks is their capacity to adapt to the data during training, learning relevant features which can help solving particular tasks. On the other hand, despite being powerful, kernels are not adaptive, and generally require a large amount of data to learn functions that lack regularity. This strongly suggests that the NTK cannot be the end of the story.

4.3 Lazy regime in optimisation

The key step in the argument of Section 4.2.3 is to show that the loss is minimised in timescales which are much shorter than the time it takes for the parameters to move away from a neighbourhood of initialisation. This behaviour, known as the *lazy regime* of wide neural networks, is more general than the particular context of two-layer neural networks. Indeed, a close inspection reveals that the crucial ingredient in the argument is the scaling of parameters at initialisation. This was noted by Chizat et al. (2019), and we closely follow their argument here.

To formalise this, let's consider a generic parametric model $h : \mathbb{R}^m \to \mathcal{H}$ and cost function $L : \mathcal{H} \to \mathbb{R}_+$, where $\mathcal{H}$ is a Hilbert space. In the following, we will assume both h and F are smooth. We are interested in the following optimation objective $F : \mathbb{R}^m \to \mathbb{R}_+$:

$$F(\boldsymbol{\theta}) \coloneqq L(h(\boldsymbol{\theta})). \tag{4.3.1}$$

Let $\boldsymbol{\theta}^0 \in \mathbb{R}^m$ denote an initial parameter, which we assume is not a critical point of the objective, i.e. $\nabla F(\boldsymbol{\theta}) \neq 0$. Then, under a single step of gradient descent with learning rate $\eta > 0$, we have:

$$\boldsymbol{\theta}^1 = \boldsymbol{\theta}^0 - \eta \nabla F(\boldsymbol{\theta}). \tag{4.3.2}$$

Define the relative change of the objective function with respect to the gradient step:

$$\Delta(F) \coloneqq \frac{|F(\boldsymbol{\theta}^1) - F(\boldsymbol{\theta}^0)|}{F(\boldsymbol{\theta}^0)}. \tag{4.3.3}$$

For small enough step-size η, this is effectively controlled by the gradient relative to the initial value of the loss:

$$\Delta(F) = \eta \frac{||\nabla F(\boldsymbol{\theta}^0)||_2}{F(\boldsymbol{\theta}^0)} + o(\eta) \tag{4.3.4}$$

On the other hand, as we have discussed in eq. (4.2.9), the rate of change of the features is controlled by the Hessian at initialisation:

$$\Delta(\nabla h) \coloneqq \frac{||\nabla h(\boldsymbol{\theta}^1) - \nabla h(\boldsymbol{\theta}^0)||_2}{||\nabla h(\boldsymbol{\theta}^0)||_2} \leq \eta \frac{||\nabla F(\boldsymbol{\theta})||_2 \cdot ||\nabla^2 h(\boldsymbol{\theta})||_{\text{op}}}{||\nabla h(\boldsymbol{\theta}^0)||_2} \tag{4.3.5}$$

We say that our model is lazy whenever $\Delta(F) \gg \Delta(\nabla h)$, in other words:

$$\frac{||\nabla F(\boldsymbol{\theta}^0)||_2}{F(\boldsymbol{\theta}^0)} \gg \frac{||\nabla^2 h(\boldsymbol{\theta}^0)||_{\text{op}}}{||\nabla h(\boldsymbol{\theta}^0)||_2} \tag{4.3.6}$$

For the square loss function where $R(h(\boldsymbol{\theta})) = {}^1\!/_2||y - h(\boldsymbol{\theta})||^2_{\mathcal{H}}$, this is equivalent to:

$$\kappa_h(\boldsymbol{\theta}_0) \coloneqq ||y - h(\boldsymbol{\theta}^0)||_{\mathcal{H}} \frac{||\nabla^2 h(\boldsymbol{\theta}^0)||_{\text{op}}}{||\nabla h(\boldsymbol{\theta}^0)||_2^2} \ll 1 \tag{4.3.7}$$

where κ_h is the inverse relative scale of the model h at initialisation. Whenever this is small, the relative change in the loss is faster than the change in the features, meaning that the timescale for the loss to be minimised will be faster than the timescale for the model to exit a linear behaviour, leading to a lazy regime. For an extensive discussion of the lazy regime in different optimisation problems, see Chizat et al. (2019).

Before concluding, it is instructive to go look back at out discussion in Section 4.2 through the lens of lazy training. Consider our usual two-layer neural network, now introducing a generic scaling factor $\alpha(p)$:

$$h(\boldsymbol{\theta}) = f(\boldsymbol{x}; \boldsymbol{\theta}) = \alpha(p) \sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle) \tag{4.3.8}$$

For any initialisation where the weights $a_j^0, \boldsymbol{w}_j^0$ are drawn independently and $O(1)$ we can repeat the discussion in Section 4.2 to show that asymptotically in $p \to \infty$:

$$\begin{aligned}|y - h(\boldsymbol{\theta}^0)| &= O(1)\\ ||\nabla^2 h(\boldsymbol{\theta}^0)||_{\mathtt{op}} &\sim \alpha(p)\\ ||\nabla h(\boldsymbol{\theta}^0)||_2^2 &\sim p\alpha(p)^2\end{aligned} \tag{4.3.9}$$

and hence:

$$\kappa_h(\boldsymbol{\theta}^0) \sim \frac{1}{\sqrt{p}} + \frac{1}{p\alpha(p)} \tag{4.3.10}$$

For any $\alpha(p)$ decreasing slower than $1/\sqrt{p}$, we indeed have $\kappa_h(p) \to 0$ as $p \to \infty$, leading to a lazy behaviour. The critical scaling to exit the lazy regime is given by $\alpha(p) = O(1/p)$, known in the machine learning literature as *mean-field scaling.*[1] Note this is precisely the scaling of the two-layer neural network in Barron's theorem from Lecture 3, and for which a large width two-layer neural network can be represented as an integral over a probability measure μ over $\mathbb{R}^{p+1}$:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{p}\sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle) \approx \int \mu(\mathrm{d}a, \mathrm{d}\boldsymbol{w}) a\sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle), \qquad \text{as } p \to \infty. \tag{4.3.11}$$

In fact, this is the scaling that maximises the relative change of the features with respect to the loss — or in other words, maximises *feature learning.* In this *feature rich* regime, the dynamics of the network is far from being a simple exponential relaxation to the bottom of a minimum. Indeed, in this regime the gradient flow for infinite width neural networks can be written in terms of a flow for the measure μ_t:

$$\partial_t \mu_t = \nabla_{\boldsymbol{\theta}} \cdot \left(\mu_t \nabla_{\boldsymbol{\theta}} \hat{R}_n(\mu_t)\right) \tag{4.3.12}$$

which is simply a continuity or transport equation for the density μ_t, see (Mei et al., 2018; Chizat and Bach, 2018; Rotskoff and Vanden-Eijnden, 2022; Sirignano and Spiliopoulos, 2020) for a detailed discussion. Analysing this flow is a mathematically much more challenging problem than the NTK gradient flow eq. (4.2.41) with $\hat{\boldsymbol{K}}_t \approx \hat{\boldsymbol{K}}_0$. In particular, precisely characterising the stationary measure is a hard problem.

Remark 16. While the limiting eq. (4.3.12) is only well-understood for shallow networks, the equivalent of the mean-field scaling for deep neural networks is known as the μ-parametrisation (μP), see Yang et al. (2022) for a discussion.

4.4 Exercises

Exercise 7. Consider a two-layer neural network with ReLU activation $\sigma(x) = x_+$:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{\sqrt{p}}\sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}, \boldsymbol{x}\rangle). \tag{4.4.1}$$

Assume that the weights are initialised as $a_j^0 \sim \mathrm{Unif}(\{-1, 1\})$, $\boldsymbol{w}^0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$.

(a) Show that the NTK kernel is given by:

$$\begin{aligned}K_{\mathrm{NTK}}(\boldsymbol{x}, \boldsymbol{x}') &:= \langle \boldsymbol{x}, \boldsymbol{x}'\rangle \mathbb{E}_{a,\boldsymbol{w}}\left[a^2 \sigma'(\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\sigma'(\langle \boldsymbol{w}, \boldsymbol{x}'\rangle)\right]\\ &= \langle \boldsymbol{x}, \boldsymbol{x}'\rangle\left[\frac{1}{2} - \frac{1}{2\pi}\arccos\left(\frac{\langle \boldsymbol{x}, \boldsymbol{x}'\rangle}{||\boldsymbol{x}||_2 \cdot ||\boldsymbol{x}'||_2}\right)\right]\end{aligned} \tag{4.4.2}$$

[1] Note that the word *mean-field* is used with different meanings in other contexts in Physics.

(b) Let $\boldsymbol{x}_i \in \mathbb{R}^d$ denote a batch of n independently sampled covariates, and assume $\boldsymbol{x}_i \in B(\mathbf{0}, 1)$. Using Hoeffding's inequality, show that if $p \geq \Omega(\epsilon^{-2} n^2 \log {}^{n}/_{\delta})$, then with probability at least $1 - \delta$ over the random initialisation we have:

$$||\hat{\boldsymbol{K}}_{\mathrm{NTK}} - \boldsymbol{K}_{\mathrm{NTK}}||_{\mathrm{F}} \leq \epsilon \tag{4.4.3}$$

where $\hat{\boldsymbol{K}}_{\mathrm{NTK}}, \boldsymbol{K}_{\mathrm{NTK}} \in \mathbb{R}^{n \times n}$ with:

$$\begin{aligned} \hat{\boldsymbol{K}}_{\mathrm{NTK},ij} &= \frac{\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle}{p} \sum_{k=1}^{p} a_k^2 \sigma'(\langle \boldsymbol{w}_k, \boldsymbol{x}_i \rangle \sigma'(\langle \boldsymbol{w}_k, \boldsymbol{x}_j \rangle), \\ \boldsymbol{K}_{\mathrm{NTK},ij} &= K_{\mathrm{NTK}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \end{aligned} \tag{4.4.4}$$

(c) Conclude that for large enough width p, we have that $\lambda_{\min}(\hat{\boldsymbol{K}}_{\mathrm{NTK}}) > 0$ with high-probability.

Exercise 8. Let $g(\boldsymbol{x}; \boldsymbol{\theta}) = \phi(f(\boldsymbol{x}; \boldsymbol{\theta}))$ where ϕ is a twice differentiable function and $f(\boldsymbol{x}; \boldsymbol{\theta})$ a two-layer neural network:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{\sqrt{p}} \sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle) \tag{4.4.5}$$

with twice-differentiable activation function σ.

(a) Considering a_j to be fixed, show that for any $\boldsymbol{\theta}$, the Hessian matrix $\boldsymbol{H}_g$ of g can be related to the Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta})$ of f by:

$$\boldsymbol{H}_g(\boldsymbol{\theta}) = \phi'(f(\boldsymbol{x}; \boldsymbol{\theta}))\boldsymbol{H}(\boldsymbol{\theta}) + \phi''(f(\boldsymbol{x}; \boldsymbol{\theta}))\nabla_{\boldsymbol{w}} f(\boldsymbol{x}; \boldsymbol{\theta}) \nabla_{\boldsymbol{w}} f(\boldsymbol{x}; \boldsymbol{\theta})^\top \tag{4.4.6}$$

(b) Under standard initialisation $a^0 \sim \mathrm{Unif}([-1, 1])$, what is the scaling in p of the operator norm of each of the terms above?

(c) Conclude that $g(\boldsymbol{x}; \boldsymbol{\theta})$ does not linearise as $p \to \infty$.

Exercise 9. Generalise the argument leading to Proposition 1 to the case where the second layer weights a_j are not fixed.

Chapter 5

Introduction to the asymptotic analysis of random matrices

5.1 Motivation

Random matrix theory is the area of mathematics concerned with the properties of matrices with random entries. This is a vast field, and here we will deliberately limit the discussion to the aspects which are relevant to asymptotic analysis of ridge regression.

We have seen that the analysis of the risk of ridge regression boils down to the investigation of the following bias and variance terms:

$$B(\boldsymbol{\theta}_\star, \boldsymbol{X}, \lambda) = \lambda^2 \operatorname{Tr}\left\{\boldsymbol{\theta}_\star\boldsymbol{\theta}_\star^\top \left(\hat{\boldsymbol{\Sigma}}_n + \lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\Sigma} \left(\hat{\boldsymbol{\Sigma}}_n + \lambda \boldsymbol{I}_d\right)^{-1}\right\} \tag{5.1.1}$$

$$V(\boldsymbol{X}, \lambda, \sigma^2) = \frac{\sigma^2}{n} \operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Sigma}}_n + \lambda \boldsymbol{I}_d\right)^{-1}\right\} - \frac{\lambda\sigma^2}{n} \operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Sigma}}_n + \lambda \boldsymbol{I}_d\right)^{-2}\right\} \tag{5.1.2}$$

where $\hat{\boldsymbol{\Sigma}}_n = {}^1\!/\!{}_n \boldsymbol{X}^\top \boldsymbol{X}$ is the sample covariance matrix and we assumed the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ has i.i.d. rows $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. The expressions above take one of the following two forms:

$$\operatorname{Tr}\{\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{M}}(z)\}, \qquad \operatorname{Tr}\{\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{M}}(z)\boldsymbol{B}\boldsymbol{R}_{\boldsymbol{M}}(z)\} \tag{5.1.3}$$

where $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$ are deterministic matrices and $\boldsymbol{R}(z) \in \mathbb{R}^{d\times d}$ is the resolvent matrix associated to the symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{d\times d}$:

$$\boldsymbol{R}_{\boldsymbol{M}}(z) = (\boldsymbol{M} - z\boldsymbol{I}_d)^{-1}, \qquad z \in \mathbb{C} \setminus \operatorname{spec}(\boldsymbol{M}) \tag{5.1.4}$$

Note the resolvent is a symmetric matrix defined everywhere in the complex plane, except at the values in the real axis where the eigenvalues of $\boldsymbol{M}$ lie. It contains useful information on both the spectrum of $\boldsymbol{M}$ and its eigenvectors, since it can be diagonalised on the same basis of $\boldsymbol{M}$:

$$\boldsymbol{R}_{\boldsymbol{M}}(z) = \sum_{i=1}^{d} \frac{\boldsymbol{u}_i\boldsymbol{u}_i^\top}{\lambda_i - z} \tag{5.1.5}$$

where $\boldsymbol{u}_i$ are the eigenvectors of $\boldsymbol{M}$. Therefore, by the Cauchy integral formula we have:

$$f(\lambda_j) = -\int_{\gamma_j} \frac{\mathrm{d}z}{2\pi i} f(z)\boldsymbol{R}_{\boldsymbol{M}}(z), \qquad \boldsymbol{u}_j\boldsymbol{u}_j^\top = -\int_{\gamma_j} \frac{\mathrm{d}z}{2\pi i}\boldsymbol{R}_{\boldsymbol{M}}(z) \tag{5.1.6}$$

where γ_j is a contour circling the eigenvalue λ_j and f is any analytic function. Therefore, the traces in eq. (5.1.3) can be seen as extracting some statistics about eigenvalues or eigenvectors of $\boldsymbol{M}$ through the resolvent.

Our goal in this lecture is to introduce the main concepts and tools used to study the asymptotic behaviour of these quantities when $d \to \infty$.

5.2 Main notions

Let $\boldsymbol{M} \in \mathbb{R}^{d\times d}$ denote a symmetric random matrix and $\operatorname{spec}(\boldsymbol{M}) = \{\lambda_1, \dots, \lambda_d\} \subset \mathbb{R}$ its real eigenvalues. These are random variables, and one of the central goals of random matrix theory is to characterise their statistics. Studying the statistics of single eigenvalues can be challenging, so instead we turn to studying large aggregates of eigenvalues.

Definition 4 (Empirical spectral measure)**.** Let $\boldsymbol{M} \in \mathbb{R}^{d\times d}$ denote a symmetric matrix with eigenvalues $\operatorname{spec}(\boldsymbol{M}) = \{\lambda_1, \dots, \lambda_d\} \subset \mathbb{R}$. The empirical spectral measure of $\boldsymbol{M}$ is defined as:

$$\hat{\mu}_{\boldsymbol{M}} = \frac{1}{d}\sum_{i=1}^{d} \delta_{\lambda_i} \tag{5.2.1}$$

Note that $\int \hat{\mu}_{\boldsymbol{M}}(\mathrm{d}x) = 1$ and therefore $\hat{\mu}_{\boldsymbol{M}}$ is also probability measure.

Remark 17. The empirical spectral measure is the normalised counting measure of how many eigenvalues lie in an interval:

$$\hat{\mu}_{\boldsymbol{M}}([a,b]) = \frac{1}{d}\{\# \text{ eigenvalues } \lambda_i \in [a,b]\}. \tag{5.2.2}$$

Since $\hat{\mu}$ is a measure, studying its limiting behaviour is mathematically tricky. Instead, it is convenient to study its *Stieltjes transform*:

Definition 5 (Stieltjes transform)**.** Let μ denote a probability measure over $\mathbb{R}$. The *Stieltjes transform* of μ is defined as:

$$s_\mu(z) = \int \frac{\mu(\mathrm{d}\lambda)}{\lambda - z}, \qquad z \in \mathbb{C} \setminus \operatorname{supp}(\mu) \tag{5.2.3}$$

The Stieltjes transform can be thought as an "integrated version" of the measure. It satisfies some useful properties:

Properties 1. The Stieltjes transform satisfies the following properties:

(a) **Boundness:** For all $z \in \mathbb{C} \setminus \operatorname{supp}(\mu)$:

$$|s_\mu(z)| \leq \frac{1}{\operatorname{dist}(z, \operatorname{supp}(\mu))} \leq \frac{1}{|\operatorname{Im}(z)|} \tag{5.2.4}$$

This is straightforward to show. Letting $z = x + i\eta$ with $\eta > 0$:

$$|s_\mu(z)| \leq \int \mu_{\boldsymbol{M}}(\mathrm{d}\lambda)\left|\frac{1}{\lambda - z}\right| = \int \mu_{\boldsymbol{M}}(\mathrm{d}\lambda)\frac{1}{\sqrt{(\lambda-x)^2+\eta^2}} \leq \int \mu_{\boldsymbol{M}}(\mathrm{d}\lambda)\frac{1}{\eta} = \frac{1}{\eta} \tag{5.2.5}$$

(b) **Sign:** We have that $\operatorname{Im}(z)\operatorname{Im}(s_\mu(z)) \geq 0$ since:

$$\eta \operatorname{Im}[s_\mu(x+i\eta)] = \eta\int \mu(\mathrm{d}\lambda)\operatorname{Im}\left[\frac{1}{\lambda-(x+i\eta)}\right] = \frac{1}{\pi}\int \mu(\mathrm{d}\lambda)\frac{\eta^2}{(\lambda-x)^2+\eta^2} \geq 0 \tag{5.2.6}$$

(c) **Derivatives:** The Stieltjes transform is an analytic function on $\mathbb{C}\setminus\operatorname{supp}(\mu)$. In particular, it is infinitely differentiable, and the derivatives can be similarly bounded:

$$|s^{(k)}(z)| \leq \frac{(k-1)!}{|\operatorname{Im}(z)|^{k+1}} \tag{5.2.7}$$

In particular, this implies $s_\mu(x)$ is an increasing function on all connected components of $x \in \mathbb{R}\setminus\operatorname{supp}(\mu)$ since $s'_\mu(x) > 0$.

(d) **Moments:** Suppose $\mathrm{supp}(\mu) = [-M, M]$ has bounded support. Then, for any $|z| > M$ we can Taylor expand:

$$s_\mu(z) = \int \frac{\mu(\mathrm{d}\lambda)}{\lambda - z} = -\frac{1}{z} \int \left(1 - \frac{\lambda}{z}\right)^{-1} \mu(\mathrm{d}\lambda) = -\frac{1}{z} \int \mu(\mathrm{d}\lambda) \sum_{k=0}^{\infty} \left(\frac{\lambda}{z}\right)^k$$
$$= -\sum_{k=0}^{\infty} z^{-(k+1)} \mathbb{E}_{X\sim\mu}[X^k] \tag{5.2.8}$$

Therefore, the moments $\mathbb{E}_{X\sim\mu}[X^k]$ of the distribution μ are given by the coefficients of the Taylor expansion of $\tilde{s}_\mu(u) = s_\mu(1/z)$. This means the Stieltjes transform can also be seen as a moment generating function for μ.

(e) **Large z asymptotics:** As a corollary of the previous point, we have:

$$s_\mu(z) = -\frac{1}{z} + O(|z|^{-2}), \quad \text{as } |z| \to \infty \tag{5.2.9}$$

Since the moments of μ can be computed from s_μ, property (d) suggests that we can fully reconstruct the probability measure from the Stieltjes transform. This intuition is indeed the case, and is formalised by the *Stieltjes-Perron inversion formula*:

Proposition 8 (Stieltjes-Perron inversion formula)**.** Let a, b denote continuity points of the probability measure μ. Then:

$$\mu([a,b]) = \frac{1}{\pi} \lim_{\eta\to 0^+} \int_a^b \mathrm{Im}[s_\mu(x + i\eta)]\mathrm{d}x. \tag{5.2.10}$$

In the case where μ admits a density f at x, this is simply given by:

$$f(x) = \lim_{\eta\to 0^+} \mathrm{Im}[s_\mu(x + i\eta)]. \tag{5.2.11}$$

Sketch of the proof. The main idea of the proof is to note that:

$$\frac{1}{\pi} \mathrm{Im}[s_\mu(x + i\eta)] = \frac{1}{\pi} \int \mu(\mathrm{d}\lambda)\, \mathrm{Im}\left[\frac{1}{\lambda - (x + i\eta)}\right] = \frac{1}{\pi} \int \mu(\mathrm{d}\lambda) \frac{\eta}{(\lambda - x)^2 + \eta^2} \tag{5.2.12}$$

This has an interesting interpretation: the imaginary part of the Stieltjes transform can be seen as a convolution between μ and the Cauchy distribution[1]. To see this, let $X \sim \mu$ and $Y \sim$ Cauchy. Then, $X + tY$ has density exactly given by $\frac{1}{\pi} \mathrm{Im}[s_\mu(x + i\eta)]$, and for any $f : \mathbb{R} \to \mathbb{R}$:

$$\mathbb{E}[f(X + Y)] = \int \frac{\mathrm{d}x}{\pi} f(x)\, \mathrm{Im}[s_\mu(x + i\eta)] \tag{5.2.13}$$

As a consequence, the density $\frac{1}{\pi} \mathrm{Im}[s_\mu(x + i\eta)]$ converges weakly to the density μ as $\eta \to 0^+$. □

The last property we will need from the Stieltjes transform is that it not only captures everything we need about the measure, but this also transfer under limits.

Proposition 9. Let μ_n denote a sequence of probability measures on $\mathbb{R}$. Then, μ_n converges weakly to μ if and only if s_{μ_n} converges point-wise to s_μ in the upper-half complex plane $\mathcal{C}_+ = \{z \in \mathbb{C} : \mathrm{Im}(z) > 0\}$.

[1] Recall the Cauchy distribution is a continuous probability distribution with probability density function $\mathrm{Cauchy}(x) = \frac{1}{\pi(1+x^2)}$ supported on $\mathbb{R}$.

Remark 18. Proposition 9 also holds for random sequences of probability measures μ_n under almost surely convergence and convergence in probability in both sides of the "if and only if" statement.

⚠ We can have sequences of probability measures μ_n that do not converge, but for which $s_{\mu_n}(z) \to s(z)$ not corresponding to the Stieltjes transform of a probability measure. For example:

$$\mu_n = \frac{1}{2}\delta_0 + \frac{1}{2}\delta_n \tag{5.2.14}$$

We have:

$$s_{\mu_n}(z) = -\frac{1}{2z} + \frac{1}{2}\frac{1}{n-z} \to -\frac{1}{2z} \tag{5.2.15}$$

So far, our discussion of the Stieltjes transform has been for a generic probability distribution over $\mathbb{R}$. We now turn our attention to the particular case when this is the empirical spectral measure $\hat{\mu}$ of a symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{d\times d}$. In this case, note the Stieltjes transform can also be written as:

$$s_{\hat{\mu}}(z) = \int \frac{\mathrm{d}\hat{\mu}(\lambda)}{\lambda - z} = \frac{1}{d}\sum_{i=1}^{d}\frac{1}{\lambda_i - z} = \frac{1}{d}\operatorname{Tr}(\boldsymbol{M} - z\boldsymbol{I}_d)^{-1} =: \frac{1}{d}\operatorname{Tr}\boldsymbol{R}_{\boldsymbol{M}}(z) \tag{5.2.16}$$

which shows that the Stieltjes transform of the empirical spectral measure is nothing but the empirical average diagonal of the resolvent matrix, connecting back to the discussion in section 5.1. In particular, if $\boldsymbol{M}$ is a random matrix, $\hat{\mu}$ is a random probability measure and $s_{\hat{\mu}}(z)$ a random complex function. The fact that the Stieltjes transform can be seen as the empirical average suggests it is prompt to concentration in the high-dimensional limit $d \to \infty$. This will play a key role in the analysis that follow.

5.3 The Wigner semi-circle law

So far, our discussion has been general. We now turn our attention to a particular example: the Gaussian Orthogonal Ensemble (GOE).

Definition 6 (Gaussian Orthogonal Ensemble)**.** The GOE is the ensemble of random symmetric matrices $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ with independent upper-triangular Gaussian entries:

$$\boldsymbol{W} \sim \texttt{GOE}(d) \quad \Leftrightarrow \quad \begin{cases} W_{ii} \sim \mathcal{N}(0, {}^{2}/_{n}), & 1 \leq i \leq d \\ W_{ij} \sim \mathcal{N}(0, {}^{1}/_{n}), & 1 \leq i < j \leq d \end{cases} \tag{5.3.1}$$

Remark 19. An alternative and equivalent characterisation of $\boldsymbol{W} \sim \boldsymbol{W} \sim \texttt{GOE}(d)$ is:

$$\boldsymbol{W} = \frac{1}{\sqrt{2d}}(\boldsymbol{G} + \boldsymbol{G}^\top) \tag{5.3.2}$$

where $\boldsymbol{G} \in \mathbb{R}^{d\times d}$ is a matrix with Gaussian i.i.d. entries $G_{ij} \sim \mathcal{N}(0,1)$.

Properties 2. GOE matrices $\boldsymbol{W} \sim \texttt{GOE}(d)$ satisfy the following useful properties:

(a) **Normalisation:**

$$\frac{1}{d}\operatorname{Tr}\boldsymbol{W}^2 \xrightarrow{a.s.} 1, \qquad d \to \infty \tag{5.3.3}$$

Since:

$$\begin{aligned}\frac{1}{d}\mathbb{E}\left[\operatorname{Tr}\boldsymbol{W}^2\right] &= \frac{1}{d}\mathbb{E}\left[\sum_{i,j=1}^{d} W_{ij}W_{ji}\right] = \frac{1}{d}\sum_{i,j=1}^{d}\mathbb{E}[W_{ij}^2] = \frac{1}{n}\sum_{i=1}^{d}\mathbb{E}[W_{ii}^2] + \frac{1}{n}\sum_{i\neq j}^{d}\mathbb{E}[W_{ij}^2] \\ &= \frac{2}{n} + \frac{n-1}{n} = 1 + \frac{1}{n} \xrightarrow{d\to\infty} 1\end{aligned} \tag{5.3.4}$$

(b) **Norms:** An entry of $\boldsymbol{W} \sim \texttt{GOE}(d)$ has a typical magnitude $W_{ij} = O(1/\sqrt{n})$. From property (a), we have:

$$||\boldsymbol{W}||_{\texttt{F}} := \sqrt{\operatorname{Tr}[\boldsymbol{W}^\top \boldsymbol{W}]} = O(\sqrt{d}) \tag{5.3.5}$$

Similarly, since the eigenvalues must sum to $\operatorname{Tr}\boldsymbol{W} = \sum_{i=1}^d \lambda_i = O(d)$ and their average square distance from the origin $1/d \operatorname{Tr}\boldsymbol{W}^2 = 1/d \sum_{i=1}^d \lambda_i^2 = O(1)$, it suggests that we have:

$$||\boldsymbol{W}||_{\texttt{op}} = O(1). \tag{5.3.6}$$

Although this heuristic is correct, proving it is not trivial. See Chapter 2.3 in Terence Tao's notes. Together, this implies that the typical spacing between eigenvalues is $O(1/d)$.

(c) **Rotation invariance:** Let $\boldsymbol{U} \in O(d)$ be an orthogonal matrix $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_d$ and $\boldsymbol{W} \sim \texttt{GOE}(d)$. Then, $\boldsymbol{U}\boldsymbol{W}\boldsymbol{U}^\top \sim \texttt{GOE}(d)$ (i.e. $\boldsymbol{U}\boldsymbol{W}\boldsymbol{U}^\top \overset{d}{=} \boldsymbol{W}$). This follows from eq. (5.3.2) the rotational invariance of Gaussian matrices.

The asymptotic behaviour of the empirical spectral measure of GOE matrices was derived by Physicist Eugene Wigner in (Wigner, 1955), who used GOE matrices to model the Hamiltonian of complex nuclei. This is given by the celebrated *Wigner semi-circle law*, see fig. 5.1 for an illustration.

Theorem 8 (Wigner semi-circle law)**.** Let $\boldsymbol{W} \sim \texttt{GOE}(d)$. Then, in the limit $d \to \infty$ the Stieltjes transform $s_{\hat{\mu}}$ of the empirical spectral density $\hat{\mu}_{\boldsymbol{W}}$ converges almost surely to a deterministic function:

$$s_{\hat{\mu}}(z) = \frac{1}{d} \operatorname{Tr}\left(\boldsymbol{W} - z\boldsymbol{I}_d\right)^{-1} \xrightarrow{a.s.} s_\mu(z) = \frac{\sqrt{z^2-4}-z}{2}, \quad z \in \mathbb{C} \setminus [-2,2]. \tag{5.3.7}$$

In particular, s_μ corresponds to the Stieltjes transform of a probability measure with a density supported at $[-2,2]$, known as the Wigner semi-circle law:

$$\mu(\mathrm{d}x) = \frac{\sqrt{4-x^2}}{2\pi} \mathbf{1}_{[-2,2]}(x)\mathrm{d}x \tag{5.3.8}$$

Which thanks to Proposition 9 implies the almost sure weak convergence of $\hat{\mu}_{\boldsymbol{W}}$ to μ as $d \to \infty$.

Sketch of the proof. The proof is separated in two steps:

(I) Show that the Stieltjes transform of the empirical spectral measure $s_{\hat{\mu}}$ concentrates almost surely to its expectation:

$$s_{\hat{\mu}}(z) \xrightarrow{a.s.} \mathbb{E}[s_{\hat{\mu}}(z)], \quad \text{as } d \to \infty \tag{5.3.9}$$

(II) Show that:

$$\mathbb{E}[s_{\hat{\mu}}(z)] \to s_\mu(z), \quad \text{as } d \to \infty \tag{5.3.10}$$

where s_μ can be exactly characterised as the solution of a quadratic equation.

Part I: concentration — To show concentration, we start by showing that the Stieltjes transform $s_{\hat{\mu}}(z)$ is a Lipschitz continous function of the Gaussian matrix $\sqrt{2d}\boldsymbol{G}$:

Lemma 2. Let $\boldsymbol{W} \sim \texttt{GOE}(d)$ denote a GOE matrix:

$$\boldsymbol{W} = \frac{\boldsymbol{G} + \boldsymbol{G}^\top}{\sqrt{2d}} \tag{5.3.11}$$

where $\boldsymbol{G}$ is a Gaussian matrix with i.i.d. $\mathcal{N}(0,1)$ entries. Then, for any $z \in \mathbb{C} \setminus \operatorname{spec}(\boldsymbol{W})$, the Stieltjes transform $s_{\hat{\mu}_{\boldsymbol{W}}}(z)$ of the the empirical spectral distribution of $\boldsymbol{W}$ is a Lipschitz continuous function of $\sqrt{2d}\boldsymbol{G}$ with Lipschitz constant given by:

$$L_d(z) := \frac{1}{d \operatorname{Im}(z)^2} \tag{5.3.12}$$

Sketch of the proof. This follows from a combination of matrix identities and inequalities for the different matrix norms:

$$\begin{aligned}
|s_{\hat{\mu}_{\boldsymbol{W}}}(z) - s_{\hat{\mu}_{\boldsymbol{W}'}}(z)| &= \left| \frac{1}{d} \operatorname{Tr}(\boldsymbol{W} - z\boldsymbol{I}_d)^{-1} - \frac{1}{d} \operatorname{Tr}\big(\boldsymbol{W}' - z\boldsymbol{I}_d\big)^{-1} \right| \\
&\overset{(a)}{=} \left| \frac{1}{d} \operatorname{Tr}\left\{(\boldsymbol{W} - z\boldsymbol{I}_d)^{-1}(\boldsymbol{W} - \boldsymbol{W}')(\boldsymbol{W}' - z\boldsymbol{I}_d)^{-1}\right\} \right| \\
&\overset{(b)}{\leq} \frac{1}{d} ||(\boldsymbol{W} - z\boldsymbol{I}_d)^{-1}||_{\mathsf{op}} ||(\boldsymbol{W}' - z\boldsymbol{I}_d)^{-1}||_{\mathsf{op}} ||\boldsymbol{W} - \boldsymbol{W}||_* \\
&\overset{(d)}{\leq} \frac{2}{\sqrt{d}} \frac{1}{\operatorname{Im}(z)^2} ||\boldsymbol{G} - \boldsymbol{G}'||_{\mathsf{F}} \\
&= \frac{1}{d} \frac{1}{\operatorname{Im}(z)^2} ||\sqrt{2d}\boldsymbol{G} - \sqrt{2d}\boldsymbol{G}'||_{\mathsf{F}}
\end{aligned} \tag{5.3.13}$$

where in (a) we used the resolvent identity:

$$\boldsymbol{A}^{-1} - \boldsymbol{B}^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{A} - \boldsymbol{B})\boldsymbol{B}^{-1}, \tag{5.3.14}$$

in (b) we used that:

$$\operatorname{Tr}(\boldsymbol{A}\boldsymbol{B}) \leq ||\boldsymbol{A}||_* \cdot ||\boldsymbol{B}||_{\mathsf{op}} \tag{5.3.15}$$

and finally in (c) we used the fact that the operator norm is exactly the spectral radius, i.e. for $z \in \mathbb{C} \setminus \operatorname{spec}(\boldsymbol{W})$:

$$\begin{aligned}
||\boldsymbol{R}_{\boldsymbol{W}}(z)||_{\mathsf{op}} &:= \sup_{\xi \in \operatorname{spec}((\boldsymbol{W} - z\boldsymbol{I}_d)^{-1})} |\xi| \\
&= \sup_{\lambda \in \operatorname{spec}(\boldsymbol{W})} \frac{1}{|\lambda - z|} \\
&= \frac{1}{\operatorname{dist}(z, \operatorname{spec}(\boldsymbol{W}))} \leq \frac{1}{\operatorname{Im}(z)}
\end{aligned} \tag{5.3.16}$$

together with the following bound:

$$||\boldsymbol{W}||_* \leq \sqrt{d} ||\boldsymbol{W}||_{\mathsf{F}} \tag{5.3.17}$$

□

This allow us to apply standard tail bounds for Lipschitz functions of Gaussian variables:

Lemma 3 (Theorem 5.6 in (Boucheron et al., 2013))**.** Let $\boldsymbol{g} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ be a vector with i.i.d. standard Gaussian entries, and consider $f : \mathbb{R}^d \to \mathbb{R}$ an L-Lipschitz function. Then, for all $t > 0$:

$$\mathbb{P}[|f(\boldsymbol{g}) - \mathbb{E}[f(\boldsymbol{g})]| \geq t] \leq e^{-\frac{t^2}{2L^2}} \tag{5.3.18}$$

Applying this result to the Stieltjes transform, for any $t > 0$ and $z \in \mathbb{C} \setminus \operatorname{supp}(\boldsymbol{W})$:

$$\mathbb{P}\left[|s_{\hat{\mu}}(z) - \mathbb{E}[s_{\hat{\mu}}(z)]| \geq t\right] \leq e^{-\frac{t^2}{2L_d(z)^2}} = e^{-\frac{d^2 \operatorname{Im}(z)^4 t^2}{2}} \tag{5.3.19}$$

In particular, this implies convergence in probability as $d \to \infty$ for $|z| = O(1)$. This can be made stronger by using the following lemma:

Lemma 4. Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ denote two independent random vectors from the same distribution with finite variance, and consider a L-Lipschitz function $f : \mathbb{R}^d \to \mathbb{R}$. Then:

$$\operatorname{Var}(f(\boldsymbol{x})) \leq L^2 \operatorname{Var}(\boldsymbol{x}) \tag{5.3.20}$$

Sketch of the proof. This is a consequence of a simple calculation:

$$\begin{aligned}
2\mathrm{Var}(f(\boldsymbol{x})) &\overset{(a)}{=} 2\mathrm{Var}\left(f(\boldsymbol{x}) - f(\boldsymbol{x}')\right) \\
&\overset{\text{def.}}{=} \mathbb{E}\left[(f(\boldsymbol{x}) - f(\boldsymbol{x}'))^2\right] - \mathbb{E}\left[f(\boldsymbol{x}) - f(\boldsymbol{x}')\right]^2 \\
&\overset{(b)}{=} \mathbb{E}\left[(f(\boldsymbol{x}) - f(\boldsymbol{x}'))^2\right] \\
&\overset{(c)}{\leq} L^2\mathbb{E}\left[||\boldsymbol{x} - \boldsymbol{x}'||_2^2\right] \\
&= 2L^2\mathrm{Var}(\boldsymbol{x}) \qquad (5.3.21)
\end{aligned}$$

where in (a) we used that $\boldsymbol{x}, \boldsymbol{x}'$ are independent (and therefore uncorrelated) to write $\mathrm{Var}(f(\boldsymbol{x}) - f(\boldsymbol{x}')) = \mathrm{Var}(f(\boldsymbol{x})) + \mathrm{Var}(-f(\boldsymbol{x}')) = 2\mathrm{Var}(f(x))$, in (b) we used that $\boldsymbol{x}, \boldsymbol{x}'$ are identically distributed and in (c) the fact f is Lipschitz. □

Applying lemma 4 to the Stieltjes transform:

$$\mathrm{Var}(s_{\hat{\mu}}(z)) \leq L_d(z)^2 = \frac{1}{d^2}\frac{1}{\mathrm{Im}(z)^4} \qquad (5.3.22)$$

which implies almost sure convergence:

$$s_{\hat{\mu}}(z) \xrightarrow{a.s.} \mathbb{E}[s_{\hat{\mu}}(z)], \qquad \text{as } d \to \infty \qquad (5.3.23)$$

at fixed $|z| = O(1)$.

Part II: asymptotic characterisation — We now turn to the second part of the proof, which consists of finding an exact characterisation of the limit $\mathbb{E}[s_{\hat{\mu}}(z)] \to s_{\mu}(z)$. We will follow a proof scheme known as *leave-one-out*. The key idea is to note that since the entries of $\boldsymbol{W}$ are i.i.d., any sub-matrix of size $(d-1) \times (d-1)$ is still a GOE matrix. Moreover, for large d the properties of the submatrix should be almost the same as the ones of the full matrix. In mathematical terms, we fix an $i \in [d]$ and define the square matrix $\boldsymbol{W}^{(i)} \in \mathbb{R}^{(d-1)\times(d-1)}$ obtained by deleting the corresponding row and column:

$$\boldsymbol{W}^{(i)}_{jk} = (W_{jk})_{j,k\neq i} \qquad (5.3.24)$$

As we said before, this is also a GOE matrix up to an adjustment of the normalisation:

$$\sqrt{\frac{d}{d-1}}\boldsymbol{W}^{(i)} \sim \mathtt{GOE}(d-1). \qquad (5.3.25)$$

The key idea now is to relate the Stieltjes transform of $\boldsymbol{W}$ to that of $\boldsymbol{W}^{(i)}$. To ease the notation, let s_d and s_{d-1} denote the Stieltjes transforms associated to $\boldsymbol{W}$ and $\boldsymbol{W}^{(i)}$, respectively. By definition, we have:

$$\mathbb{E}[s_d(z)] = \mathbb{E}\left[\frac{1}{d}\sum_{i=1}^{d}(\boldsymbol{W} - z\boldsymbol{I}_d)^{-1}_{ii}\right] = \mathbb{E}\left[(\boldsymbol{W} - z\boldsymbol{I}_d)^{-1}_{ii}\right] \qquad (5.3.26)$$

where in the last equality we used all elements in the diagonal have the same distribution (i.e. the expectation is independent of i). Now, we can use the expression for the inverse of a block matrix:

$$\begin{bmatrix} W_{11} & \boldsymbol{w}_1 \\ \boldsymbol{w}_1^\top & \boldsymbol{W}^{(1)} \end{bmatrix}^{-1} = \frac{1}{W_{11} - z - \boldsymbol{w}_1^\top(\boldsymbol{W}^{(1)} - z\boldsymbol{I}_d)^{-1}\boldsymbol{w}_1} \qquad (5.3.27)$$

where for concreteness we wrote for $i = 1$, but the above is true for any minor of $\boldsymbol{W}$. Therefore:

$$\mathbb{E}[s_d(z)] = \mathbb{E}\left[\frac{1}{W_{11} - z - \boldsymbol{w}_1^\top(\boldsymbol{W}^{(1)} - z\boldsymbol{I}_d)^{-1}\boldsymbol{w}_1}\right] \overset{(a)}{=} \mathbb{E}\left[\frac{1}{-z - \boldsymbol{w}_1^\top(\boldsymbol{W}^{(1)} - z\boldsymbol{I}_d)^{-1}\boldsymbol{w}_1}\right] + o(1) \tag{5.3.28}$$

where in (a) we used the fact that $W_{11} = O(1/\sqrt{d})$ is subleading with respect to the remaining terms, which are $O(1)$. Note that the row $\boldsymbol{w}_1$ is a random vector with covariance $1/d\boldsymbol{I}_d$, and is independent from $\boldsymbol{W}^{(1)}$. Therefore, we should expect that:

$$\boldsymbol{w}_1^\top(\boldsymbol{W}^{(1)} - z\boldsymbol{I}_d)^{-1}\boldsymbol{w}_1 = \frac{1}{d}\mathbb{E}\,\mathrm{Tr}\Big(\boldsymbol{W}^{(1)} - z\boldsymbol{I}_d\Big)^{-1} + o(1) =: \mathbb{E}[s_{d-1}(z)] + o(1) \tag{5.3.29}$$

Finally, we also expect both s_{d-1}, s_d to concentrate (due to Part I) to the same value when $d \to \infty$, in other words:

$$s_d(z) = s_{d-1}(z) + o(1) \xrightarrow{a.s.} s_\mu(z) \tag{5.3.30}$$

Putting together, we expect $s_\mu(z)$ to satisfy:

$$s(z) = \frac{1}{-z - s(z)}. \tag{5.3.31}$$

This quadratic equation has two solutions:

$$s_\pm(z) = \frac{-z \pm \sqrt{z^2 - 4}}{2}. \tag{5.3.32}$$

and it is easy to check that only $s_\mu(z) = s_+(z)$ satisfy the properties 1 of the Stieltjes transform, as claimed in eq. (5.3.7).

Now, to make the informal argument above rigorous, we need to justify both eq. (5.3.29) and eq. (5.3.30). The first is a consequence of Hanson-Wright inequality:

Theorem 9 (Hanson-Wright inequality)**.** *Let $\boldsymbol{x} \in \mathbb{R}^d$ be a random vector with independent, mean zero sub-Gaussian coordinates, and let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ denote a deterministic matrix. Then, for every $t \geq 0$ we have:*

$$\mathbb{P}\Big[|\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}]| \geq t\Big] \leq 2e^{-c\min\left\{\frac{t^2}{K^4||\boldsymbol{A}||_{\mathrm{F}}^2}, \frac{t}{K^2||\boldsymbol{A}||_{\mathrm{op}}}\right\}} \tag{5.3.33}$$

where $K = \max_i ||x_i||_{\psi_2}$.

See Theorem 6.2.1 in Vershynin (2018) for a detailed discussion. Recalling that in our case we have:

$$||\boldsymbol{R}_{\boldsymbol{W}}(z)||_{\mathrm{op}} \leq \frac{1}{\mathrm{Im}(z)}, \qquad ||\boldsymbol{R}_{\boldsymbol{W}}(z)||_{\mathrm{F}} \leq \frac{\sqrt{d}}{\mathrm{Im}(z)} \tag{5.3.34}$$

This give us:

$$\left|\boldsymbol{w}_1^\top(\boldsymbol{W}^{(1)} - z\boldsymbol{I}_d)^{-1}\boldsymbol{w}_1 - \mathbb{E}[s_{d-1}(z)]\right| = O\left(\sqrt{\frac{\log d}{d}}\right) \tag{5.3.35}$$

Finally, eq. (5.3.30) is justified by Weyl's interlacing lemma:

Lemma 5 (Weyl's interlacing lemma)**.** Let $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$ be two symmetric matrices with eigenvalues $(\lambda_i(\boldsymbol{A}))_{i\in[d]}$, $(\lambda_i(\boldsymbol{B}))_{i\in[d]}$ arranged in nonincreasing order. The, for all $i \in [n]$:

$$\lambda_i(\boldsymbol{A}+\boldsymbol{B}) \leq \lambda_{i+j}(\boldsymbol{A}) + \lambda_{d-j}(\boldsymbol{B}), \quad j = 0, 1, \dots, d-i \tag{5.3.36}$$

$$\lambda_{i-j+1}(\boldsymbol{A}) + \lambda_j(\boldsymbol{B}) \leq \lambda_i(\boldsymbol{A}+\boldsymbol{B}), \quad j = 1, \dots, i \tag{5.3.37}$$

$$\tag{5.3.38}$$

In particular, taking $i = 1$ in the first equation and $i = d$ in the second, together with the fact $\lambda_j(\boldsymbol{B}) = -\lambda_{d+1-j}(-\boldsymbol{B})$ for $j = 1, \dots, d$ implies that:

$$\max_{j\in[d]}\{\lambda_j(\boldsymbol{A}) - \lambda_j(\boldsymbol{B})\} \leq ||\boldsymbol{A}-\boldsymbol{B}||_{\mathtt{op}} \tag{5.3.39}$$

Applying this for $\boldsymbol{A} = \boldsymbol{W}$ and $\boldsymbol{B} = \boldsymbol{W}^{(i)}$ implies that the difference between their eigenvalues is of $O(1)$, which implies:

$$s_d(z) = s_{d-1}(z) + O(1/d) \tag{5.3.40}$$

Therefore, together we can state that:

$$\mathbb{E}[s_d(z)] = \mathbb{E}\left[\frac{1}{-z - \mathbb{E}[s_d(z)] + o(1)}\right] \overset{(a)}{=} \frac{1}{-z - \mathbb{E}[s_d(z)]} + o(1) \tag{5.3.41}$$

where in (a) we used that

$$\frac{1}{-z - \mathbb{E}[s_d(z)] + o(1)} - \frac{1}{-z - \mathbb{E}[s_d(z)} = \frac{o(1)}{(-z - \mathbb{E}[s_d(z)] + o(1))(-z - \mathbb{E}[s_d(z))} \tag{5.3.42}$$

as long as the denominator is bounded away from zero (which is the case, since it is lower bounded by $|\operatorname{Im}(z)| > 0$).

Equation (5.3.41) is enough to provide a characterisation of $\mathbb{E}[s_d(z)]$ at large d. But to pass to the limit, it remains to show that the self-consistent equation is robust to $o(1)$ changes, i.e. that the solution of eq. (5.3.41) converge to $s_\mu(z)$. To do so, we cam track closer the error by noting that eq. (5.3.41) can also be written as:

$$\mathbb{E}[s_d(z)] = \frac{1}{-z - \mathbb{E}[s_d(z)] + \epsilon_d(z)} \tag{5.3.43}$$

with $\epsilon_d(z) \xrightarrow{a.s.} 0$ as $d \to \infty$ with fixed $|z| = O(1)$. This has solution:

$$s_d^{\pm}(z) = \frac{-(z + \epsilon_d(z)) \pm \sqrt{(z + \epsilon_d(z))^2 - 4}}{2} \tag{5.3.44}$$

To decide the correct branch, we can look at the $|z| \to \infty$ asymptotics in eq. (5.2.9), for which $s(z)$ needs to satisfy:

$$s_\mu(z) = -\frac{1}{z} + O(|z|^{-2}), \text{ as } |z| \to \infty \tag{5.3.45}$$

It is easy to check that the positive branch is the only one consistent with this. Therefore, at $d \to \infty$ we can conclude:

$$s_d^{+}(z) \xrightarrow{a.s.} s_\mu(z) := \frac{-z + \sqrt{z^2 - 4}}{2}, \text{ as } d \to \infty \tag{5.3.46}$$

□

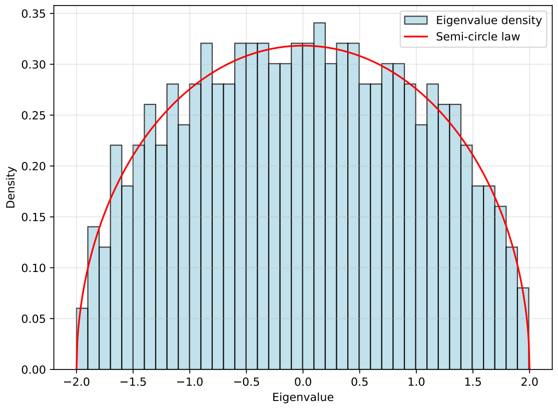

Figure 5.1: Histogram of eigenvalues of a GOE matrix of dimension $d = 500$ with 40 bins. The red solid curve denotes de Wigner semi-circle law $\mu(\mathrm{d}x) = \frac{\sqrt{4-x^2}}{2\pi}\mathbf{1}_{[-2,2]}\mathrm{d}x$.

Remark 20 (Local vs. global laws)**.** The result we proved in Theorem 8 is known as a *global law*, since we assumed $|z| = O(1)$ throughout the proof. Indeed, from the inversion formula in proposition 8, controlling the rate of convergence of the Stieltjes transform in eq. (5.3.7) at a point $z = x + i\eta$ requires the control of $O(\eta d)$ eigenvalues around the point x. Therefore, since we assumed $\eta = O(1)$, our global law is only valid on a *macroscopic* scale $\eta d \gg 1$. A careful handling of the concentration rates can lead to finer results known as *local laws*, valid on an *mesoscopic scale* $1 \ll \eta d \ll d$ (or $\eta d = o(d)$). For Wigner matrices, such local results were first established by Erdős et al. (2009). Finally, note that we cannot hope to do better than that. The typical separation of eigenvalues of GOE matrix inside the bulk is of $O(d^{-1})$. Therefore, at the *microscopic scale* $\eta = \Theta({}^{1}/_{d})$ we we have to deal with statistical properties of single eigenvalues, rather than extensive aggregates. At the edge of the spectrum, eigenvalues have stronger fluctuations of $O(d^{-2/3})$, which are characterised by the celebrated Tracy-Widom law (Tracy and Widom, 2000) — see Figure A.1 for an illustration.

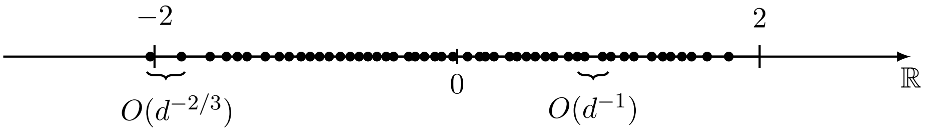

Figure 5.2: Scale of eigenvalue fluctuations for GOE matrices.

Remark 21 (Universality)**.** Gaussian orthogonal matrices are not the only ensemble of random matrices that have an asymptotic spectral density given by the semi-circle law. Indeed, this is true for a large ensemble of random matrices with i.i.d. entries, known as *Wigner matrices*. For instance, the ensemble of adjacency matrices of random Erdös-Rényi graphs, which can be fairly sparse, also have asymptotic semi-circle law (Erdős et al., 2012). See Tao and Vu (2011) and references therein to go deeper.

5.4 Sample covariance matrices and anisotropic laws

Now that we have studied in detail one of the classical random matrix theory result, we turn our attention to our original motivation, which is to understand limits of traces of the type eq. (5.1.3) that appear in the study of ridge regression. There are two main differences with respect to the case studied in Section 5.3:

- The random matrices appearing in eqs. (6.3.4) and (6.3.5) do not have independent entries, and therefore are not Wigner matrices. Indeed, they take the form of *sample covariance matrices*:

$$\hat{\boldsymbol{\Sigma}}_n = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X} \in \mathbb{R}^{d\times d} \tag{5.4.1}$$

  where $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ are random rectangular matrices with i.i.d. rows $\boldsymbol{x}_i \in \mathbb{R}^d$. In the context of random matrix theory, these are also known as *Wishart matrices*. Wishart matrices are widespread in statistics, where one is often interested in estimating the covariance $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}_i\boldsymbol{x}_i^\top]$ from n i.i.d. samples of the covariates $\boldsymbol{x}_i$, $i \in [n]$ (note we will always assume the covariates are centred $\mathbb{E}[\boldsymbol{x}_i] = 0$).

- Differently from the discussion in Section 5.3 which revolved around the asymptotic limit of traces of the resolvent (a.k.a. Stieltjes transform), the expression in eq. (5.1.3) involve products of the resolvent with deterministic matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$, which in regression problems will either be rank-one matrices of the signal $\boldsymbol{A} = \boldsymbol{\theta}_\star\boldsymbol{\theta}_\star^\top$ or the population covariance matrix itself $\boldsymbol{B} = \boldsymbol{\Sigma}$ — see the expression of the bias eq. (6.3.4) for a concrete example. Asymptotic limits of these type of expressions are known as *anisotropic laws*, in contrast to theorem 8 which are isotropic.

5.4.1 Classical statistical regime

The large d limit of Wishart matrices notably depends on the interplay with the number of samples n. For instance, the *classical regime* that is often studied in statistics textbooks considers the limit of large sample sizes $n \to \infty$ at fixed covariate dimension $d = O(1)$, and for which the following guarantees hold:

(a) Entrywise consistency:

$$\hat{\boldsymbol{\Sigma}}_n \xrightarrow{a.s.} \boldsymbol{\Sigma}, \qquad \text{as } n \to \infty, \quad d = O(1) \tag{5.4.2}$$

which is a consequence of the strong law of large numbers for individual entries.

(b) Entrywise Gaussian fluctuations:

$$\sqrt{n}\text{vec}(\hat{\boldsymbol{\Sigma}}_n - \boldsymbol{\Sigma}) \xrightarrow{d} \mathcal{N}(0, V) \qquad \text{as } n \to \infty \text{ with } d = O(1) \tag{5.4.3}$$

where V is a covariance tensor with entries:

$$V_{jklm} = \mathbb{E}[(X_{ij}X_{ik} - \Sigma_{jk})(X_{il}X_{im} - \Sigma_{lm})] \tag{5.4.4}$$

Note this depends on fourth moments of the underlying distribution. This is a consequence of the Central Limit Theorem applied to $\hat{\boldsymbol{\Sigma}}_n$ when viewed as a vector in $\mathbb{R}^{d^2}$.

(c) Let $\text{spec}(\boldsymbol{\Sigma}) = \{\lambda_1, \dots, \lambda_d\}$ and $\text{spec}(\hat{\boldsymbol{\Sigma}}_n) = \{\hat{\lambda}_1, \dots, \hat{\lambda}_d\}$. Then, almost sure convergence entrywise implies point-wise almost sure convergence of the individual eigenvalues:

$$\hat{\lambda}_k \xrightarrow{a.s.} \lambda_k, \qquad \text{as } n \to \infty \text{ with } d = O(1) \tag{5.4.5}$$

Moreover, we also have Gaussian fluctuations for the eigenvalues:

$$\sqrt{n}(\hat{\lambda}_k - \lambda_k) \xrightarrow{d} \mathcal{N}(0, \sigma_k^2) \qquad \text{as } n \to \infty \text{ with } d = O(1) \tag{5.4.6}$$

For a variance σ_k^2 depending on fourth moments of the underlying distribution.

Remark 22. The sample covariance matrix $\hat{\boldsymbol{\Sigma}}_n$ is the maximum likelihood estimator for $\boldsymbol{\Sigma}$. Therefore, the above results also follow directly from consistency of the MLE in the limit $n \to \infty$ at fixed $d = O(1)$.

These results suffice to characterise the traces of interest in eq. (5.1.3) in the classical limit. For instance, they imply that:

$$\text{Tr}\{\boldsymbol{A}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}\} \xrightarrow{a.s.} \text{Tr}\{\boldsymbol{A}(\boldsymbol{\Sigma} - z\boldsymbol{I}_d)^{-1}\}, \qquad \text{as } n \to \infty \text{ with } d = O(1) \tag{5.4.7}$$

5.4.2 High-dimensional regime

The results derived in the classical regime are weakly dependent on the distribution of the covariates. Indeed, they only require mild condition on the existence of fourth moments, which we deliberately omited. This is a common trend in classical statistics: as data is abundant $n \gg 1$, the details of the underlying distribution is not very important for consistent estimation. However, in real life limits don't exist, and "large" begs the question: *how large does n needs to be with respect to d?* Unfortunately, the above results do not allow us to answer this question as they do not track the explicit dependence on the dimension $d = O(1)$. A simple refinement of the above results can be derived under an assumption on the tails of the covariates:

Theorem 10 (Vershynin (2018), Theorem 4.7.1)**.** Let $\boldsymbol{x}_i \in \mathbb{R}^d$, $i \in [n]$ denote independent sub-Gaussian vectors with zero mean and covariance $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}_i \boldsymbol{x}_i^\top] \in \mathbb{R}^{d\times d}$. Then, with probability $1 - 2e^{-t}$:

$$||\hat{\boldsymbol{\Sigma}}_n - \boldsymbol{\Sigma}||_{\mathsf{op}} \leq C \left(\sqrt{\frac{d+t}{n}} + \frac{d+t}{n} \right) ||\boldsymbol{\Sigma}||_{\mathsf{op}} \tag{5.4.8}$$

Theorem 10 tell us that the sample covariance matrix is operator norm consistent as long as ${}^{d}/_{n} \to 0$. Our goal now will be to use RMT to quantify the differences between $\hat{\boldsymbol{\Sigma}}_n$ and $\boldsymbol{\Sigma}$ when both $n, d \to \infty$ with $n = \Theta(d)$, i.e. ${}^{d}/_{n} = \gamma_n \to \gamma = O(1)$.

Theorem 11 (Anisotropic law for Wishart matrices)**.** Let $\hat{\boldsymbol{\Sigma}}_n = {}^{1}/_{n}\boldsymbol{X}^\top \boldsymbol{X} \in \mathbb{R}^{d\times d}$ with $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\Sigma}^{1/2}$, where $\boldsymbol{Z}$ is a sub-Gaussian matrix with zero mean and unit variance and $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is a positive-definite matrix with eigenvalues $\operatorname{spec}(\boldsymbol{\Sigma}) = \{\lambda_k : k \in [d]\} \subset \mathbb{R}_+$ and bounded operator norm $||\boldsymbol{\Sigma}||_{\mathsf{op}} < C$. Assume that the empirical measure of eigenvalues $\hat{\mu}_n = {}^{1}/_{d}\sum_{i\in[d]} \delta_{\lambda_i}$ converges (weakly) to a probability distribution μ on $\mathbb{R}_+$ with compact support as $d \to \infty$. Then, for any $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ with bounded operator norm:

$$\frac{1}{d}\operatorname{Tr}\{\boldsymbol{A}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}\} \xrightarrow{a.s.} -\frac{1}{z\tilde{s}(z)}\frac{1}{d}\operatorname{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-1}\right\} \text{ as } d \to \infty \text{ and } {}^{d}/_{n} \to \gamma., \tag{5.4.9}$$

where $\tilde{s}(z)$ is the unique solution of the following self-consistent equation:

$$\frac{1}{\tilde{s}(z)} + z = \gamma \int_0^\infty \mu(\mathrm{d}\lambda) \frac{\lambda}{1 + \lambda \tilde{s}(z)} \tag{5.4.10}$$

Remark 23. Before we move to the proof of this result, a few remarks are in order.

- Note that when $\gamma \to 0^+$, $\tilde{s}(z) \to -{}^{1}/_{z}$ and eq. (5.4.9) gives the classical result in eq. (5.4.7), as expected. For $\gamma > 0$, this provides a remarkable result: the high-dimensional limit of trace statistics of the empirical covariance resolvent is equal to the population, up to a renormalisation of the spectral parameter $z \mapsto {}^{1}/_{\tilde{s}(z)}$.

- The quantity $\tilde{s}(z)$ is known as the companion Stieltjes transform, and can be shown to be given by:

$$\tilde{s}_d(z) := \frac{1}{n}\operatorname{Tr}\left({}^{1}/_{n}\boldsymbol{X}\boldsymbol{X}^\top - z\boldsymbol{I}_n\right)^{-1} \xrightarrow{a.s.} \tilde{s}(z) \quad \text{as} \quad n, d \to \infty \tag{5.4.11}$$

  i.e. it is the Stieltjes transform of the data Gram matrix ${}^{1}/_{n}\boldsymbol{X}\boldsymbol{X}^\top$. This is related to the standard Stieltjes transform for the sample covariance matrix:

$$s_d(z) := \frac{1}{d}\operatorname{Tr}\left({}^{1}/_{n}\boldsymbol{X}^\top\boldsymbol{X} - z\boldsymbol{I}_d\right)^{-1} \xrightarrow{a.s.} s(z) \quad \text{as} \quad n, d \to \infty \tag{5.4.12}$$

by the following relation:

$$s_d(z) = \frac{1}{\gamma}\tilde{s}_d(z) + \frac{1-\gamma}{\gamma z} \tag{5.4.13}$$

which is a simple consequence of the fact that the matrices $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{X}^\top\boldsymbol{X}$ have the same spectrum up to the zero eigenvalues.

- The anisotropic law in eq. (5.4.9) is sometimes referred in the literature as a *deterministic equivalent*. More precise, we say that a random matrix $\boldsymbol{M} \in \mathbb{R}^{d\times d}$ has deterministic equivalent $\bar{\boldsymbol{M}} \in \mathbb{R}$, denoted $\boldsymbol{M} \leftrightarrow \bar{\boldsymbol{M}}$, if for any deterministic matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ with bounded operator norm:

$$\frac{1}{d}\operatorname{Tr}\boldsymbol{A}\boldsymbol{M} \to \frac{1}{d}\operatorname{Tr}\boldsymbol{A}\bar{\boldsymbol{M}}, \qquad \text{as } d\to\infty \tag{5.4.14}$$

with the convergence being sometimes almost surely or in probability. In other words, measuring any scalar or "trace-like" statistics of the random matrix is the same as doing the same measurement on the deterministic matrix. See (Couillet and Liao, 2022) for a reference that employs this notion.

Sketch of the proof. We now sketch the main ideas in the proof of theorem 11 using a similar leave-one-out argument as in the the Wigner case, without aiming at the same level of rigour. As in the Wigner case, the proof can be separated in two steps: (a) the almost sure convergence of the (companion) Stieltjes transform; (b) finding an exact asymptotic characterisation of the trace of interest:

$$\frac{1}{d}\operatorname{Tr}\boldsymbol{A}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1} \overset{(b)}{=} \operatorname{Tr}\boldsymbol{A}(-z\tilde{s}_d(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d)^{-1} + o(1) \overset{(a)}{=} \operatorname{Tr}\boldsymbol{A}(-z\tilde{s}(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d)^{-1} + o(1) \tag{5.4.15}$$

One could proceed at directly proving the approximations above. However, this would assume that we know the asymptotic limit of the trace in advance. Instead, here we focus on how to derive (b) from scratch, using a leave-one-out argument over the independent rows of the sample covariance matrix $\hat{\boldsymbol{\Sigma}}_n$. For that, define:

$$\hat{\boldsymbol{\Sigma}}_n^{(i)} = \frac{1}{n}\sum_{j\neq i}\boldsymbol{x}_j\boldsymbol{x}_j^\top \tag{5.4.16}$$

such that $\hat{\boldsymbol{\Sigma}}_n = \hat{\boldsymbol{\Sigma}}_n^{(i)} + {}^1\!/_n\boldsymbol{x}_i\boldsymbol{x}_i^\top$. Similarly, define the leave-one-out resolvent matrix:

$$\boldsymbol{R}^{(i)}(z) := \boldsymbol{R}_{\hat{\boldsymbol{\Sigma}}_n^{(i)}}(z) = \left(\hat{\boldsymbol{\Sigma}}_n^{(i)} - z\boldsymbol{I}_d\right)^{-1}. \tag{5.4.17}$$

This can be related to the sample covariance resolvent via the Sherman-Morrison lemma:

Lemma 6 (Sherman-Morrison lemma)**.** Let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ denote an invertible matrix and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{d\times d}$ two vectors such that $\boldsymbol{v}^\top\boldsymbol{A}^{-1}\boldsymbol{u} \neq -1$. Then:

$$(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v})^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^\top\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^\top\boldsymbol{A}^{-1}\boldsymbol{u}} \tag{5.4.18}$$

Applying it to the sample covariance resolvent:

$$\begin{aligned}\boldsymbol{R}(z) := (\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1} = \left(\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{x}_j\boldsymbol{x}_j^\top - z\boldsymbol{I}_d\right)^{-1} &= \left(\frac{1}{n}\sum_{j\neq i}^{n}\boldsymbol{x}_j\boldsymbol{x}_j^\top + \frac{\boldsymbol{x}_i\boldsymbol{x}_i^\top}{n} - z\boldsymbol{I}_d\right)^{-1} \\ &= \left(\hat{\boldsymbol{\Sigma}}_n^{(i)} - z\boldsymbol{I}_d\right)^{-1}\left[\boldsymbol{I}_d - \frac{\boldsymbol{x}_i\boldsymbol{x}_i^\top\left(\hat{\boldsymbol{\Sigma}}_n^{(i)} - z\boldsymbol{I}_d\right)^{-1}}{n + \boldsymbol{x}_i^\top\left(\hat{\boldsymbol{\Sigma}}_n^{(i)} - z\boldsymbol{I}_d\right)^{-1}\boldsymbol{x}_i}\right]\end{aligned} \tag{5.4.19}$$

Hence:

$$\boldsymbol{R}(z) - \boldsymbol{R}^{(i)}(z) = -\frac{\boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{R}^{(i)}(z)}{n + \boldsymbol{x}_i^\top \boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i}, \tag{5.4.20}$$

and we expect that for any deterministic matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ with bounded spectral norm and $z \in \mathbb{C}_+$:

$$\operatorname{Tr} \boldsymbol{A}\left(\boldsymbol{R}(z) - \boldsymbol{R}^{(i)}(z)\right) = o(1) \tag{5.4.21}$$

Note in particular that:

$$\begin{aligned}\operatorname{Tr} \boldsymbol{A} &= \operatorname{Tr} \boldsymbol{R}(z)^{-1}\boldsymbol{R}(z)\boldsymbol{A}\\ &= \operatorname{Tr} \boldsymbol{R}(z)\hat{\boldsymbol{\Sigma}}_n\boldsymbol{A} - z\operatorname{Tr} \boldsymbol{R}(z)\boldsymbol{A}\end{aligned} \tag{5.4.22}$$

The first term can be written as:

$$\begin{aligned}\operatorname{Tr}\{\boldsymbol{R}(z)\hat{\boldsymbol{\Sigma}}_n\boldsymbol{A}\} &= \frac{1}{n}\sum_{i=1}^n \operatorname{Tr}\left\{\boldsymbol{R}(z)\boldsymbol{x}_i\boldsymbol{x}_i^\top \boldsymbol{A}\right\}\\ &= \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^\top \boldsymbol{A}\boldsymbol{R}(z)\boldsymbol{x}_i\end{aligned} \tag{5.4.23}$$

we would like to apply the Hanson-Wright inequality theorem 9 to this term. However, $\boldsymbol{R}$ is correlated with $\boldsymbol{x}_i$, and as in the Wigner case we need to first remove this correlation. From eq. (5.4.19) it follows that:

$$\boldsymbol{R}(z)\boldsymbol{x}_i = \frac{\boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i}{1 + {}^1\!/_n\boldsymbol{x}_i^\top \boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i} \tag{5.4.24}$$

and hence:

$$\operatorname{Tr}\{\boldsymbol{R}(z)\hat{\boldsymbol{\Sigma}}_n\boldsymbol{A}\} = \sum_{i=1}^n \frac{\boldsymbol{x}_i^\top \boldsymbol{A}\boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i}{n + \boldsymbol{x}_i^\top \boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i} \tag{5.4.25}$$

By theorem 9, we have almost surely that for $|z| = O(1)$:

$$\boldsymbol{x}_i^\top \boldsymbol{A}\boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i = \operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{R}^{(i)}(z)\} + o(d), \qquad \boldsymbol{x}_i^\top \boldsymbol{R}^{(i)}(z)\boldsymbol{x}_i = \operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{R}^{(i)}(z)\} + o(d) \tag{5.4.26}$$

Hence:

$$\operatorname{Tr}\{\boldsymbol{R}(z)\hat{\boldsymbol{\Sigma}}_n\boldsymbol{A}\} = \sum_{i=1}^n \frac{\operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{R}^{(i)}(z)\}}{n + \operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{R}^{(i)}(z)\}} + o(d) = n\frac{\operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{R}(z)\}}{n + \operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{R}(z)\}} + o(d) \tag{5.4.27}$$

where in the last equality we used that $\boldsymbol{R}^{(i)} = \boldsymbol{R} + o(1)$. Now inserting this back into eq. (5.4.22), we have that:

$$\begin{aligned}\operatorname{Tr} \boldsymbol{A} &= n\frac{\operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{R}(z)\}}{n + \operatorname{Tr}\{\boldsymbol{\Sigma}\boldsymbol{R}(z)\}} - z\operatorname{Tr} \boldsymbol{R}(z)\boldsymbol{A} + o(d)\\ &= \operatorname{Tr}\left\{\boldsymbol{R}(z)\left(\frac{\boldsymbol{\Sigma}}{1 + \frac{1}{n}\operatorname{Tr} \boldsymbol{\Sigma}\boldsymbol{R}(z)} - z\boldsymbol{I}_d\right)\boldsymbol{A}\right\} + o(d)\end{aligned} \tag{5.4.28}$$

which is valid for any deterministic $\boldsymbol{A} \in \mathbb{R}^{d\times d}$. Rearranging this equation:

$$\operatorname{Tr}\left\{\left[\frac{1}{z}\left(\frac{\boldsymbol{R}(z)\boldsymbol{\Sigma}}{1 + \frac{1}{n}\operatorname{Tr} \boldsymbol{R}(z)\boldsymbol{\Sigma}} - \boldsymbol{I}_d\right) - \boldsymbol{R}(z)\right]\boldsymbol{A}\right\} = o(d). \tag{5.4.29}$$

and comparing with the definition of the deterministic equivalent $\bar{R}(z)$:

$$\operatorname{Tr} \boldsymbol{A}(\boldsymbol{R}(z) - \bar{\boldsymbol{R}}(z)) = o(d) \tag{5.4.30}$$

one would like to identify:

$$\bar{\boldsymbol{R}}(z) = \frac{1}{z}\left(\frac{\boldsymbol{R}(z)\boldsymbol{\Sigma}}{1+\frac{1}{n}\operatorname{Tr}\boldsymbol{R}(z)\boldsymbol{\Sigma}} - \boldsymbol{I}_d\right) + o(d) \tag{5.4.31}$$

However, this still depends on the resolvent $\boldsymbol{R}(z)$, a random quantity. To get rid of this dependence, we evaluate eq. (5.4.28) at $\boldsymbol{A} = \boldsymbol{I}_d$, which after recognising the Stieltjes transform $s_d(z) = {}^1\!/\!_d \operatorname{Tr}\boldsymbol{R}(z)$ give us:

$$d = \frac{\operatorname{Tr}\left\{\boldsymbol{R}(z)\boldsymbol{\Sigma}\right\}}{1+\frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}\boldsymbol{R}(z)\right\}} - zds_d(z). \tag{5.4.32}$$

This can be rearranged to give:

$$1 + zs_d(z) = \frac{\frac{1}{d}\operatorname{Tr}\left\{\boldsymbol{R}(z)\boldsymbol{\Sigma}\right\}}{1+\frac{\gamma}{d}\operatorname{Tr}\left\{\boldsymbol{R}(z)\boldsymbol{\Sigma}\right\}} = \frac{1}{\gamma}\left(1 - \frac{1}{1+\gamma\operatorname{Tr}\{\boldsymbol{R}(z)\boldsymbol{\Sigma}\}}\right) \tag{5.4.33}$$

Now solving for $\operatorname{Tr}\{\boldsymbol{R}(z)\boldsymbol{\Sigma}\}$ give us:

$$\frac{1}{1+\gamma\operatorname{Tr}\{\boldsymbol{R}(z)\boldsymbol{\Sigma}\}} = 1 - \gamma(1 + zs_d(z)) = -z\tilde{s}_d(z) \tag{5.4.34}$$

where we recognised the companion Stieltjes transform (5.4.11) from the relation in eq. (5.4.13). This shows that actually the terms depending on $\boldsymbol{R}$ are actually only a function of $\tilde{s}_d(z)$, a quantity which concentrate. Inserting this back in eq. (5.4.28)

$$\begin{aligned}\operatorname{Tr}\boldsymbol{A} &= \operatorname{Tr}\left\{\boldsymbol{R}(z)\left(-z\tilde{s}_d(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d\right)\boldsymbol{A}\right\} + o(d) && (5.4.35)\\ &= \operatorname{Tr}\left\{\boldsymbol{R}(z)\left(-z\tilde{s}(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d\right)\boldsymbol{A}\right\} + o(d) && (5.4.36)\end{aligned}$$

where in the second equality we used that $\tilde{s}_d(z) = \tilde{s}(z) + o(1)$. Finally, we can evaluate this at the (deterministic) value $\boldsymbol{A} = (-z\tilde{s}(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d)^{-1}$ to get:

$$\frac{1}{d}\operatorname{Tr}\{(-z\tilde{s}(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d)^{-1}\} = \frac{1}{d}\operatorname{Tr}\boldsymbol{R}(z) + o(1) = s(z) + o(1) \tag{5.4.37}$$

This is exactly the result stated in eq. (5.4.9). Using again eq. (5.4.13) allow us to get the self-consistent equation on $\tilde{s}(z)$ stated in eq. (6.3.3):

$$\frac{1}{\tilde{s}(z)} + z = \lim_{d\to\infty}\frac{\gamma}{d}\operatorname{Tr}\big(\boldsymbol{\Sigma}(1+\tilde{s}(z)\boldsymbol{\Sigma})^{-1}\big) = \int_0^\infty \mu(\mathrm{d}\lambda)\frac{\lambda}{1+\lambda\tilde{s}(z)} \tag{5.4.38}$$

The argument above can be made rigorous by justifying the different approximations we have taken. We now briefly sketch how to proceed.

Using the resolvent identity in eq. (5.3.14) we can write the difference between the resolvent and its (finite d) equivalent:

$$(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1} = (-z\tilde{s}_d(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d)^{-1}(\boldsymbol{I}_d - \boldsymbol{\Delta}) \tag{5.4.39}$$

where:

$$\boldsymbol{\Delta} \coloneqq \hat{\boldsymbol{\Sigma}}_n(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1} - z\tilde{s}_d(z)\boldsymbol{\Sigma}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1} \tag{5.4.40}$$

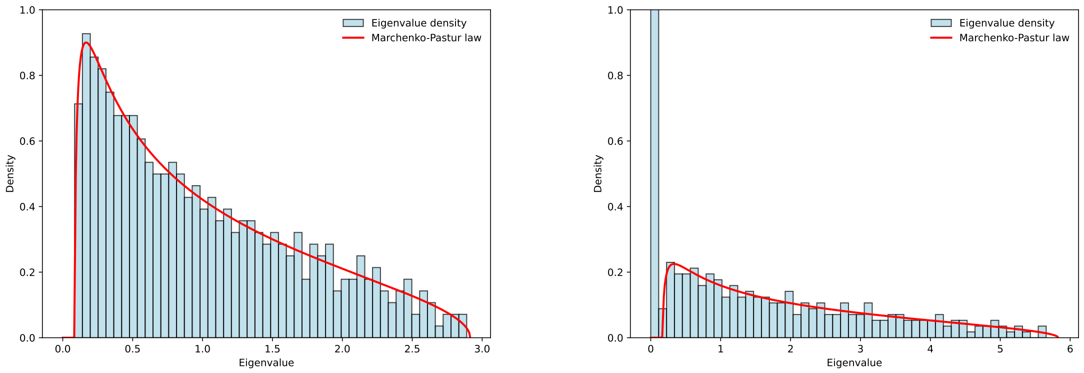

Figure 5.3: Histogram of eigenvalues of a Wishart matrix of dimension $d = 500$ with 40 bins for $\gamma = 0.5$ (**left**) and $\gamma = 2$ (**right**). The red solid curve denotes de Marchenko-Pastur law given by eq. (5.4.43)

Therefore, proving eq. (5.4.15) amounts to showing that $\operatorname{Tr}(-z\tilde{s}_d(z)\boldsymbol{\Sigma} - z\boldsymbol{I}_d)^{-1}\boldsymbol{\Delta} = o(d)$. Using the leaving-one-out argument as in eq. (5.4.19), we can show that $\boldsymbol{\Delta}$ can be re-written as:

$$\boldsymbol{\Delta} = \sum_{i=1}^{n} \frac{\boldsymbol{x}_i\boldsymbol{x}_i^\top(\hat{\boldsymbol{\Sigma}}_n^{(i)} - z\boldsymbol{I}_d)^{-1} - \boldsymbol{\Sigma}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}}{n + \boldsymbol{x}_i^\top(\hat{\boldsymbol{\Sigma}}_n^{(i)} - z\boldsymbol{I}_d)^{-1}\boldsymbol{x}_i} \tag{5.4.41}$$

writing $\boldsymbol{x}_i = \boldsymbol{\Sigma}^{1/2}\boldsymbol{u}_i$ for independent sub-Gaussian vectors $\boldsymbol{u}_i \in \mathbb{R}^d$ with zero mean and unit variance, controlling $\operatorname{Tr}\boldsymbol{\Delta}$ involves controlling traces of the type $\operatorname{Tr}\boldsymbol{A}(\boldsymbol{u}_i\boldsymbol{u}_i - \boldsymbol{I}_d)$ for deterministic $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ with bounded operator norm. See (Silverstein, 1995) for a detailed proof. □

5.4.3 The Marchenko-Pastur law

In the isotropic case $\boldsymbol{\Sigma} = \boldsymbol{I}_d$, the solution of the self-consistent equation eq. (6.3.3) together with eq. (5.4.13) give us the celebrated result by Marchenko and Pastur for the asymptotic Stieltjes transform of Wishart matrices (Pastur and Martchenko, 1967):

$$s(z) = \frac{1 - \gamma - z + \sqrt{(1 - \gamma + z)^2 - 4z}}{2\gamma z} \tag{5.4.42}$$

This is the Stieltjes transform of the celebrated *Marchenko-Pastur law*:

$$\mu_{\mathrm{mp}}(\mathrm{d}x) = \left(1 - \frac{1}{\gamma}\right)_+ \delta_0 + \frac{\sqrt{(\gamma_+ - x)(x - \gamma_-)}}{2\pi\gamma x}\mathbf{1}_{[\gamma_-,\gamma_+]}(x)\mathrm{d}x \tag{5.4.43}$$

where the edges of the distribution are $\gamma_\pm = (1 \pm \sqrt{\gamma})^2$. Figure 5.3 illustrates this result. Recall that $\boldsymbol{\Sigma} = \boldsymbol{I}_d$, and therefore the spectral measure of the population covariance is simply a point mass:

$$\mu_{\boldsymbol{\Sigma}} = \delta_1 \tag{5.4.44}$$

This is strikingly different from μ_{mp}, where the eigenvalues are spread over a full interval $[\gamma_-, \gamma_+] \subset \mathbb{R}_+$. In particular, when $\gamma > 1$ ($d > n$), there are $d - n$ zero eigenvalues (since $\operatorname{rank}(\hat{\boldsymbol{\Sigma}}_n) = n$ almost surely). Therefore, in this high-dimensional regime a naive statistician might be mistakenly led to believe that the true data has a low-dimensional structure where in fact it is completely isotropic.

5.4.4 Other useful results

To conclude of exposition of random matrix theory, we present an additional result concerning bi-product of traces that are relevant to our discussion of ridge regression. These can be derived and proven following exactly the same argument as in Theorem 11. However, their derivation is more cumbersome, and we refer the interested reader to (Bach, 2024) for a detailed discussion.

Theorem 12 (Proposition 1 in (Bach, 2024))**.** Under the same assumptions as in Theorem 12, for any $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$ with bounded operator norm we have:

$$\operatorname{Tr}\{\boldsymbol{A}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}\boldsymbol{B}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}\} \xrightarrow{a.s.} \frac{1}{z^2\tilde{s}(z)^2}\operatorname{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma}+\frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-1}\boldsymbol{B}\left(\boldsymbol{\Sigma}+\frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-1}\right\}$$
$$+\frac{1}{z^2\tilde{s}(z)^2}\frac{\operatorname{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma}+\frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-2}\boldsymbol{\Sigma}\right\}\cdot\operatorname{Tr}\left\{\boldsymbol{B}\left(\boldsymbol{\Sigma}+\frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-2}\boldsymbol{\Sigma}\right\}}{n-\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma}+{}^1\!/\!\tilde{s}(z)\boldsymbol{I}_d)^{-2}\right\}} \tag{5.4.45}$$

where $\tilde{s}(z)$ is the unique solution of the following self-consistent equation eq. (6.3.3).

5.5 To go further

A nice list of useful lecture notes on random matrix theory.

- Terence Tao's lecture notes on random matrix theory taught at UCLA.
- Lecture 17 of Song Mei's STAT260 course taught at Berkeley.
- Florent Benaych-Georges and Antti Knowles lecture notes on local laws for Wigner matrices.
- Chapter 6 of Djalil Chafai lecture notes for the course "*Phénomènes de grande dimension*" taught at ENS (in French).
- Charles Bordenave lecture notes on random matrix theory for a course taught at IMPA.

Books:

- Zhidong Bai and Jack W. Silverstein "*Spectral Analysis of Large Dimensional Random Matrices* ", available online at the editor's webpage. Classical reference by the people behind many of the results in RMT.
- Romain Couillet and Zhenyu Liao book "*Random Matrix Methods for Machine Learning*", available online at the author's webpage. A good recent reference with detailed proofs and a discussion on the applications to machine learning.
- Marc Potters and Jean-Philippe Bouchaud book "*A First Course in Random Matrix Theory*". Very good reference written by physicists, with plenty of intuition.

Chapter 6

Random design analysis of ridge regression

6.1 Setting

Consider a supervised regression setting with training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R} : i \in [n]\}$. In this lecture, we will be interested in the analysis of ridge regressor $f(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_\lambda) = \langle \hat{\boldsymbol{\theta}}_\lambda, \boldsymbol{x} \rangle$ with:

$$\hat{\boldsymbol{\theta}}_\lambda(\boldsymbol{X}, \boldsymbol{y}) := \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle)^2 + \lambda ||\boldsymbol{\theta}||_2^2 \tag{6.1.1}$$

$$= \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y} \tag{6.1.2}$$

where $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y} \in \mathbb{R}^n$ denote the covariate matrix and response vector, obtained by stacking $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ row-wise. Since the ridge predictor is a linear function, it can only express linear dependences on the data. Therefore, it is natural to assume that data has been independently drawn from a linear model:

$$y_i = \langle \boldsymbol{\theta}_\star, \boldsymbol{x}_i \rangle + \varepsilon_i. \tag{6.1.3}$$

In particular, we will be interested in analysing the so-called *random design setting.*

Assumption 1. Throughout this lecture, we assume that:

- Gaussian covariates, i.e. $\boldsymbol{x}_i = \boldsymbol{\Sigma}^{1/2} \boldsymbol{z}_i$ with $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and $\boldsymbol{\Sigma} \succ 0$ a positive-definite matrix.
- The label noise are drawn independently from $\boldsymbol{x}_i$, are zero-mean and have finite variance $\mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty$.
- The target weights have finite norm $||\boldsymbol{\theta}_\star||_2^2 < \infty$.

Note that the above implicitly define a joint distribution $p(\boldsymbol{x}, y)$. In particular, the Bayes risk is given by the noise variance $R_\star = \sigma^2$, and corresponds to the Bayes predictor $f_\star(\boldsymbol{x}) = \langle \boldsymbol{\theta}_\star, \boldsymbol{x} \rangle$.

Our goal is to derive a precise characterisation of the excess risk:

$$R(\hat{\boldsymbol{\theta}}_\lambda) - \sigma^2 = \mathbb{E}_{\boldsymbol{x}} \left[\left(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_\lambda) \right)^2 \right] \tag{6.1.4}$$

$$= \langle \hat{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}_\star, \boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}_\star) \rangle \tag{6.1.5}$$

$$:= ||\hat{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}_\star||_{\boldsymbol{\Sigma}}^2 \tag{6.1.6}$$

where in the last equality we simply recognised the definition of the *Mahalanobis norm*. In other words: the excess risk is a mean-squared error weighted by the most relevant directions in the data (i.e. with largest eigenvalues).

Note that the excess risk above is a random quantity (since $\hat{\boldsymbol{\theta}}_\lambda(\boldsymbol{X}, \boldsymbol{y})$ is a function of the training data, which is random). To simplify the analysis, we will rather consider the average of the expected risk with respect to the label noise:

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[R(\hat{\boldsymbol{\theta}}_\lambda)\right] - \sigma^2 = \mathbb{E}_{\boldsymbol{\varepsilon}}\left[||\hat{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}_\star||^2_{\boldsymbol{\Sigma}}\right] \tag{6.1.7}$$

With more work, one can show concentration of $R(\hat{\boldsymbol{\theta}}_\lambda)$ over $\boldsymbol{\varepsilon}$ under an additional sub-Gaussian assumption.

⚠ $\mathbb{E}_{\boldsymbol{\varepsilon}}\left[R(\hat{\boldsymbol{\theta}}_\lambda(\boldsymbol{X}, \boldsymbol{y}))\right]$ is still a random quantity since it depends on the random data $\boldsymbol{X}, \boldsymbol{y}$.

6.2 Bias-variance decomposition

In regression problems with additive noise $y_i = f_\star(\boldsymbol{x}_i) + \varepsilon_i$, it is common to write the expected excess risk in terms of a (squared) bias and variance decomposition with respect to the label noise:

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[R(\hat{\boldsymbol{\theta}}_\lambda)\right] - \sigma^2 = B(f_\star, \boldsymbol{X}, \lambda) + V(\boldsymbol{X}, \lambda) \tag{6.2.1}$$

where:

$$B(f_\star, \boldsymbol{X}, \lambda) = \mathbb{E}_{\boldsymbol{x}}\left[(f_\star(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{\varepsilon}}[f(\boldsymbol{x}; \hat{\boldsymbol{\theta}})])^2\right] \tag{6.2.2}$$

$$V(\boldsymbol{X}, \lambda) = \mathrm{Var}_{\boldsymbol{\varepsilon}}(f(\boldsymbol{x}; \hat{\boldsymbol{\theta}})) = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{\varepsilon}}\left[(f(\boldsymbol{x}; \hat{\boldsymbol{\theta}}) - \mathbb{E}_{\boldsymbol{\varepsilon}}[f(\boldsymbol{x}; \hat{\boldsymbol{\theta}})])^2\right] \tag{6.2.3}$$

In particular, note that the variance is independent of the target function $f_\star$.

⚠ Note that the (squared) bias and variance are defined with respect to the training data label noise $\boldsymbol{\varepsilon} \in \mathbb{R}^n$. In particular, they are still random functions of $\boldsymbol{X}$.

Explicit expressions for the bias and variance can be worked out from the definition. But in our case it is simpler to note that we can write:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_\lambda(\boldsymbol{X}, \boldsymbol{y}) &= \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\theta}_\star + \boldsymbol{\varepsilon}) && (6.2.4)\\
&= \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\theta}_\star + \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{X}^\top \boldsymbol{\varepsilon} && (6.2.5)\\
&\overset{(a)}{=} \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} (\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d - n\lambda \boldsymbol{I}_d)\boldsymbol{\theta}_\star + \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{X}^\top \boldsymbol{\varepsilon} && (6.2.6)\\
&= \boldsymbol{\theta}_\star - n\lambda \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\theta}_\star + \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{X}^\top \boldsymbol{\varepsilon} && (6.2.7)
\end{aligned}$$

where in (a) we added and subtracted $n\lambda \boldsymbol{I}_d$. Therefore, we have:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\varepsilon}}\left[R(\hat{\boldsymbol{\theta}}_\lambda)\right] - \sigma^2 &= \mathbb{E}_{\boldsymbol{\varepsilon}}\left[||\hat{\boldsymbol{\theta}}_\lambda - \boldsymbol{\theta}_\star||^2_{\boldsymbol{\Sigma}}\right] && (6.2.8)\\
&= (n\lambda)^2 \langle \boldsymbol{\theta}_\star, \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\Sigma} \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\theta}_\star \rangle \\
&\quad + \sigma^2 \operatorname{Tr}\left\{\boldsymbol{X}^\top \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\Sigma} \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1}\right\} && (6.2.9)
\end{aligned}$$

where we used that which allow us to identify:

$$B(\boldsymbol{\theta}_\star, \boldsymbol{X}, \lambda) = (n\lambda)^2 \langle \boldsymbol{\theta}_\star, \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\Sigma} \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\theta}_\star \rangle \tag{6.2.10}$$

$$V(\boldsymbol{X}, \lambda) = \sigma^2 \operatorname{Tr}\left\{\boldsymbol{X}^\top \boldsymbol{X} \left(\boldsymbol{X}^\top \boldsymbol{X} + n\lambda \boldsymbol{I}_d\right)^{-2} \boldsymbol{\Sigma}\right\} \tag{6.2.11}$$

where in the last expression we used the fact that the matrices inside the trace commute. It is also common to see the expressions above written in terms of the data empirical covariance matrix:

$$\hat{\boldsymbol{\Sigma}}_n \coloneqq \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^\top = \frac{1}{n}\boldsymbol{X}^\top \boldsymbol{X} \tag{6.2.12}$$

which reads:

$$B(\boldsymbol{\theta}_\star, \boldsymbol{X}, \lambda) = \lambda^2 \langle \boldsymbol{\theta}_\star, \left(\hat{\boldsymbol{\Sigma}}_n + \lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\Sigma} \left(\hat{\boldsymbol{\Sigma}}_n + \lambda \boldsymbol{I}_d\right)^{-1} \boldsymbol{\theta}_\star \rangle \tag{6.2.13}$$

$$V(\boldsymbol{X}, \lambda, \sigma^2) = \frac{\sigma^2}{n} \operatorname{Tr}\left\{\hat{\boldsymbol{\Sigma}}_n \left(\hat{\boldsymbol{\Sigma}}_n + \lambda \boldsymbol{I}_d\right)^{-2} \boldsymbol{\Sigma}\right\} \tag{6.2.14}$$

Characterising the behaviour of the quantities above becomes, at this point, a random matrix theory problem. Before looking at the general result, let's do a warm up.

6.2.1 Warm-up: ordinary least-squares

We now consider the ordinary least-squares case $\lambda = 0$, which is equivalent to solving a system of n equations with d unknowns:

$$\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\theta} \overset{!}{=} \boldsymbol{X}^\top \boldsymbol{y} \tag{6.2.15}$$

For $n \geq d$, since $\boldsymbol{X}^\top \boldsymbol{X} \in \mathbb{R}^{n\times n}$ is invertible with high-probability, the solution to this problem is unique. For $n < d$, $\boldsymbol{X}^\top \boldsymbol{X}$ is rank defficient, i.e. we have a linear system with more unknowns than variables, and many solutions exist. These solutions can be explicitly written as:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{v}}(\boldsymbol{X}, \boldsymbol{y}) = \boldsymbol{X}^{+}\boldsymbol{y} + \boldsymbol{v} \tag{6.2.16}$$

where $\boldsymbol{X}^+$ is the Moore-Penrose inverse of $\boldsymbol{X}$ and $\boldsymbol{v}$ is any vector in the kernel of $\boldsymbol{X}^\top \boldsymbol{X}$. One particular solution is the *least-norm solution* for which $\boldsymbol{v} = 0$, and corresponds to the $\lambda \to 0^+$ limit of the ridge estimator $\hat{\boldsymbol{\theta}}_\lambda$.

Interestingly, the bias and variance of the ordinary least-squares estimator can be easily computed when $n > d + 1$. Since $\hat{\boldsymbol{\Sigma}}_n$ is almost surely invertible in this case, we have:

$$B(\boldsymbol{\theta}_\star, \boldsymbol{X}, \lambda = 0^+) = 0 \tag{6.2.17}$$

$$V(\boldsymbol{X}, \lambda = 0^+, \sigma^2) = \frac{\sigma^2}{n} \operatorname{Tr}\left\{\hat{\boldsymbol{\Sigma}}_n^{-1} \boldsymbol{\Sigma}\right\} \tag{6.2.18}$$

i.e. the excess risk is fully given by the variance. Writing $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\Sigma}^{1/2}$ for $\boldsymbol{Z}$ a Gaussian i.i.d. matrix with zero mean and unit variance, we have:

$$V(\boldsymbol{X}, \lambda = 0^+, \sigma^2) = \sigma^2 \operatorname{Tr}\left\{(\boldsymbol{Z}^\top \boldsymbol{Z})^{-1}\right\} \tag{6.2.19}$$

Curiously, this is independent of the data covariance matrix. The random matrix $(\boldsymbol{Z}^\top \boldsymbol{Z})^{-1}$ is an inverse Wishart matrix with n degrees-of-freedom, a well-studied random matrix ensemble. In particular, its mean is equal to:

$$\mathbb{E}[(\boldsymbol{Z}^\top \boldsymbol{Z})^{-1}] = \frac{1}{n-d-1}\boldsymbol{I}_d \tag{6.2.20}$$

implying that:

$$\mathbb{E}[V(\boldsymbol{X}, \lambda = 0^+, \sigma^2)] = \frac{\sigma^2 d}{n-d-1} \tag{6.2.21}$$

This result in expectation can be turned into a high-probability bound for the excess risk when $n \to \infty$.[1] Curiously, this expression is fairly close to the excess risk under fixed design setting $\mathbb{E}[R] - \sigma^2 = {}^{\sigma^2 d}\!/_n$, and imply that under the well-specified case the risk converges to zero at a $O(n^{-1})$ rate as $n \to \infty$. An alternative but asymptotic way to get this result is to recognise that this is exactly the $z \to 0$ limit of the Stieltjes transform of the Marchenko-Pastur distribution.

6.3 High-dimensional asymptotics

We now leverage the random matrix theory results discussed in the previous lecture to provide a sharp characterisation of the bias variances in the high-dimensional proportional asymptotics $n, d \to \infty$ with ${}^d\!/_n \to \gamma = \Theta(1)$. Recall that in the previous lecture we have shown that:

Theorem 13. Let $\hat{\boldsymbol{\Sigma}}_n = {}^1\!/_n \boldsymbol{X}^\top \boldsymbol{X} \in \mathbb{R}^{d\times d}$ with $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{\Sigma}^{1/2}$, where $\boldsymbol{Z}$ is a sub-Gaussian matrix with zero mean and unit variance and $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is a positive-definite matrix with eigenvalues $\operatorname{spec}(\boldsymbol{\Sigma}) = \{\lambda_k : k \in [d]\} \subset \mathbb{R}_+$ and bounded operator norm $||\boldsymbol{\Sigma}||_{\mathrm{op}} < C$. Assume that the empirical measure of eigenvalues $\hat{\mu}_n = {}^1\!/_d \sum_{i\in[d]} \delta_{\lambda_i}$ converges (weakly) to a probability distribution μ on $\mathbb{R}_+$ with compact support as $d \to \infty$. Then, for any $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{d\times d}$ with bounded operator norm, the following asymptotic equivalents hold in the proportional limit where $d \to \infty$ with ${}^d\!/_n \to \gamma > 0$:

$$\operatorname{Tr}\{\boldsymbol{A}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}\} \asymp -\frac{1}{z\tilde{s}(z)} \operatorname{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-1}\right\} \tag{6.3.1}$$

$$\begin{aligned}\operatorname{Tr}\{\boldsymbol{A}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}\boldsymbol{B}(\hat{\boldsymbol{\Sigma}}_n - z\boldsymbol{I}_d)^{-1}\} &\asymp \frac{1}{z^2\tilde{s}(z)^2} \operatorname{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-1}\boldsymbol{B}\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-1}\right\} \\ &+ \frac{1}{z^2\tilde{s}(z)^2} \frac{\operatorname{Tr}\left\{\boldsymbol{A}\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-2}\boldsymbol{\Sigma}\right\} \cdot \operatorname{Tr}\left\{\boldsymbol{B}\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(z)}\boldsymbol{I}_d\right)^{-2}\boldsymbol{\Sigma}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + {}^1\!/_{\tilde{s}(z)}\boldsymbol{I}_d)^{-2}\right\}}\end{aligned} \tag{6.3.2}$$

where $\tilde{s}(z)$ is the unique solution of the following self-consistent equation:

$$\frac{1}{\tilde{s}(z)} + z = \frac{\gamma}{d} \operatorname{Tr}\left\{\boldsymbol{\Sigma}(\tilde{s}(z)\boldsymbol{\Sigma} + \boldsymbol{I}_d)^{-1}\right\} \tag{6.3.3}$$

Remark 24. Note that in eq. (6.3.1) and (6.3.2) we used the *asymptotic equivalent notation* "$\asymp$". We say $a_n \asymp b_n$ as $n \to \infty$ if $a_n = \Theta(b_n)$ or equivalently $a_n = O(b_n)$ and $b_n = O(a_n)$ see **??** for a detailed discussion. In particular, this implies that $\lim_{n\to\infty} {}^{a_n}\!/_{b_n} \to 1$. When employing this notation in our context, we are always referring to the proportional asymptotical limit, and when dealing with random quantities the convergence will be almost surely or in probability. When both sides of $\asymp$ are of the same order in n, this implies convergence (a.s. or in probability) of the normalised quantities, e.g. ${}^{a_n}\!/_n \to {}^{b_n}\!/_n$ if $a_n, b_n = \Theta(n)$. Therefore, the main convenience of this notation is to speak of asymptotic limits without having to care for the normalisation of the quantities involved.

Now rewriting the bias and variance in the form of eqs. (6.3.2) and (6.3.2):

$$B(\boldsymbol{\theta}_\star, \boldsymbol{X}, \lambda) = \lambda^2 \operatorname{Tr}\left\{\boldsymbol{\theta}_\star\boldsymbol{\theta}_\star^\top\left(\hat{\boldsymbol{\Sigma}}_n + \lambda\boldsymbol{I}_d\right)^{-1}\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Sigma}}_n + \lambda\boldsymbol{I}_d\right)^{-1}\right\} \tag{6.3.4}$$

$$V(\boldsymbol{X}, \lambda, \sigma^2) = \frac{\sigma^2}{n} \operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Sigma}}_n + \lambda\boldsymbol{I}_d\right)^{-1}\right\} - \frac{\lambda\sigma^2}{n} \operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Sigma}}_n + \lambda\boldsymbol{I}_d\right)^{-2}\right\} \tag{6.3.5}$$

[1] For instance, by showing that $\mathbb{E}\operatorname{Tr}\left[(\boldsymbol{Z}^\top\boldsymbol{Z})^{-1}\right]^2$ is vanishing with $n \to \infty$.

We can readily apply theorem 13. Let's start with the bias. Using eq. (6.3.2) with $\boldsymbol{A} = \boldsymbol{\theta}_\star\boldsymbol{\theta}_\star^\top$ and $\boldsymbol{B} = \boldsymbol{\Sigma}$ evaluated at $z = -\lambda$ give us:

$$\begin{aligned}
B(\boldsymbol{\theta}_\star, \boldsymbol{X}, \lambda) &\asymp \frac{1}{\tilde{s}(-\lambda)^2}\operatorname{Tr}\left\{\boldsymbol{\theta}_\star\boldsymbol{\theta}_\star\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(-\lambda)}\boldsymbol{I}_d\right)^{-2}\boldsymbol{\Sigma}\right\} \\
&\quad + \frac{1}{\tilde{s}(-\lambda)^2}\frac{\operatorname{Tr}\left\{\boldsymbol{\theta}_\star\boldsymbol{\theta}_\star\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(-\lambda)}\boldsymbol{I}_d\right)^{-2}\boldsymbol{\Sigma}\right\}\cdot\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(-\lambda)}\boldsymbol{I}_d\right)^{-2}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + {}^{1}\!/\!{}_{\tilde{s}(-\lambda)}\boldsymbol{I}_d)^{-2}\right\}} \\
&= \frac{1}{\tilde{s}(-\lambda)^2}\boldsymbol{\theta}_\star^\top\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(-\lambda)}\boldsymbol{I}_d\right)^{-2}\boldsymbol{\theta}_\star\left[1 + \frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \frac{1}{\tilde{s}(-\lambda)}\boldsymbol{I}_d\right)^{-2}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + {}^{1}\!/\!{}_{\tilde{s}(-\lambda)}\boldsymbol{I}_d)^{-2}\right\}}\right]
\end{aligned} \tag{6.3.6}$$

It will be convenient to define $\kappa(\lambda) := {}^{1}\!/\!{}_{\tilde{s}(-\lambda)}$ and rewrite the brackets$1 + \frac{x}{1-x} = \frac{1}{1-x}$:

$$B(\boldsymbol{\theta}_\star, \boldsymbol{X}, \lambda) \asymp \frac{\kappa(\lambda)^2\langle\boldsymbol{\theta}_\star, \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\boldsymbol{\theta}_\star\rangle}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}} \tag{6.3.7}$$

The variance is composed of two terms. For the first, we use eq. (6.3.1) with $\boldsymbol{A} = \boldsymbol{\Sigma}$:

$$\frac{\sigma^2}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Sigma}}_n + \lambda\boldsymbol{I}_d\right)^{-1}\right\} \asymp \frac{\sigma^2\kappa(\lambda)}{n\lambda}\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-1}\right\} \tag{6.3.8}$$

while for the second we use eq. (6.3.2) with $\boldsymbol{A} = \boldsymbol{\Sigma}$ and $\boldsymbol{B} = \boldsymbol{I}_d$:

$$\begin{aligned}
\frac{\lambda\sigma^2}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\hat{\boldsymbol{\Sigma}}_n + \lambda\boldsymbol{I}_d\right)^{-2}\right\} &\asymp \frac{\sigma^2\kappa(\lambda)^2}{n\lambda}\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\right\}\left[1 + \frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}}\right] \\
&= \frac{\sigma^2\kappa(\lambda)^2}{n\lambda}\frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\right\}}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}}
\end{aligned} \tag{6.3.9}$$

Putting together:

$$V(\boldsymbol{X}, \lambda, \sigma^2) \asymp \frac{\sigma^2\kappa(\lambda)}{n\lambda}\left[\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-1}\right\} - \kappa(\lambda)\frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\right\}}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}}\right] \tag{6.3.10}$$

Noting that we can write:

$$\begin{aligned}
\kappa\operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d\right)^{-2}\right\} &= \operatorname{Tr}\left\{(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d - \boldsymbol{\Sigma})\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d\right)^{-2}\right\} \\
&= \operatorname{Tr}\left\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d\right)^{-1}\right\} - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d\right)^{-2}\right\}
\end{aligned} \tag{6.3.11}$$

We can equate the denominator and simplify the first term:

$$\begin{aligned}
V(\boldsymbol{X}, \lambda, \sigma^2) &\asymp \frac{\sigma^2\kappa(\lambda)}{n\lambda}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}\frac{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-1}\right\}}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}} \\
&\overset{(a)}{=} \sigma^2\frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}}
\end{aligned} \tag{6.3.12}$$

where in (a) we used the self-consistent equation eq. (6.3.3) to rewrite ${}^{1}\!/\!{}_{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa\boldsymbol{\Sigma})^{-1}\right\} = 1 - {}^{\lambda}\!/\!{}_{\kappa}$. Summarising, this leads to the following result:

Proposition 10 (Asymptotic risk of ridge regression)**.** Under Assumption 1, the asymptotic excess risk of the ridge regressor eq. (6.1.1) converges, in the proportional limit $n, d \to \infty$ with $d/n \to \gamma > 0$ is given by:

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\left[R(\hat{\boldsymbol{\theta}}_\lambda)\right] - \sigma^2 \xrightarrow{a.s.} \mathcal{B}(\boldsymbol{\theta}_\star, \boldsymbol{\Sigma}, \lambda, \gamma) + \mathcal{V}(\boldsymbol{\Sigma}, \lambda, \sigma^2, \gamma), \text{ as } n, d \to \infty \tag{6.3.13}$$

where the asymptotic bias $\mathcal{B}$ and variance $\mathcal{V}$ are given by:

$$\begin{aligned} \mathcal{B}(\boldsymbol{\theta}_\star, \boldsymbol{\Sigma}, \lambda, \gamma) &= \frac{\kappa(\lambda)^2 \langle \boldsymbol{\theta}_\star, \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\boldsymbol{\theta}_\star\rangle}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}} \\ \mathcal{V}(\boldsymbol{\Sigma}, \lambda, \sigma^2, \gamma) &= \sigma^2 \frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d)^{-2}\right\}} \end{aligned} \tag{6.3.14}$$

where $\kappa(\lambda) \geq 0$ is the unique solution of the following self-consistent equation:

$$1 - \frac{\lambda}{\kappa} = \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d)^{-1}\right\} \tag{6.3.15}$$

Remark 25. A few comments on this result are in order.

- Note that without loss of generality we can assume $\boldsymbol{\Sigma}$ to be a diagonal matrix $\boldsymbol{\Sigma} = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$. Therefore, a sufficient condition for different terms in proposition 10 to be well defined in the asymptotic limit is that the empirical spectral measure of $\boldsymbol{\Sigma}$ admits a limit:

$$\frac{1}{d}\sum_{\lambda \in \operatorname{spec}(\boldsymbol{\Sigma})} \delta_\lambda \xrightarrow{\text{weakly}} \mu \tag{6.3.16}$$

  in which case all the traces can be written in terms of expectations with respect to μ, for example:

$$\frac{1}{n}\operatorname{Tr}\{\boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d)^{-1}\} \to \gamma \int \mu(\mathrm{d}t)\frac{t}{t + \kappa}, \text{ as } n, d \xrightarrow{d/n \to \gamma} \infty. \tag{6.3.17}$$

- Nevertheless, we deliberately chose write the equations above in terms of traces since, despite being derived in the proportional regime, the universality of these formulas is remarkable. For instance, Cheng and Montanari (2024) have derived multiplicative, non-asymptotic rates for the limit above under fairly generic assumptions on the covariates. These formulas hold even in the $d \to \infty$ case where $\boldsymbol{\theta}_\star$ can be seen as an element of a Hilbert space and $\boldsymbol{\Sigma}$ a covariance operator, as it was first noted by Bordelon et al. (2020); Cui et al. (2021) and proven in (Misiakiewicz and Saeed, 2024) under some conditions on the tail of the covariance spectrum.

6.3.1 Degrees-of-freedom interpretation

Proposition 10 involve the following two quantities, known in the literature as the *degrees of freedom* of the matrix $\boldsymbol{\Sigma}$:

$$\mathrm{df}_1(\kappa) \coloneqq \operatorname{Tr}\{\boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d\right)^{-1}\} = \sum_{j=1}^{d}\frac{\lambda_j}{\kappa + \lambda_j} \tag{6.3.18}$$

$$\mathrm{df}_2(\kappa) \coloneqq \operatorname{Tr}\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_d\right)^{-2}\} = \sum_{j=1}^{d}\frac{\lambda_j^2}{(\kappa + \lambda_j)^2} \tag{6.3.19}$$

The degrees-of-freedom is a widespread notion in the signal processing and kernel literature, where it is often used as a notion of *effective dimension* when comparing kernel operators defined on infinie dimensional Hilbert spaces, see for example (Zhang, 2005; Caponnetto and De Vito, 2007).

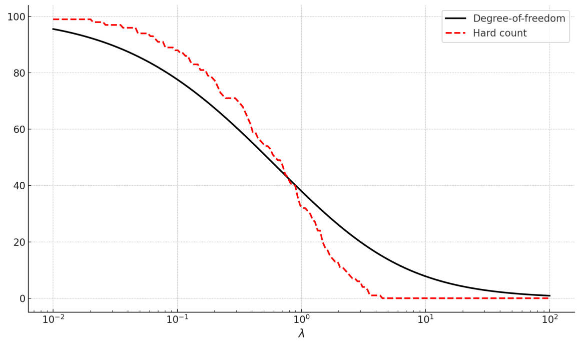

Figure 6.1: Degrees-of-freedom $\mathrm{df}_1(\lambda)$ and hard count $\phi(\lambda)$ as a function of $\lambda \geq 0$ for $\lambda_i \sim \exp(i)$ independently, $i \in [n]$

Note that $\mathrm{df}_1, \mathrm{df}_2$ are strictly decreasing functions of $\kappa \geq 0$, and since $\boldsymbol{\Sigma} \succeq 0$ we have:

$$0 \leq \mathrm{df}_2(\kappa) \leq \mathrm{df}_1(\kappa) \leq \mathrm{rank}(\boldsymbol{\Sigma}) \tag{6.3.20}$$

with equality on the right for $\kappa = 0$. The degrees-of-freedom $\mathrm{df}_1(\kappa), \mathrm{df}_2(\kappa)$ can be seen as a "soft count" of how many eigenvalues are larger than the parameter κ, since eigenvalues $\lambda_j \gg \kappa$ contribute to the sum, while eigenvalues $\lambda_j \ll \kappa$ are shrank. To make this relationship more quantitative, consider the *hard count* of how many eigenvalues of $\boldsymbol{\Sigma}$ are larger than a certain value κ:

$$\phi(\kappa) := \sum_{j=1}^{d} \mathbf{1}_{\lambda_j \geq \kappa} = \#\{k : \lambda_k \geq \kappa\}, \tag{6.3.21}$$

Note $1 - \phi(\kappa)$ is the *cumulative distribution function* (c.d.f.) of the empirical spectral distribution $\hat{\mu}_{\boldsymbol{\Sigma}}$. This can also be written as an integral over $\hat{\mu}_{\boldsymbol{\Sigma}}$:

$$\phi(\kappa) = d \int_{\kappa}^{\infty} \hat{\mu}_{\boldsymbol{\Sigma}}(\mathrm{d}\lambda) = d \int_{\mathbb{R}} \mathbf{1}_{\lambda \geq \kappa}(\lambda)\ \hat{\mu}_{\boldsymbol{\Sigma}}(\mathrm{d}\lambda) \tag{6.3.22}$$

to be compared with:

$$\mathrm{df}_1(\kappa) = d \int_{\mathbb{R}} \frac{\lambda}{\lambda + \kappa} \hat{\mu}_{\boldsymbol{\Sigma}}(\mathrm{d}\lambda) \tag{6.3.23}$$

6.3.2 Equivalent denoising problem

The asymptotic formulas in proposition 10 have an intuitive interpretation in terms of an effectively denoising problem. To see this, consider the problem of retrieving $\boldsymbol{\theta}_\star \in \mathbb{R}^d$ from the following noisy observation:

$$\boldsymbol{u} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{\theta}_\star + \frac{\tau}{\sqrt{n}}\boldsymbol{z}, \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d). \tag{6.3.24}$$

where $\tau > 0$ is the noise standard deviation. Then, the following regularised estimator:

$$\hat{\boldsymbol{\theta}}_{\text{den.}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\arg\min} \left\{ ||\boldsymbol{u} - \boldsymbol{\Sigma}^{1/2}\boldsymbol{\theta}||_2^2 + \kappa_\star ||\boldsymbol{\theta}||_2^2 \right\} \tag{6.3.25}$$

is statistically equivalent to the one in proposition 10 in the proportional high-dimensional limit, as long as we identify:

$$\tau^2 = \sigma^2 + \mathbb{E}_{\boldsymbol{z}}[||\hat{\boldsymbol{\theta}}_{\text{den.}} - \boldsymbol{\theta}_\star||_{\boldsymbol{\Sigma}}^2] \tag{6.3.26}$$

To see this, note that the quadratic problem in eq. (6.3.25) has explicit solution:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{\text{den.}} &= (\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-1}\boldsymbol{\Sigma}^{1/2}\boldsymbol{u} \\
&= (\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-1}\boldsymbol{\Sigma}^{1/2}\left(\boldsymbol{\Sigma}^{1/2}\boldsymbol{\theta}_\star + \frac{\tau}{\sqrt{n}}\boldsymbol{z}\right) \tag{6.3.27} \\
&= \boldsymbol{\theta}_\star - \kappa_\star(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-1}\boldsymbol{\theta}_\star + \frac{\tau}{\sqrt{n}}(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-1}\boldsymbol{\Sigma}^{1/2}\boldsymbol{z} \tag{6.3.28}
\end{aligned}$$

Hence, we have:

$$\mathbb{E}_{\boldsymbol{z}}\left[||\hat{\boldsymbol{\theta}}_{\text{den.}} - \boldsymbol{\theta}_\star||^2_{\boldsymbol{\Sigma}}\right] = \kappa_\star^2\langle \boldsymbol{\theta}_\star, \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\boldsymbol{\theta}_\star\rangle + \frac{\tau^2}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\right\} \tag{6.3.29}$$

It remains to find τ^2. For that, we insert this in eq. (6.3.26) and solve for τ^2 to yield:

$$\tau^2 = \frac{\sigma^2 - \kappa_\star^2\langle \boldsymbol{\theta}_\star, \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\boldsymbol{\theta}_\star\rangle}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\right\}} \tag{6.3.30}$$

Inserting this back into eq. (6.3.29) give us:

$$\mathbb{E}_{\boldsymbol{z}}\left[||\hat{\boldsymbol{\theta}}_{\text{den.}} - \boldsymbol{\theta}_\star||^2_{\boldsymbol{\Sigma}}\right] = \frac{\kappa_\star^2\langle \boldsymbol{\theta}_\star, \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d\right)^{-2}\boldsymbol{\theta}_\star\rangle}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\right\}} + \sigma^2\frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d\right)^{-2}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\right\}} \tag{6.3.31}$$

which is precisely the expression for the asymptotic excess risk from proposition 10.

Remark 26. Two comments are in order:

- The equivalent denoising problem gives a nice interpretation of the different quantities involved in proposition 10. For instance, $\kappa_\star(\lambda)$ appears exactly in the same role as λ in section 6.1, and therefore plays the role of an effective, self-induced ℓ_2-regularisation.
- In fact, the self-induced regularisation is always larger than the original ridge regularisation. This can be seen from studying the behaviour of the self-consistent eq. (6.3.15) for both $\gamma > 1$ and $\gamma < 1$:
  - For $\gamma < 1$ ($d < n$), we have $\operatorname{df}_1(\kappa(\lambda)) \leq d < n$, and therefore the self-consistent eq. (6.3.15) implies:

    $$0 \leq 1 - \frac{\lambda}{\kappa} \leq \gamma. \tag{6.3.32}$$

    Since $\lambda \mapsto \kappa_\star(\lambda)$ is non-decreasing, the solution must satisfy:

    $$\kappa_\star(\lambda) \in \left[\lambda, \frac{\lambda}{1-\gamma}\right], \qquad . \tag{6.3.33}$$

    with in particular $\kappa(0) = 0$.
  - For $\gamma > 1$ ($d > n$), the self-consistent eq. (6.3.15) has a solution $\kappa_\star(0) > 0$ defined by the implicit equation:

    $$\operatorname{df}_1(\kappa(0)) = n \tag{6.3.34}$$

  In other words, the effective regularisation is always larger or equal the original regularisation: $\kappa_\star \geq \lambda$. Since $\kappa \mapsto \operatorname{df}_1(\kappa)$ is a convex map, by Jensen's inequality:

  $$\operatorname{df}_1(\kappa(\lambda)) \leq \frac{\operatorname{Tr}\boldsymbol{\Sigma}}{\kappa(\lambda) + {}^1\!/\!_d\operatorname{Tr}\boldsymbol{\Sigma}} \leq \frac{\operatorname{Tr}\boldsymbol{\Sigma}}{\kappa(\lambda)} \tag{6.3.35}$$

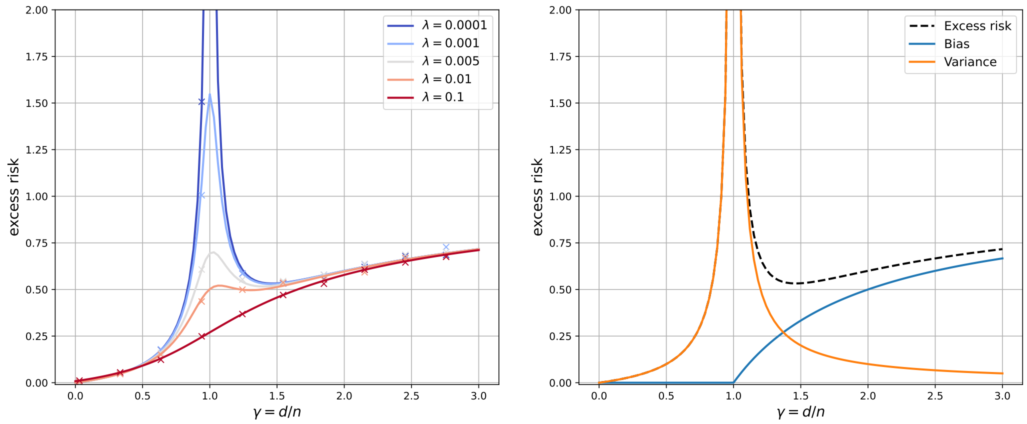

Figure 6.2: Excess risk of ridge regression as a function of $\gamma = {}^{d}/{}_{n}$ for $\sigma^2 = 0.1$, $||\boldsymbol{\theta}_\star||_2^2 = 1$ and isotropic covariates $\boldsymbol{\Sigma} = \boldsymbol{I}_d$. (**Left**) Increasing values of λ. (**Right**) Bias-variance decomposition of the excess risk for the ridge interpolator $\lambda = 0^+$ (a.k.a. ordinary least-squares estimator).

which from eq. (6.3.15) implies that:

$$0 \leq 1 - \frac{\lambda}{\kappa(\lambda)} \leq \frac{\operatorname{Tr} \boldsymbol{\Sigma}}{\kappa(\lambda)} \tag{6.3.36}$$

and again, since $\lambda \mapsto \kappa(\lambda)$ is non-decreasing:

$$\kappa_\star(\lambda) \in \left[\lambda, \lambda + \frac{\operatorname{Tr} \boldsymbol{\Sigma}}{n}\right] \tag{6.3.37}$$

Therefore, in both cases we have an effective regularisation larger than the original ridge regularisation: $\kappa_\star(\lambda) \geq \lambda$.

- Similarly, ${}^{\tau}/{}_{\sqrt{n}}$ plays a similar role to the noise level σ. It is interesting to note that in the effective denoising problem, the effective noise level is itself a function of the excess risk, and is a decreasing function of the number of samples n.

The denoising problem in eq. (6.3.24) and (6.3.25) is also known in the statistical literature as a *sequence model*, see for example (Tsybakov, 2008).

6.3.3 Case study: isotropic covariates

Let's explore our result on possibly the simplest setting, the case of isotropic covariates $\boldsymbol{\Sigma} = \boldsymbol{I}_d$. In this case, the self-consistent eq. (6.3.15) is simply a quadratic equation:

$$1 - \frac{\lambda}{\kappa} = \frac{\gamma}{1+\kappa} \tag{6.3.38}$$

This admits two solutions:

$$\kappa_\pm(\lambda) = \frac{1}{2}\left(\lambda - 1 + \gamma \pm \sqrt{(1-\gamma-\lambda)^2 + 4\lambda}\right) \tag{6.3.39}$$

of which only the positive branch $\kappa_\star(\lambda) := \kappa_+(\lambda)$ is positive for $\lambda \geq 0$. Further, the bias and variance simplify to:

$$\begin{aligned}\mathcal{B}(\boldsymbol{\theta}_\star, \lambda, \gamma) &= \frac{\kappa_\star(\lambda)^2}{\left(1+\kappa_\star(\lambda)\right)^2 - \gamma}||\boldsymbol{\theta}_\star||_2^2 \\ \mathcal{V}(\lambda, \sigma^2, \gamma) &= \frac{\sigma^2\gamma}{\left(1+\kappa_\star(\lambda)\right)^2 - \gamma}\end{aligned} \tag{6.3.40}$$

Figure 6.2 (left) illustrates the asymptotic risk as a function of $\gamma = {}^{d}/_{n}$ for different values on regularisation λ. Note that for small values of λ, the excess risk becomes a non-monotonic function of γ, with a divergence developing at $\gamma = 1$ as $\lambda \to 0^{+}$. On fig. 6.2 (right), we plot the bias and variance contribution to the excess risk in this limit, which shows that this divergence is mainly driven by the variance. This behaviour can be understood from the explicit solution eq. (6.3.39). Note that:

$$\lim_{\lambda \to 0^{+}} \kappa_{\star}(\lambda) = \frac{1}{2}(\gamma - 1 + |\gamma - 1|) = \begin{cases} \gamma - 1 & \gamma > 1 \\ 0 & \gamma \leq 1 \end{cases} \tag{6.3.41}$$

and therefore:

$$\lim_{\lambda \to 0^{+}} \mathcal{B}(\boldsymbol{\theta}_{\star}, \lambda, \gamma) = \begin{cases} 0 & \gamma \leq 1 \\ 1 - \frac{1}{\gamma} & \gamma > 1 \end{cases} \tag{6.3.42}$$

$$\lim_{\lambda \to 0^{+}} \mathcal{V}(\lambda, \sigma^{2}, \gamma) = \begin{cases} \frac{\gamma\sigma^{2}}{1-\gamma} & \gamma < 1 \\ \frac{\sigma^{2}}{\gamma-1} & \gamma > 1 \end{cases} \tag{6.3.43}$$

with a divergence going as $\mathcal{V} \asymp O({}^{1}/_{|1-\gamma|})$ at around $\gamma = 1$.

Remark 27. A few comments are in order.

- Note that the $\gamma < 1$ solution is incredibly close to the non-asymptotic expression we found directly by looking at the ordinary least-squares estimator in section 6.2.1 for $n > d + 1$, with perfect agreement at the limit.

- However, the exact asymptotic formula also give us the behaviour of the least-square solution in the $\gamma \geq 1$ regime. The first curious observation is that at $\gamma = 1$ the variance blows up as $O(|\gamma - 1|^{-1})$, and indeed this is precisely the case where the expected value of the inverse Wishart distribution ceases to exist.

- Consistently to our general discussion in remark 26, for $\gamma > 1$ we have $\kappa_{\star}(\lambda = 0^{+}) = \gamma - 1 > 0$. Recall that from the equivalent denoising problem in eq. (6.3.25), $\kappa_{\star}$ plays the role of the ridge regularisation. This means that we have a non-zero, *self-induced regularisation* in the region $\gamma > 1$. The larger γ, the stronger is this regularisation.

- In the regime $\gamma > 1$, the bias is also non-zero. This is intuitive since we are choosing one (the minimum norm) among all the existing zero loss solutions in this regime. Curiously, the variance also decreases for $\gamma > 1$.

- Although the singularity at $\gamma = 1$ resembles the double descent phenomenon observed in neural networks, the minimum of the risk is achieved at the $\gamma < 1$ region, meaning that it is not beneficial to take $d > n$. In other words, the minimum norm solution overfits.

6.3.4 Case study: the double descent phenomenon

The isotropic case captures the non-monotonic behaviour of the excess risk around the interpolation threshold, but different from neural networks the least-norm solution in this case still overfits in the "overparametrised" regime.

To discuss a model that captures the benign overfitting in neural networks, we need to consider a richer, anisotropic model. Note that one of the limitations of the isotropic case is that the number of parameters in the model $f(\boldsymbol{x}; \boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle$ is the same as the dimensionality of the covariates. Since the covariates are isotropic, increasing the number of parameters is akin to increasing the dimensionality of the covariate space, effectively decreasing the signal-to-noise ratio or sample complexity ${}^{n}/_{d}$ of the problem. This is different from neural networks with more than one-layer, e.g. $f(\boldsymbol{x}; \theta) = \langle \boldsymbol{a}, \sigma(\boldsymbol{W}\boldsymbol{x}) \rangle$

with $\boldsymbol{W} \in \mathbb{R}^{p\times d}$, where we can increase the number of parameters by increasing the width p without changing the input dimension d.

We now introduce a model that seeks to mimic the behaviour of neural networks. Labels are generated from an isotropic latent Gaussian variable:

$$y_i = \langle \boldsymbol{\beta}_\star, \boldsymbol{z}_i\rangle + \xi_i, \qquad \boldsymbol{z}_i \sim \mathcal{N}(0, \boldsymbol{I}_d), \qquad \xi_i \sim \mathcal{N}(0, \tau^2) \tag{6.3.44}$$

However, the statistician does not observe the latent covariates, but rather a noisy projection:

$$\boldsymbol{x}_i = \boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{u}_i \tag{6.3.45}$$

where $\boldsymbol{W} \in \mathbb{R}^{p\times d}$ is a fixed matrix and $\boldsymbol{u}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ i.i.d. In other words, the statistician performs regression on the training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^{p+1} : i = 1, \dots, n\}$. This model, known under the umbrella of *hidden manifold model* (Goldt et al., 2020) or *latent space model* (Hastie et al., 2022), models the well-known *manifold hypothesis* that high-dimensional data depends on a few "relevant features" lying on a lower-dimensional manifold.

This model is a particular case of the one introduced in assumption 1. To see this, note that $(y_i, \boldsymbol{x}_i, \boldsymbol{z}_i)$ are jointly Gaussian variables:

$$\begin{bmatrix} y_i \\ \boldsymbol{x}_i \\ \boldsymbol{z}_i \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \boldsymbol{0}_p \\ \boldsymbol{0}_d \end{bmatrix}, \begin{bmatrix} \tau^2 + ||\boldsymbol{\beta}||_2^2 & (\boldsymbol{W}\boldsymbol{\beta}_\star)^\top & \boldsymbol{\beta}_\star^\top \\ \boldsymbol{W}\boldsymbol{\beta}_\star & \boldsymbol{W}\boldsymbol{W}^\top + \boldsymbol{I}_p & \boldsymbol{W}^\top \\ \boldsymbol{\beta}_\star & \boldsymbol{W} & \boldsymbol{I}_d \end{bmatrix}\right), \qquad \text{i.i.d.} \tag{6.3.46}$$

Therefore, by Gaussian conditioning we have:

$$y_i|\boldsymbol{x}_i \sim \mathcal{N}\left((\boldsymbol{W}^\top\boldsymbol{W} + \boldsymbol{I}_p)^{-1}\boldsymbol{W}\boldsymbol{\beta}_\star, \boldsymbol{x}_i, \tau^2 + \langle \boldsymbol{\beta}_\star, (\boldsymbol{W}^\top\boldsymbol{W} + \boldsymbol{I}_d)^{-1}\boldsymbol{\beta}_\star\rangle\right) \tag{6.3.47}$$

In other words, this model is statistically equivalent to:

$$y_i = \langle \boldsymbol{\theta}_\star, \boldsymbol{x}_i\rangle + \varepsilon_i \tag{6.3.48}$$

with:

$$\boldsymbol{\theta}_\star = (\boldsymbol{W}\boldsymbol{W}^\top + \boldsymbol{I}_p)^{-1}\boldsymbol{W}\boldsymbol{\beta}_\star \tag{6.3.49}$$

and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ is an effective Gaussian noise with variance $\sigma^2 = \tau^2 + \langle \boldsymbol{\beta}_\star, (\boldsymbol{W}^\top\boldsymbol{W} + \boldsymbol{I}_d)^{-1}\boldsymbol{\beta}_\star\rangle$.

Remark 28. From the perspective of the latent model, the effective noise accounts for both for the label noise τ^2 but also for the model misspecification (i.e. the fact that we are fitting in a p dimensional space instead of the d dimensional space the signal lives). In particular, when $p = 0$ we have $\sigma^2 = \tau^2 + ||\boldsymbol{\beta}_\star||_2^2$. Note that beyond the anisotropy, a key difference of the model above is that the target weights in eq. (6.3.49) are correlated with the top right eigenvectors of $\boldsymbol{W}$.

For concreteness, let's look at a simple particular case:

- We assume that $n, p, d \to \infty$ at constant rates $\gamma = {}^{p}/_{n}$ and $\alpha = {}^{n}/_{d}$.
- We assume $\boldsymbol{\beta}_\star \in \mathbb{S}^{d-1}$, i.e. $||\boldsymbol{\beta}_\star||_2 = 1$.
- We assume that $\boldsymbol{W} \in \mathbb{R}^{p\times d}$ is given by:

$$\boldsymbol{W} = \begin{cases} \begin{bmatrix} \sqrt{{}^{p}/_{d}}\,\boldsymbol{I}_d \\ \boldsymbol{0}_{(p-d)\times d} \end{bmatrix} & \text{if } p \geq d \\ \begin{bmatrix} \boldsymbol{I}_p & \boldsymbol{0}_{p\times(d-p)} \end{bmatrix} & \text{if } p < d \end{cases} \tag{6.3.50}$$

Note this implies:

$$\boldsymbol{W}^\top\boldsymbol{W} = \begin{cases} {}^{p}/_{d}\boldsymbol{I}_d & \text{if } p \geq d \\ \begin{bmatrix} \boldsymbol{I}_p & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} & \text{if } p < d \end{cases}, \qquad \boldsymbol{W}\boldsymbol{W}^\top = \begin{cases} \begin{bmatrix} {}^{p}/_{d}\boldsymbol{I}_d & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} & \text{if } p \geq d \\ \boldsymbol{I}_p & \text{if } p < d \end{cases} \tag{6.3.51}$$

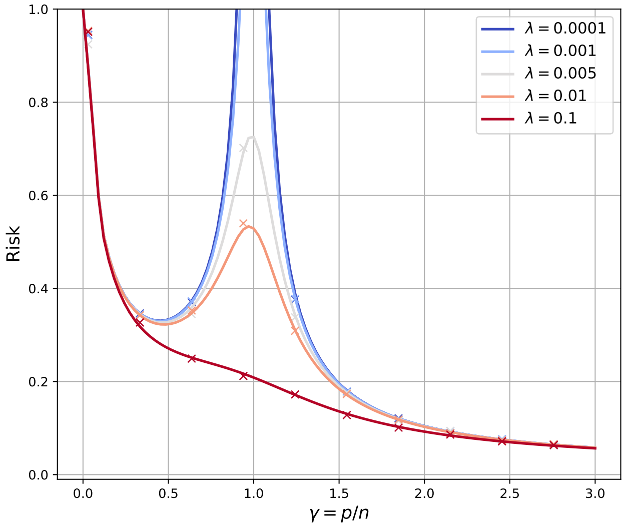

Figure 6.3: Risk of ridge regression as a function of $\gamma = {}^p\!/\!_n$ for the latent space model defined in Section 6.3.4 for $\tau^2 = 0$, $\alpha = {}^n\!/\!_d = 10$. Solid curves show the theoretical result, obtained from solving the self-consistent eq. (6.3.54), and crosses are finite size simulations with $d = 100$.

Remark 29. For the third assumption, any full-rank matrix with fixed Frobenius norm $||\boldsymbol{W}||_{\mathrm{F}}^2 = p$ would be equally good. Equation (6.3.50) is the simplest such example.

Under these assumptions, we have the following simplification:

$$\boldsymbol{\theta}_\star = (\boldsymbol{W}\boldsymbol{W}^\top + \boldsymbol{I}_p)^{-1}\boldsymbol{W}\boldsymbol{\beta}_\star = \boldsymbol{W}(\boldsymbol{W}^\top\boldsymbol{W} + \boldsymbol{I}_d)^{-1}\boldsymbol{\beta}_\star = \frac{\boldsymbol{W}\boldsymbol{\beta}_\star}{1+\alpha\gamma}$$

$$\sigma^2 = \begin{cases} \tau^2 + \frac{1}{1+\alpha\gamma} & \text{if } p \geq d \\ \tau^2 + 1 - {}^{\alpha\gamma}\!/\!_2 & \text{if } p < d \end{cases} \tag{6.3.52}$$

Further, we can also simplify the expression of the degrees-of-freedom:

$$\mathrm{df}_a(\kappa) = \mathrm{Tr}\left\{\boldsymbol{\Sigma}^a(\boldsymbol{\Sigma} + \kappa\boldsymbol{I}_p)^{-a}\right\} = \begin{cases} d\left(\frac{\alpha\gamma+1}{\alpha\gamma+1+\kappa}\right)^a + \frac{p-d}{(1+\kappa)^a} & \text{if } p \geq d \\ p\left(\frac{\alpha\gamma+1}{\alpha\gamma+1+\kappa}\right)^a & \text{if } p < d \end{cases}, \qquad a = 1, 2 \tag{6.3.53}$$

Therefore, in the proportional high-dimensional limit, the self-consistent equation eq. (6.3.15) reads:

$$1 - \frac{\lambda}{\kappa} = \min(\gamma, {}^1\!/\!_\alpha)\frac{\alpha\gamma+1}{\alpha\gamma+\kappa+1} + \left(\gamma - \frac{1}{\alpha}\right)_+ \frac{1}{1+\kappa} \tag{6.3.54}$$

As before, this is a quadratic equation that can be solved explicitly. However, differently from the isotropic case the expressions are cumbersome, so we refrain from writing them here, and focus instead on the discussion of the interpolator $\lambda = 0^+$.

For $\gamma < 1$ $(p < n)$, $\kappa(0) = 0$ is a solution of eq. (6.3.54), and we have $\mathcal{B} = 0$. Then, the risk is completely dominated by the variance:

$$R = \sigma^2 + \mathcal{V} = \sigma^2\left(1 + \frac{\gamma}{1-\gamma}\right) = \begin{cases} \left(\tau^2 + \frac{1}{1+\alpha\gamma}\right)\frac{1}{1-\gamma} & \text{if } p \geq d \\ \left(\tau^2 + 1 - \frac{\alpha\gamma}{2}\right)\frac{1}{1-\gamma} & \text{if } p < d \end{cases} \tag{6.3.55}$$

Note that as $\gamma \to 1^+$, the variance (and hence the risk) diverge as $(1-\gamma)^{-1}$, just as in the isotropic case studied in section 6.3.3. Assuming that $\alpha > 1$, for $\gamma > 1$ the expressions for the bias and the variance read:

$$\mathcal{B}(\gamma, \alpha, \tau^2) = \kappa_\star(0)^2 \frac{A}{1-B}$$
$$\mathcal{V}(\gamma, \alpha, \tau^2) = \left(\tau^2 + \frac{1}{\alpha\gamma+1}\right) \frac{B}{1-B}. \tag{6.3.56}$$

where we have defined:

$$A(\gamma, \alpha, \tau^2) = \langle \boldsymbol{\theta}_\star, \boldsymbol{\Sigma} \left(\boldsymbol{\Sigma} + \kappa(\lambda)\boldsymbol{I}_d\right)^{-2} \boldsymbol{\theta}_\star \rangle = \frac{\alpha\gamma}{(\alpha\gamma+1)(\alpha\gamma+\kappa_\star(0)+1)^2} \tag{6.3.57}$$

$$B(\gamma, \alpha, \tau^2) = \frac{1}{n}\mathrm{df}_2 = \frac{1}{\alpha}\left(\frac{\alpha\gamma+1}{\alpha\gamma+1+\kappa_\star(0)}\right)^2 + \left(\gamma - \frac{1}{\alpha}\right)\frac{1}{(1+\kappa_\star(0))^2} \tag{6.3.58}$$

and $\kappa_\star(0) > 0$ is given by:

$$\kappa_\star(0) = \gamma - 1 - \frac{\alpha\gamma}{2} + \frac{1}{2}\sqrt{(4+\alpha^2)\gamma^2 - 4\gamma} \tag{6.3.59}$$

Figure 6.3 shows the risk of ridge regression for the latent variable model for different values of the regularisation λ. For $\lambda \approx 0^+$ (interpolator), we can clearly see the divergence at $\gamma \to 1$ discussed above, also known as *double descent* or *interpolation peak*. Differently from the isotropic case in section 6.3.3, for $\gamma > 1$, the risk is a decreasing function of γ, meaning that overparametrisation does not hurt generalisation. Moreover, the minimal risk is achieved at large parametrisation $\gamma \to \infty$, when the predictor perfectly interpolates the training data. This phenomenon is known as *benign overfitting*, and is at odds with the classical statistical intuition that interpolating the training data always hurts generalisation.

Remark 30 (Historical note)**.** Both the interpolation peak (Opper et al., 1990; Krogh and Hertz, 1991) and observation that neural networks continue to improve their performance as the number of neurons is increased are quite old (Geman et al., 1992), see (Loog et al., 2020) for a detailed historical discussion. These results were mostly forgotten, and were independently rediscovered in the recent development of machine learning theory driven by the deep learning boom (Zhang et al., 2021). The term “double descent” was coined by Belkin et al. (2019), motivated by different empirical works that observed an interpolation peak in the context of neural networks (Nakkiran et al., 2021; Spigler et al., 2019).

6.4 To go further

6.4.1 Random Features Model

The double descent curve discussed in Section 6.3.4 is not a particular feature of the latent variables model, and manifests in different problems of interest in machine learning. One important example is the *random features model*, where the predictor is given by:

$$f_\theta(\boldsymbol{x}) = \langle \boldsymbol{a}, \sigma(\boldsymbol{W}\boldsymbol{x})\rangle \tag{6.4.1}$$

where $\boldsymbol{W} \in \mathbb{R}^{p\times d}$ is a full-rank matrix, typically taken to be random, and the weights $\boldsymbol{\theta} \in \mathbb{R}^p$ are trained by ridge regression:

$$\hat{\boldsymbol{a}}_\lambda(\boldsymbol{X}, \boldsymbol{y}) = \underset{\boldsymbol{a}\in\mathbb{R}^p}{\arg\min} \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \langle \boldsymbol{a}, \sigma(\boldsymbol{W}\boldsymbol{x})\rangle\right)^2 + \frac{\lambda}{2}||\boldsymbol{a}||_2^2$$
$$= \frac{1}{n}\left(\frac{1}{n}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda \boldsymbol{I}_p\right)^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y} \tag{6.4.2}$$

where we have defined the features matrix $\boldsymbol{\Phi} = \sigma(\boldsymbol{X}\boldsymbol{W}^\top) \in \mathbb{R}^{p\times d}$. Note that this model is equivalent to a two-layer neural network of width p with frozen first layer weights. Indeed, it has been widely studied in the literature as a proxy model for neural networks in the lazy regime.[2]

The challenge of studying this model is that the features matrix $\boldsymbol{\Phi}$ is not a Gaussian matrix, as it was the case for the latent variable model discussed in section 6.3.4. Nevertheless, quite remarkably Theorem 13 can still be applied to characterise the asymptotic properties of the random features model in the proportional regime, thanks to a phenomenon known as *Gaussian universality*, and which was first discussed in this context by Mei and Montanari (2022); Gerace et al. (2020). We now give a brief intuitive discussion, referring the reader interested in the details to the original literature.

The ridge operator in eq. (6.4.2):

$$\boldsymbol{y} \in \mathbb{R}^n \mapsto \frac{1}{n}\left(\frac{1}{n}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda \boldsymbol{I}_p\right)^{-1}\boldsymbol{\Phi}^\top\boldsymbol{y} \tag{6.4.3}$$

projects the response onto the column-space of Image$(\boldsymbol{\Phi}^\top) \subset \mathbb{R}^p$, which is a linear subspace of the feature space. To see this mathematically, denote by $\boldsymbol{\Phi} = \sum_{j=1}^r \lambda_j \boldsymbol{u}_j \boldsymbol{v}_j^\top$ the singular-value decomposition of the features $\boldsymbol{\Phi}$ with $r := \operatorname{rank}(\boldsymbol{\Phi}) \leq \min(n,p)$. Then, we can re-write eq. (6.4.2) as:

$$\hat{\boldsymbol{a}}_\lambda(\Phi, \boldsymbol{y}) = \sum_{j=1}^r \frac{\lambda_j}{\lambda_j^2 + n\lambda}\langle \boldsymbol{u}_j, \boldsymbol{y}\rangle \boldsymbol{v}_j \tag{6.4.4}$$

Therefore, assuming that $y_i = f_\star(\boldsymbol{x}) + \varepsilon_i$ for some target function $f_\star : \mathbb{R}^d \to \mathbb{R}$, the predictor $f(\boldsymbol{x}; \hat{\boldsymbol{a}}_\lambda) = \langle \hat{\boldsymbol{a}}_\lambda, \boldsymbol{\varphi}(\boldsymbol{x})\rangle$ can learn at best a linear component of the target function $f_\star$ in the space spanned by the features $\boldsymbol{\varphi}(\boldsymbol{x})$. For instance, in the vanilla ridge case $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x}$ this would imply that only a linear component of the target can be learned: $f_\star(\boldsymbol{x}) = \langle \boldsymbol{\beta}_\star, \boldsymbol{x}\rangle + f_\star^{>1}(\boldsymbol{x})$, with the non-linear component $f_\star^{>1}$ effectively behaving as part of the label noise ε_i when projected on $\hat{\boldsymbol{a}}_\lambda$. A non-linear feature map $\boldsymbol{\varphi}(\boldsymbol{x})$ therefore allows, in principle, to learn higher order, non-linear components.

To make this discussion more concrete, it is useful to decompose the target function in an orthonormal basis with respect to the distribution of the covariates. Since we assume $\boldsymbol{x}_i \sim \mathcal{N}(0, {}^1\!/_d\boldsymbol{I}_d)$, this is given by the Hermite polynomials:

$$f_\star(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d} c_{\boldsymbol{\alpha}} h_{\boldsymbol{\alpha}}(\boldsymbol{x}) \tag{6.4.5}$$

where $h_{\boldsymbol{\alpha}}(\boldsymbol{x})$ are the Hermite tensors, which form an orthonormal basis of $L^2(\mathbb{R}^d, \gamma_d)$ — where we denote γ_d the Gaussian p.d.f. in dimension d. This basis induces an orthogonal decomposition of $L^2(\mathbb{R}^d, \gamma_d) = \bigoplus_{\ell\geq 1} V_\kappa$, where V_κ is the linear space spanned by polynomials of degree $\ell = |\boldsymbol{\alpha}|$. The coefficients $c_{\boldsymbol{\alpha}}$ quantify how much of the total energy of the target $||f_\star||^2_{\gamma_d} = \sum_{\boldsymbol{\alpha}} c^2_{\boldsymbol{\alpha}}$ lies in each subspace.

Assuming the features $\boldsymbol{\Phi}$ are full-rank $r = \min(n,p)$[3], since the ridge predictor in eq. (6.4.4) spans a linear subspace of dimension r, a naive power counting suggests that to learn the component of the target in subspace V_ℓ requires $r = O(d^\ell)$, with the minimum between the number of samples n and the width p being the bottleneck for approximating V_ℓ. Therefore, in a polynomial scaling regime $n, p = \Theta(d^\ell)$, we can learn at best a degree ℓ polynomial approximation of the target function $f_\star$. In particular, under the proportional asymptotics discussed in Section 6.3, it is enough to consider a linear target function $f_\star(\boldsymbol{x}) = \langle \boldsymbol{\beta}_\star, \boldsymbol{x}\rangle$.

⚠ It is important to keep in mind the discussion in this section is specific to ridge regression.

[2] Although, as we have seen in Lecture 4, random features are only one component of the kernel in the lazy regime, the other being the NTK.

[3] For instance, for $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{0}, {}^1\!/_d\boldsymbol{I}_d)$ and $\boldsymbol{w}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, $\boldsymbol{\Phi} = \sigma(\boldsymbol{X}\boldsymbol{W}^\top)$ will be a full-rank matrix with high-probability with respect to $\boldsymbol{X}, \boldsymbol{W}$.

An important consequence of this discussion is that in the high-dimensional limit a random features map sees the target function at a limited resolution. Considering the expansion of the feature map in the Hermite basis:

$$\varphi_j(\boldsymbol{x}) = \sigma\left(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\right) = \sum_{\ell\geq 0} b_\ell h_\ell(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle) \tag{6.4.6}$$

Its first and second moments are given by:

$$\mathbb{E}_{\boldsymbol{x}}[\sigma\left(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\right)] = b_0 \tag{6.4.7}$$

$$\mathbb{E}_{\boldsymbol{x}}[\sigma\left(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\right)\sigma\left(\langle \boldsymbol{w}_{0,k}, \boldsymbol{x}\rangle\right)] = \sum_{\ell\geq 0} b_\ell^2 \left(\frac{\langle \boldsymbol{w}_j, \boldsymbol{w}_{0,k}\rangle}{d}\right)^\ell \tag{6.4.8}$$

In particular, note that if $\boldsymbol{w}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, with high-probability $1/d\langle \boldsymbol{w}_j, \boldsymbol{w}_k\rangle = O(d^{-1/2})$ for $j \neq k$ and $1/d||\boldsymbol{w}_j||^2 = 1$, meaning that to leading order in d, the features population covariance $\boldsymbol{\Sigma} = \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{\varphi}(\boldsymbol{x})\boldsymbol{\varphi}(\boldsymbol{x})^\top]$ is given by:[4]

$$\boldsymbol{\Sigma} = b_0^2 \boldsymbol{1}_p \boldsymbol{1}_p^\top + b_1^2 \frac{\boldsymbol{W}_0 \boldsymbol{W}_0^\top}{d} + b_\star^2 \boldsymbol{I}_p + o_{\mathbb{P},d}(1) \tag{6.4.9}$$

where we have defined:

$$b_\star^2 = \sum_{\ell\geq 2} b_\ell^2 = \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\sigma(z)^2\right] - b_0^2 - b_1^2 \tag{6.4.10}$$

This implies that under the proportional high-dimensional limit, the features $\boldsymbol{\varphi}(\boldsymbol{x}) = \sigma(\boldsymbol{W}_0\boldsymbol{x})$ have the same first and second moments as the following Gaussian covariates:

$$\boldsymbol{g} = b_0 \boldsymbol{1}_p + b_1 \boldsymbol{W}_0 \boldsymbol{x} + b_\star \boldsymbol{u}, \qquad \boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p) \tag{6.4.11}$$

This is exactly the latent variable model we studied in section 6.3.4! This suggests that in the proportional high-dimensional limit, we can trade the study of the original non-linear random features model in eq. (6.4.2) for the study of an equivalent Gaussian covariate model. This is an instance of a more general universality phenomenon, known as a Gaussian equivalence. We refer the reader for the original works for a full discussion of Gaussian universality in this context (Mei and Montanari, 2022; Gerace et al., 2020; Goldt et al., 2022; Hu and Lu, 2022; Montanari and Saeed, 2022)

6.4.2 Benign overfitting

Whether a predictor can benignantly overfit the data will crucially depends on the geometry of the covariates. Intuitively, benign overfitting means that the predictor is able to fit the signal in the data (so it can generalise) while also fitting the noise (so it can interpolate). This is only possible when the signal is strong, and lies in a sufficiently small subspace of the covariate space, ensuring that there is enough "room" left to accommodate the noise.

Note that this is precisely the case in the latent variable model discussed above: the signal $\boldsymbol{\beta}_\star$ is in $\mathbb{R}^d$ while the predictor and the covariates are in $\boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^p$. For $p \geq d$, the covariate covariance $\boldsymbol{\Sigma}$ has a block structure, with the directions corresponding to the d dimensional signal space being $O(p/d)$, while the remaining $p - d$ directions being $O(1)$. Hence, when the system is overparametrised $p \gg d$, the covariance has a small but strong signal block, with a weak but large orthogonal block.

General conditions for benign overfitting in the context of ridge regression were first studied by Bartlett et al. (2020); Tsigler and Bartlett (2023). Here, we will closely follow an argument from Misiakiewicz and Montanari (2024) which is based on the formulas from proposition 10.

[4] Note this is normalised such that $\operatorname{Tr}\boldsymbol{\Sigma} = \Theta(p)$

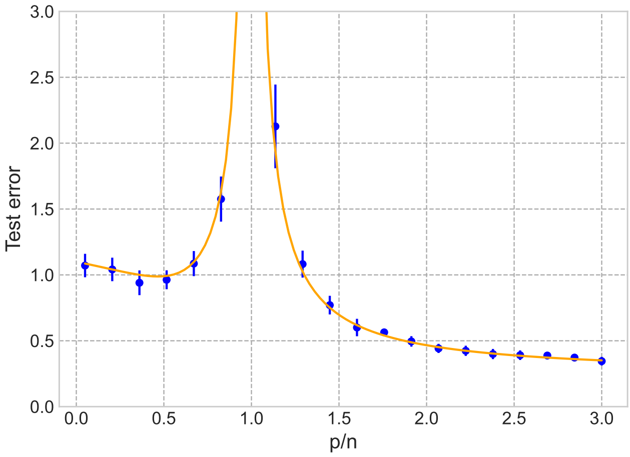

Figure 6.4: Test error of the random features ridge regressor eq. (6.4.2) as function of $\gamma = {}^{p}\!/_{n}$ at fixed ${}^{n}\!/_{d} = 1.5$ and $\lambda = 0^{+}$. The solid line denote the theoretical result obtained from proposition 10 under the Gaussian equivalent covariance eq. (6.4.9), and points denote finite-size simulations with $d = 500$.

⚠ Technically, the discussion that follows requires a non-asymptotic control of the risk, which goes beyond our proportional asymptotic result proposition 10. Nevertheless, as previously discussed, one can derive non-asymptotic multiplicative rates for the deterministic equivalents in theorem 13. We refer the interested reader to (Cheng and Montanari, 2024; Misiakiewicz and Saeed, 2024; Defilippis et al., 2024).

From eq. (6.3.20) and the self-consistent equation for $\kappa_\star$, we know that for any $\lambda \geq 0$:

$$\operatorname{Tr} \boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2} \leq \operatorname{Tr} \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_p)^{-1} = n\left(1 - \frac{\lambda}{\kappa_\star(\lambda)}\right) \leq n \tag{6.4.12}$$

We now assume that actually we have a tighter control of $\mathrm{d}f_2(\kappa_\star)$:

$$\operatorname{Tr} \boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2} \leq n\left(1 - \frac{1}{c_\star}\right) \tag{6.4.13}$$

for a constant $c_\star \in (1, \infty)$ which is problem dependent. This implies an immediate upper-bound on the bias and variance:

$$\begin{aligned}\mathcal{B}(\boldsymbol{\theta}_\star, \boldsymbol{\Sigma}, \lambda, \gamma) &= \frac{\kappa(\lambda)^2 \langle \boldsymbol{\theta}_\star, \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d\right)^{-2} \boldsymbol{\theta}_\star \rangle}{1 - \frac{1}{n}\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\right\}} \leq c_\star \kappa_\star^2 \langle \boldsymbol{\theta}_\star, \boldsymbol{\Sigma}\left(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d\right)^{-2} \boldsymbol{\theta}_\star \rangle \\ \mathcal{V}(\boldsymbol{\Sigma}, \lambda, \sigma^2, \gamma) &= \sigma^2 \frac{\operatorname{Tr}\left\{\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d\right)^{-2}\right\}}{n - \operatorname{Tr}\left\{\boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\right\}} \leq \frac{c_\star \sigma^2}{n} \operatorname{Tr} \boldsymbol{\Sigma}^2(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-2}\end{aligned} \tag{6.4.14}$$

reducing the problem to the study of $\mathrm{f}_2(\kappa_\star)$ and the quadratic form in the bias. The key idea to control these terms is to split the target and the covariance into a low- and a high-frequency part:

$$\boldsymbol{\Sigma} = \sum_{\ell=1}^{k_\star} \lambda_\ell \boldsymbol{v}_\ell \boldsymbol{v}_\ell^\top + \sum_{\ell=k_\star+1}^{d} \lambda_\ell \boldsymbol{v}_\ell \boldsymbol{v}_\ell^\top := \boldsymbol{\Sigma}_{\leq k_\star} + \boldsymbol{\Sigma}_{>k_\star} \tag{6.4.15}$$

$$\boldsymbol{\theta}_\star = \sum_{\ell=1}^{k_\star} \langle \boldsymbol{\theta}_\star, \boldsymbol{v}_\ell \rangle \boldsymbol{v}_\ell + \sum_{\ell=k_\star+1}^{d} \langle \boldsymbol{\theta}_\star, \boldsymbol{v}_\ell \rangle \boldsymbol{v}_\ell := \boldsymbol{\theta}_{\star,\leq k_\star} + \boldsymbol{\theta}_{\star,>k_\star} \tag{6.4.16}$$

where, motivated by the discussion in section 6.3.1 we take the cut-off to be:

$$k_\star = \max\{k : \lambda_k \geq \kappa_\star\} \tag{6.4.17}$$

Using this decomposition allow us to further upper-bound the bias and variance. Starting with the variance, we have:

$$\begin{aligned}
\operatorname{Tr} \boldsymbol{\Sigma}^2(\boldsymbol{\Sigma}+\kappa_\star \boldsymbol{I}_d)^{-2} &= \sum_{\ell=1}^{k_\star} \frac{\lambda_\ell^2}{(\lambda_\ell+\kappa_\star)^2} + \sum_{\ell=k_\star+1}^{d} \frac{\lambda_\ell^2}{(\lambda_\ell+\kappa_\star)^2} \\
&\leq \sum_{\ell=1}^{k_\star} \frac{\lambda_\ell^2}{\lambda_\ell^2} + \sum_{\ell=k_\star+1}^{d} \frac{\lambda_\ell^2}{\kappa_\star^2} && (6.4.18) \\
&\leq k_\star + \sum_{\ell=k_\star+1}^{d} \frac{\lambda_\ell^2}{\lambda_{k_\star+1}^2} && (6.4.19)
\end{aligned}$$

where in the last inequality we have used that by construction $\lambda_{k+1} < \kappa_\star$ to upper-bound the second term. While this is a perfectly good bound, to bring it closer to the result derived by Bartlett et al. (2020), we can use the self-consistent equations to further rewrite the second term. Indeed, we can bound:

$$\begin{aligned}
n \geq \operatorname{Tr} \boldsymbol{\Sigma}(\boldsymbol{\Sigma}+\kappa_\star \boldsymbol{I}_d)^{-1} &= \sum_{\ell=1}^{k_\star} \frac{\lambda_\ell}{\lambda_\ell+\kappa_\star} + \sum_{\ell=k_\star+1}^{d} \frac{\lambda_\ell}{\lambda_\ell+\kappa_\star} \\
&\geq \sum_{\ell=1}^{k_\star} \frac{\lambda_\ell}{2\lambda_\ell} + \sum_{\ell=k_\star+1}^{d} \frac{\lambda_\ell}{2\kappa_\star} \\
&= \frac{k_\star}{2} + \sum_{\ell=k_\star+1}^{d} \frac{\lambda_\ell}{2\lambda_{k_\star+1}} && (6.4.20)
\end{aligned}$$

In particular, since $k_\star \geq 1$ we have $n \geq \sum_{\ell>k_\star} \lambda_\ell / 2\lambda_{k_\star+1}$ and so the variance term can be bounded by:

$$\begin{aligned}
\mathcal{V} &\leq \frac{c_\star\sigma^2}{n}\left[k_\star + \sum_{\ell=k_\star+1}^{d} \frac{\lambda_\ell^2}{\lambda_{k_\star+1}^2}\right] \\
&\leq c_\star\sigma^2\left[\frac{k_\star}{n} + \frac{\sigma_{k_\star^2}}{\sigma_{k_\star+1}^2}\frac{\left(\sum_{\ell>k_\star}\lambda_\ell\right)^2}{\sum_{\ell>k_\star}\lambda_\ell^2}n\right] && (6.4.21)
\end{aligned}$$

The ratio:

$$r(k) = \frac{\left(\sum_{\ell>k}\lambda_\ell\right)^2}{\sum_{\ell>k}\lambda_\ell^2} \qquad (6.4.22)$$

is a classical quantity is Physics and Random Matrix Theory, where $r(1)$ is *spectrum participation ratio*. It measures how "localised" is the spectrum of a matrix. Indeed, if all eigenvalues are equal, $r(1) = 1$, while if only a few eigenvalues dominate the spectrum, we have $r(1) \ll 1$. Here, we define it with respect to the tail of the spectrum.

Similarly, we can bound the bias:

$$\begin{aligned}
\mathcal{B} &\leq c_\star \sum_{\ell=1}^{d} \frac{\kappa_\star^2\lambda_\ell}{(\lambda_\ell+\kappa_\star)^2}\langle\boldsymbol{\theta}_\star, \boldsymbol{v}_\ell\rangle^2 && (6.4.23) \\
&\leq c_\star\left[\sum_{\ell=1}^{k_\star}\frac{\kappa_\star^2}{\lambda_k}\langle\boldsymbol{\theta}_\star, \boldsymbol{v}_\ell\rangle^2 + \sum_{\ell=k_\star+1}^{d}\lambda_\ell\langle\boldsymbol{\theta}_\star, \boldsymbol{v}_\ell\rangle^2\right] && (6.4.24) \\
&\leq c_\star\left[\lambda_{k_\star}^2||\boldsymbol{\theta}_{\star,\leq k_\star}||^2_{\boldsymbol{\Sigma}^{-1}} + ||\boldsymbol{\theta}_{\star,>k_\star}||^2_{\boldsymbol{\Sigma}}\right] && (6.4.25)
\end{aligned}$$

Summarising, we have the following upper-bounds:

$$\mathcal{V} \leq c_\star \sigma^2 \left[\frac{k_\star}{n} + \frac{\sigma_{k_\star^2}}{\sigma^2_{k_\star+1}} \frac{\left(\sum_{\ell > k_\star} \lambda_\ell\right)^2}{\sum_{\ell > k_\star} \lambda_\ell^2} n \right] \tag{6.4.26}$$

$$\mathcal{B} \leq c_\star \left[\lambda^2_{k_\star} ||\boldsymbol{\theta}_{\star, \leq k_\star}||^2_{\boldsymbol{\Sigma}^{-1}} + ||\boldsymbol{\theta}_{\star, > k_\star}||^2_{\boldsymbol{\Sigma}} \right] \tag{6.4.27}$$

$$\frac{k_\star}{2} + \sum_{\ell = k_\star + 1} \frac{\lambda_\ell}{2\lambda_{k_\star+1}} \leq n \tag{6.4.28}$$

with $c_\star \in (1, \infty)$. Note this is valid for all $\lambda \geq 0$. We can also obtain a lower-bound for $k_\star$ by assuming $\lambda \leq {}^{\kappa_\star}/_2$, which can always be satisfied by taking λ small enough, since $\kappa_\star(\lambda) \geq \lambda \geq 0$. In this case:

$$\operatorname{Tr} \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-1} = n \left(1 - \frac{\lambda}{\kappa_\star}\right) \geq n \tag{6.4.29}$$

Hence:

$$\begin{aligned} n \leq \operatorname{Tr} \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \kappa_\star \boldsymbol{I}_d)^{-1} &= \sum_{\ell=1}^{k_\star} \frac{\lambda_\ell}{\lambda_\ell + \kappa_\star} + \sum_{\ell = k_\star+1}^{d} \frac{\lambda_\ell}{\lambda_\ell + \kappa_\star} \\ &\leq \sum_{\ell=1}^{k_\star} \frac{2\lambda_\ell}{\lambda_\ell} + \sum_{\ell=k_\star+1}^{d} \frac{2\lambda_\ell}{\kappa_\star} \\ &= 2k_\star + 2 \sum_{\ell = k_\star+1} \frac{\lambda_\ell}{\lambda_{k_\star+1}} \end{aligned} \tag{6.4.30}$$

This lower-bound implies that $k_\star \to \infty$ as $n \to \infty$. Together, this provides everything we need to characterise when overfitting is benign.

Assuming that $\lambda_k \to 0$ as $k \to \infty$, we have that:

$$\mathcal{B} \to 0 \qquad \text{as } n \to \infty \tag{6.4.31}$$

provided that $||\boldsymbol{\theta}_\star||_{\boldsymbol{\Sigma}^{-1}} < \infty$. A sufficient condition is that $\boldsymbol{\theta}_\star$ only with finitely many directions of the covariance $\boldsymbol{\Sigma}$. What about the variance? Assuming that ${}^{\lambda_{k_\star}}/_{\lambda_{k_\star+1}} < \infty$ as $k_\star \to \infty$, in order for the variance to vanish we need that:

$$\frac{k_\star}{n} \to 0, \qquad \frac{n}{r(k_\star)} \to 0. \tag{6.4.32}$$

Chapter 7

Implicit bias of descent algorithms

7.1 Motivation

So far, our discussion has focused mostly on the approximation and estimation properties of statistical models. However, one central ingredient in modern machine learning has been missing from our discussion: the training algorithm.

For strictly convex problems (such as the ridge regression with $\lambda > 0$), the training algorithm indeed plays a minor role: the minimiser is unique, and any "good enough" algorithm should convergence to it — with the only difference being the computational efficiency. The situation, however, is very different for non-convex problems where more than one minima can be present. In this case, different choices of algorithm (including here initialisation and choices of hyperparameter, e.g. learning rate schedule, mini-batch sampling, stopping time, etc.) can lead to different estimators with potentially drastic differences in the generalisation performance, see e.g. (Liu et al., 2020b). This can be particularly striking in problems with more than a global minima, for instance in overparametrised networks which can be trained down to achieve zero training loss (i.e. perfectly interpolate the training data). In this case, different algorithms might converge to different interpolators, all achieving zero loss, but which can have different generalisation performances — see fig. 7.1 for an illustration.

The fact that different choices of algorithm lead to predictors with different statistical properties is known as the *implicit bias* of algorithms.[1] Characterising the implicit biases of widely used algorithms such as gradient descent and stochastic gradient descent is an active research field. In this lecture, we will study two of the simplest examples.

The discussion that follows was greatly inspired by a post in the blog of Francis Bach, written by Pillaud-Vivien and Pesme (2022), as well as Scott Pesme's PhD manuscript.

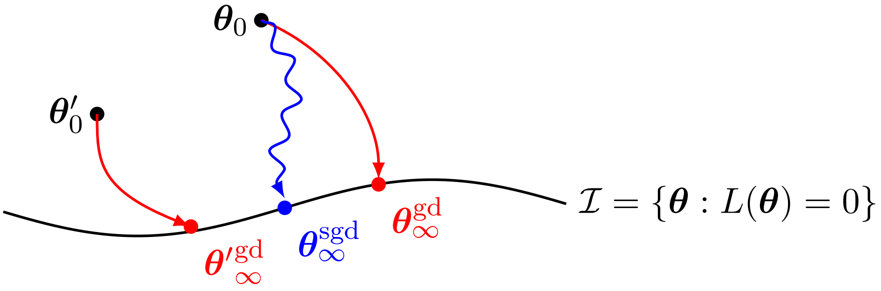
$\boldsymbol{\theta}_0$
$\boldsymbol{\theta}'_0$
$\boldsymbol{\theta}^{\mathrm{gd}}_\infty$
$\boldsymbol{\theta}^{\mathrm{sgd}}_\infty$
$\boldsymbol{\theta}'^{\mathrm{gd}}_\infty$
$\mathcal{I} = \{\boldsymbol{\theta} : L(\boldsymbol{\theta}) = 0\}$

Figure 7.1

[1] Perhaps worryingly, the naming suggests that these are biases which often get forgotten.

7.2 Implicit bias in least-squares regression

As motivated above, algorithmic bias is mostly relevant in problems for which the loss has more than a single critical point. Arguably the simplest such problem is least-squares regression in the overspecified regime ($d > n$), which we now review.

7.2.1 Recap of OLS

Consider a supervised learning regression problem with training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^{d+1} : i \in [n]\}$. Assume for simplicity that the data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is full-rank. The lest-squares problem is defined as:

$$\min_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}) := \frac{1}{2n}\sum_{i=1}^{n}(y_i - \langle \boldsymbol{\theta}, \boldsymbol{x}_i\rangle)^2 \tag{7.2.1}$$

The Hessian of the empirical risk is simply the empirical covariance of the data:

$$\nabla^2 \hat{R}_n(\boldsymbol{\theta}) = \frac{1}{n}\boldsymbol{X}^\top \boldsymbol{X} =: \hat{\boldsymbol{\Sigma}}_n \tag{7.2.2}$$

which is positive semi-definite. This implies that $\hat{R}_n$ is a convex function of $\boldsymbol{\theta} \in \mathbb{R}^d$. However, it is not necessarily strictly convex: this is only the case for $n \geq d$ for which $\boldsymbol{X}^\top \boldsymbol{X}$ is positive-definite. For $n < d$, the empirical risk is just convex, meaning it can have more than one global minimum. How do global minimum look like?

Overdetermined regime ($n \geq d$) — The empirical risk is non-negative $\hat{R}_n(\boldsymbol{\theta}) \geq 0$, with equality $\hat{R}_n(\boldsymbol{\theta}) = 0$ for a predictor that perfectly interpolates the training data $\boldsymbol{X}\hat{\boldsymbol{\theta}} = \boldsymbol{y}$. Finding an interpolator is equivalent to solving a system of n equations with d independent variables (since we assumed $\boldsymbol{X}$ full rank). In particular, this is not possible when $n > d$: there are more equations than variables, and the system is *overdetermined*. Nevertheless, the unique solution to the OLS problem in eq. (7.2.1) is given by:

$$\hat{\boldsymbol{\theta}}_{\text{ols}}(\boldsymbol{X}, \boldsymbol{y}) = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top y, \qquad n \geq d \tag{7.2.3}$$

which, for $n > d$ has strictly positive training error $\hat{R}_n(\hat{\boldsymbol{\theta}}_{\text{ols}}) > 0$. For $n = d$, $\boldsymbol{X}$ becomes invertible and the unique interpolator given by $\hat{\boldsymbol{\theta}}_{\text{ols}}(\boldsymbol{X}, \boldsymbol{y}) = \boldsymbol{X}^{-1}\boldsymbol{y}$.

Overdetermined regime ($n < d$) — It is easy to see that for $n < d$ the solution:

$$\hat{\boldsymbol{\theta}}_{\text{ols}}(\boldsymbol{X}, \boldsymbol{y}) = \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X})^{-1}\boldsymbol{y}, \qquad n \leq d \tag{7.2.4}$$

is an interpolator. Note it is precisely the limit of the ridge regressor when we take $\lambda \to 0^+$. However, this interpolator is not unique. Indeed, for any vector $\boldsymbol{v} \in \text{Ker}(\boldsymbol{X})$,[2] the sum $\hat{\boldsymbol{\theta}}_{\text{ols}} + \boldsymbol{v}$ is also an interpolator, since by definition $\boldsymbol{X}\boldsymbol{v} = \boldsymbol{0}$. This means that the space of interpolators define an affine space, which can be explicitly written as:

$$\begin{aligned}\mathcal{I} &= \{\hat{\boldsymbol{\theta}} \in \mathbb{R}^d : \boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{y}\} = \{\boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{y} + \boldsymbol{v} : \boldsymbol{v} \in \text{Ker}(\boldsymbol{X})\} \\ &= \hat{\boldsymbol{\theta}}_{\text{ols}} + \text{Ker}(\boldsymbol{X})\end{aligned} \tag{7.2.5}$$

The interpolator $\hat{\boldsymbol{\theta}}_{\text{ols}}$ is also known as the *minimum-ℓ_2 norm solution*, since by the triangular inequality:

$$||\hat{\boldsymbol{\theta}}_{\text{ols}} + \boldsymbol{v}||_2^2 \geq ||\hat{\boldsymbol{\theta}}_{\text{ols}}||_2^2 + ||\boldsymbol{v}||_2^2 \tag{7.2.6}$$

[2] Recall that for $n < d$, $\text{Ker}(\boldsymbol{X}) \neq \mathtt{0}$ and that it is isomorphic to a $d - n$ space.

which implies it has the smallest Euclidean norm of all elements of $\mathcal{I}$. In other words, it is the solution of:

$$\min ||\boldsymbol{\theta}||_2^2, \text{ such that } \boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{y} \tag{7.2.7}$$

In learning theory, it is common to use norms $||\cdot||_p$ as a proxy for the complexity of a hypothesis class, and a common objective is to find predictors with low complexity, which are often associated with better generalisation. From this perspective, the minimum-ℓ_2 norm predictor $\hat{\boldsymbol{\theta}}_{\text{ols}}$ is the least complex interpolator with respect to $||\cdot||_2$.

⚠ We don't mean that the minimum-ℓ_2 norm predictor is the one with best generalisation in $\mathcal{I}$. Indeed, whether $\hat{\boldsymbol{\theta}}_{\text{ols}}$ generalises better than, e.g. the minimum-ℓ_1 predictor, will crucially depends on the target function $f_\star(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}]$.

Now consider minimising eq. (7.2.1) in the underdetermined regime $n < d$ using a descent-based algorithm, such as gradient descent. To which interpolator in $\mathcal{I}$ will it converge to? We start by answering this question in the context of gradient flow.

7.2.2 Implicit bias of gradient flow

Consider the gradient flow algorithm for the OLS problem:

$$\dot{\boldsymbol{\theta}}(t) = -\nabla\hat{R}_n(\boldsymbol{\theta}) = \frac{1}{n}\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}) \tag{7.2.8}$$

with initial condition $\boldsymbol{\theta}(t) = \boldsymbol{\theta}_0$. This is a system of d coupled ODEs. To decouple them, let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ denote the SDV of $\boldsymbol{X}$:

$$\dot{\boldsymbol{\theta}}(t) = \frac{1}{n}\boldsymbol{V}\left(\boldsymbol{D}^\top\boldsymbol{U}^\top\boldsymbol{y} - \boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{V}^\top\boldsymbol{\theta}(t)\right) \tag{7.2.9}$$

Hence, defining $\boldsymbol{\beta} = \boldsymbol{V}^\top\boldsymbol{\theta}(t)$ and $\tilde{\boldsymbol{y}} = \boldsymbol{U}^\top\boldsymbol{y}$, we have an autonomous system:

$$\dot{\beta}_j(t) = \frac{\sigma_j}{n}(\tilde{y}_j - \sigma_j\beta_j) \tag{7.2.10}$$

Note that, depending on $\text{rank}(\boldsymbol{X}) = \min(n, d)$, the equation above take a different shape.

Overspecified regime ($n \geq d$) — For $n \geq d$, $\text{rank}(\boldsymbol{X}) = d$ and hence $\sigma_j > 0$ for all $j \in [d]$. The solution is therefore given by:

$$\beta_j(t) = \frac{\tilde{y}_j}{\sigma_j} + e^{-\frac{\sigma_j^2 t}{n}}\left(\beta_{0,j} - \frac{\tilde{y}_j}{\sigma_j}\right) \tag{7.2.11}$$

where $\boldsymbol{\beta}_0 = \boldsymbol{V}^\top\boldsymbol{\theta}_0$. Or, in terms of $\boldsymbol{\theta}$:

$$\begin{aligned}\boldsymbol{\theta}(t) &= \sum_{j=1}^{d}\left[\frac{\langle\boldsymbol{u}_j, \boldsymbol{y}\rangle}{\sigma_j}\boldsymbol{v}_j + e^{-\frac{\sigma_j^2 t}{n}}\left(\boldsymbol{\theta}_0 - \frac{\langle\boldsymbol{u}_j, \boldsymbol{y}\rangle}{\sigma_j}\boldsymbol{v}_j\right)\right] \\ &= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y} + \sum_{j=1}^{d} e^{-\frac{\sigma_j^2 t}{n}}\left(\boldsymbol{\theta}_0 - \frac{\langle\boldsymbol{u}_j, \boldsymbol{y}\rangle}{\sigma_j}\boldsymbol{v}_j\right)\end{aligned} \tag{7.2.12}$$

As expected, this converges exponentially fast to $\hat{\boldsymbol{\theta}}_{\text{ols}}$ in eq. (7.2.3) — no surprise, this is the unique global minimum since the problem is strictly convex in this regime.

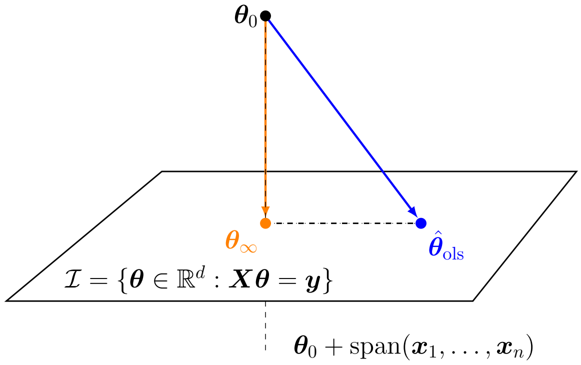
$\boldsymbol{\theta}_0$
$\boldsymbol{\theta}_\infty$
$\hat{\boldsymbol{\theta}}_{\text{ols}}$
$\mathcal{I} = \{\boldsymbol{\theta} \in \mathbb{R}^d : \boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{y}\}$
$\boldsymbol{\theta}_0 + \text{span}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$

Figure 7.2: Implicit bias of gradient flow for in the underdetermined regime $n < d$. Gradient flow converges to the orthogonal projection of the initial condition on the linear subspace of interpolators.

Underspecified regime $(n < d)$ — In this case, $\text{rank}(\boldsymbol{X}) = n$. Assuming that $\sigma_j \geq 0$ is arranged in non-increasing order, we have $\sigma_j = 0$ for all $j > n$. This means that the solution of eq. (7.2.10) is now given by:

$$\beta_j(t) = \begin{cases} \frac{\tilde{y}_j}{\sigma_j} + e^{-\frac{\sigma_j^2 t}{n}} \left(\beta_{0,j} - \frac{\tilde{y}_j}{\sigma_j}\right) & \text{for } 1 \leq j \leq n \\ \beta_{0,j} & \text{for } n < j \leq d \end{cases} \tag{7.2.13}$$

which means that only the components of the initial condition which are in $\text{span}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) \simeq \mathbb{R}^n$ change under the gradient flow, with the remaining components remaining constant. More precisely, recall that we can always decompose $\mathbb{R}^d = \text{range}(\boldsymbol{X}^\top) \oplus \text{ker}(\boldsymbol{X})$, which induces the following orthogonal decomposition of the initial condition $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_{0,\parallel} + \boldsymbol{\theta}_{0,\perp}$ with $\boldsymbol{\theta}_{0,\parallel} \in \text{range}(\boldsymbol{X}^\top)$ and $\boldsymbol{\theta}_{0,\perp} \in \text{ker}(\boldsymbol{X})$. Therefore, we can write:

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}_{0,\perp} + \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{y} + \sum_{j=1}^{d} e^{-\frac{\sigma_j^2 t}{n}} \left(\boldsymbol{\theta}_{0,\parallel} - \frac{\langle \boldsymbol{u}_j, \boldsymbol{y}\rangle}{\sigma_j}\boldsymbol{v}_j\right) \tag{7.2.14}$$

In the long-time limit, this leads to:

$$\begin{aligned} \boldsymbol{\theta}_\infty \coloneqq \lim_{t\to\infty} \boldsymbol{\theta}(t) &= \boldsymbol{\theta}_{0,\perp} + \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{y} \\ &= \boldsymbol{\theta}_{0,\perp} + \hat{\boldsymbol{\theta}}_{\text{ols}}(\boldsymbol{X}, \boldsymbol{y}) \end{aligned} \tag{7.2.15}$$

This corresponds to a particular interpolator $\mathcal{I}$: the one which is closest (in Euclidean distance) to the initial condition $\boldsymbol{\theta}_0$:

$$\boldsymbol{\theta}_\infty = \underset{\boldsymbol{\theta}\in\mathcal{I}}{\arg\min}\ ||\boldsymbol{\theta}_0 - \boldsymbol{\theta}||_2^2 \tag{7.2.16}$$

which is only equal to $\hat{\boldsymbol{\theta}}_{\text{ols}}$ if $\boldsymbol{\theta}_0 = \boldsymbol{0}$! See Figure 7.2 for an illustration.

7.2.3 Other descent algorithms

A natural question is whether gradient descent (the discretisation of gradient flow), or other descent-based algorithms, such as SGD, have different implicit biases than the one discussed above. Consider mini-batch SGD:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \nabla \hat{R}_{b_k}(\boldsymbol{\theta}_k), \qquad \text{with} \qquad \hat{R}_{b_k}(\boldsymbol{\theta}) = \frac{1}{2b}\sum_{i\in b_k}(y_i - \langle \boldsymbol{\theta}, \boldsymbol{x}_i\rangle)^2 \tag{7.2.17}$$

where η_k is a learning rate schedule and $b_k \subset [n]$ is a mini-batch of size $b := |b_k| \leq n$, which here we assume is sampled uniformly and independently at each iteration. Note that GD is a particular case of the above, obtained by choosing $b = n$ at every step. The key observation is the gradient of the empirical risk always lies in the span of the training data:

$$\nabla \hat{R}_{b_k}(\boldsymbol{\theta}_k) = -\frac{1}{|b_k|}\sum_{i\in b_k}(y_i - \langle \boldsymbol{\theta}_k, \boldsymbol{x}_i\rangle)\boldsymbol{x}_i \in \text{span}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \tag{7.2.18}$$

Therefore, for any iterate k, we have:

$$\boldsymbol{\theta}_k \in \boldsymbol{\theta}_0 + \text{span}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \tag{7.2.19}$$

which is an (affine) n-dimensional space. Assuming that the iterates of this algorithm converge to an interpolator $\boldsymbol{\theta}_k \to \boldsymbol{\theta}_\infty \in \mathcal{I}$[3], and recalling our characterisation of $\mathcal{I}$ from eq. (7.2.5):[4]

$$\mathcal{I} = \hat{\boldsymbol{\theta}}_{\text{ols}} + \ker(\boldsymbol{X}) = \hat{\boldsymbol{\theta}}_\star + \text{span}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^\perp. \tag{7.2.20}$$

where in the second equality we used that, by definition, the kernel is orthogonal to the image. This implies that:

$$\boldsymbol{\theta}_\infty - \boldsymbol{\theta}_0 \perp \boldsymbol{\theta}_\infty - \hat{\boldsymbol{\theta}}_{\text{ols}} \tag{7.2.21}$$

See fig. 7.2 for an illustration. Therefore:

$$||\hat{\boldsymbol{\theta}}_{\text{ols}} - \boldsymbol{\theta}_0||_2^2 = ||\hat{\boldsymbol{\theta}}_{\text{ols}} - \boldsymbol{\theta}_\infty||_2^2 + ||\hat{\boldsymbol{\theta}}_\infty - \boldsymbol{\theta}_0||_2^2 \geq ||\hat{\boldsymbol{\theta}}_\infty - \boldsymbol{\theta}_0||_2^2 \tag{7.2.22}$$

In other words, we must have:

$$\boldsymbol{\theta}_\infty = \underset{\boldsymbol{\theta}\in\mathcal{I}}{\arg\min}\ ||\boldsymbol{\theta}_0 - \boldsymbol{\theta}||_2^2 \tag{7.2.23}$$

Which is the same implicit bias of gradient flow. This means that, for the least-squares problem and from the perspective of the implicit bias, there is no difference between GD and SGD. A similar conclusion can be proven for more complex descent algorithms, such as SGD with momentum. However, it is important to stress that this property is specific to OLS, and is a direct consequence of the fact that the gradient lives in the span of the data, which for $n < d$ is a linear subspace of $\mathbb{R}^d$. As we will see next, more complicated architectures or loss functions will behave differently.

7.3 Implicit bias of diagonal linear networks

While the least-squares problem is instructive, it is too simple, missing some important features related to the way neural network are parametrised. We now turn our attention to another problem that remains simple enough so that an explicit mathematical analysis can be carried out, but that has additional structure that will shed light on an important aspect of implicit bias: the interplay between the network architecture and the training algorithm.

Consider the following model, known as a linear *diagonal neural network*:

$$f(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) = \sum_{j=1}^{d} u_j v_j x_j = \langle \boldsymbol{u} \odot \boldsymbol{v}, \boldsymbol{x}\rangle \tag{7.3.1}$$

where the $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ are two vectors, and $\odot$ denote the entry-wise product between vectors, also known as the Hadamard product. This hypothesis can be equivalently seen in two ways:

[3] This can be proven under small enough constant learning rate $\eta_k = \eta$ in the $d > n$ regime. We won't do it here, but it is a reasonable assumption given that the problem is strongly convex when restricted to $\boldsymbol{\theta}_k \in \boldsymbol{\theta}_0 + \text{span}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$.

[4] More generally, note this is true for any reference interpolator $\hat{\boldsymbol{\theta}}_\star \in \mathcal{I}$.

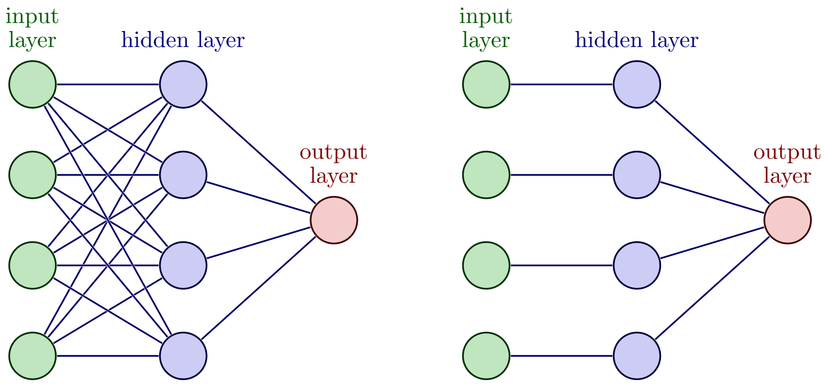

Figure 7.3: (**Left**) Standard fully connected two-layer neural network with $d = p = 4$ hidden-units and a single output. (**Right**) Diagonal two-layer neural network with $d = 4$ input nodes and a single output.

- First, it can be seen as a linear predictor $f(\boldsymbol{x}, \boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle$ with a particular parametrisation of the weights $\boldsymbol{\theta} = \boldsymbol{u} \odot \boldsymbol{v} \in \mathbb{R}^d$. Therefore, the statistical properties of our predictor are the same as the linear predictor. However, as we will going to see below, the fact that we parametrise it in a particular way has important consequences for optimisation.

- Alternatively, it can see it as a two-layer neural network $f(\boldsymbol{x}; \boldsymbol{\theta}) = \langle \boldsymbol{u}, \sigma(\boldsymbol{W}\boldsymbol{x})$ with linear activation $\sigma(z) = z$, $p = d$ hidden-units and first-layer weights which are constrained to be a diagonal matrix $\boldsymbol{W}_{jk} = v_j \delta_{jk}$ — hence the name diagonal neural network. See fig. 7.3 for an illustration.

This model was introduced in Woodworth et al. (2020), and was motivated by previous work on implicit bias of algorithms in matrix factorisation Gunasekar et al. (2017).

Remark 31 (Symmetry)**.** This parametrisation has an obvious rescaling symmetry. Indeed, for any non-zero vector $\boldsymbol{b} \in \mathbb{R}^d$ we have:

$$f(\boldsymbol{x}; \boldsymbol{b} \odot \boldsymbol{u}, \boldsymbol{b}^{-1} \odot \boldsymbol{v}) = f(\boldsymbol{x}; \boldsymbol{u}, \boldsymbol{v}) \tag{7.3.2}$$

where the inverse is applied component-wise. This symmetry will play an important role in what follows.

As in the previous section, we will be interested in the empirical risk minimisation problem over a batch of training data:

$$\min_{\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d} L(\boldsymbol{u}, \boldsymbol{v}) := \frac{1}{2n} \sum_{i=1}^{n} (y_i - \langle \boldsymbol{u} \odot \boldsymbol{v}, \boldsymbol{x}_i \rangle)^2 \tag{7.3.3}$$

The first important observation is that although the empirical risk is a convex function of the product $\boldsymbol{\theta} = \boldsymbol{u} \odot \boldsymbol{v}$, it is not a convex function of $\boldsymbol{u}, \boldsymbol{v}$! This means that in principle $L(\boldsymbol{u}, \boldsymbol{v})$ can have several critical points. To get some intuition, it is instructive to look at the $n = d = 1$ case, where the loss read:

$$\min_{(u,v) \in \mathbb{R}^2} L(u, v) := \frac{1}{2}(y - uv)^2 \tag{7.3.4}$$

where without loss of generality we set $x = 1$. It is clear that the global minima manifold is given by the hyperbola $uv = 1$. Since $\partial_u L = -(y - uv)v$ and $\partial_v L = -(y - uv)u$, the only other critical point is the saddle-point $u = v = 1$. See fig. 7.4.

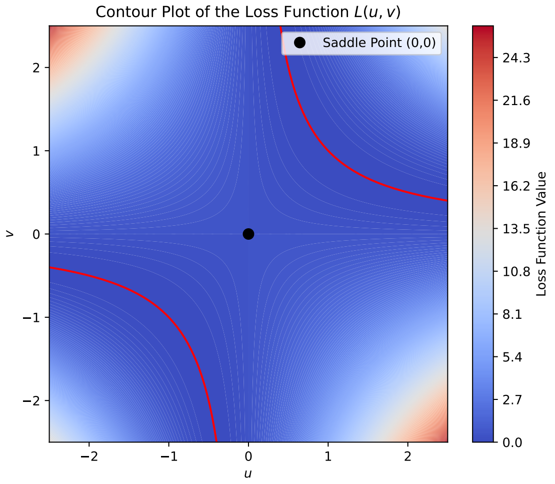

Figure 7.4: Contour plot of the loss $L(u, v) = {}^1\!/\!{}_2(y - uv)^2$ for $y = 1$. The interpolation manifold is given by the hyperbola $uv = 1$, shown in solid red. The only other critical point is a saddle point at $u = v = 0$.

Remark 32. The phenomenology above is quite general. Indeed, by introducing a constraint $\boldsymbol{\theta} = \boldsymbol{u} \odot \boldsymbol{v}$ on the overparametrised space $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{2d}$ we introduce a symmetry that effectively degenerates the global minima of the original problem into a full orbit of the group.

As we will see, when $n, d > 1$, things are similar, except that we have additional saddle-points. For concreteness, we focus the discussion that follows to the underspecified regime $d > n$.

7.3.1 Properties of the landscape

As a starting point, we show that all the extremisers of $\hat{R}_n$ must be global minima. To see this, consider the map:

$$\boldsymbol{\theta} : (\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{2d} \mapsto \boldsymbol{u} \odot \boldsymbol{v} \in \mathbb{R}^d., \tag{7.3.5}$$

and note that $L(\boldsymbol{u}, \boldsymbol{v}) = (\hat{R}_n \circ \boldsymbol{\theta})(\boldsymbol{u}, \boldsymbol{v})$. Since the map $\boldsymbol{\theta}$ is differentiable at $\mathbb{R}^{2d}$, by the chain rule we have:

$$\nabla_{(\boldsymbol{u},\boldsymbol{v})} L(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{u}, \boldsymbol{v})^\top \nabla_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}) \tag{7.3.6}$$

where $\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{d \times 2d}$ is the Jacobian matrix of $\boldsymbol{\theta}$ at $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{2d}$. This implies that critical points of $L(\boldsymbol{u}, \boldsymbol{v})$ are either critical points of $\hat{R}(\boldsymbol{\theta})$ — which by convexity are global minima — or elements or in the kernel of the Jacobian. Since we have:

$$\partial_{u_j} \theta_k = v_k \delta_{jk}, \qquad \partial_{v_j} \theta_k = u_k \delta_{jk} \tag{7.3.7}$$

the Jacobian is a full-rank matrix $\text{rank}(\boldsymbol{J}_{\boldsymbol{\theta}}(\boldsymbol{u}, \boldsymbol{v})) = d$ whenever the coordinates of $\boldsymbol{u}$ and $\boldsymbol{v}$ are not simultaneously zero (check this!). In other words, the critical points of L which are not critical points of $\hat{R}_n$ necessarely have $(u_j, v_j) = (0, 0)$ for some $j \in [d]$. However, these points cannot be extremisers of the loss, since the loss is flat across these directions. To see this, consider $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^d$ such that $(u_j, v_j) = (0, 0)$ for some $j \in [d]$. Then, it is easy to check that the $L(\boldsymbol{u}, \boldsymbol{v} + \alpha \boldsymbol{e}_j) = L(\boldsymbol{u}, \boldsymbol{v})$ where $\boldsymbol{e}_j \in \mathbb{R}^d$ is the basis vector (similarly for $\boldsymbol{u} + \alpha \boldsymbol{e}_j$, by symmetry). Recalling that by definition local minima (maxima) are such that moving in their neighbourhood increase (decrease) the loss, we can conclude that the only local extremisers of $L(\boldsymbol{u}, \boldsymbol{v})$ are the global minima (i.e. the extremisers of $\hat{R}_n$).

Recall from section 7.2 that in the underspecified regime $n < d$, these are the interpolators $\mathcal{I} = \{\boldsymbol{\theta} \in \mathbb{R}^d : \boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{y}\}$. Therefore, accounting for the symmetry eq. (7.3.2), we can write the global minimisers of $L(\boldsymbol{u}, \boldsymbol{v})$ as:

$$\underset{(\boldsymbol{u},\boldsymbol{v})\in\mathbb{R}^{2d}}{\arg\min}\ L(\boldsymbol{u}, \boldsymbol{v}) = \left\{ \left(\operatorname{sign}(\boldsymbol{\theta}_\star)\sqrt{|\boldsymbol{\theta}_\star|} \odot \boldsymbol{b}, \sqrt{|\boldsymbol{\theta}_\star|} \odot \boldsymbol{b}^{-1} \right) : \boldsymbol{\theta}_\star \in \mathcal{I}, \boldsymbol{b} \in \mathbb{R}^d, \boldsymbol{b} \neq \boldsymbol{0} \right\}. \tag{7.3.8}$$

where the non-linear operations are understood entry-wise. Now let's look at the remaining critical points. From the discussion above, we know these are saddle-points $(\boldsymbol{u}_c, \boldsymbol{v}_c) \in \mathbb{R}^{2d}$ with a subset of coordinates $(u_{c,j}, v_{c,j}) = (0, 0)$, and hence $\theta_{c,j} := \theta_j(\boldsymbol{u}_c, \boldsymbol{v}_c) = 0$. Consider the gradient of L:

$$\begin{aligned}
\nabla_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{v}) &= -\frac{1}{n}\boldsymbol{v} \odot \boldsymbol{X}^\top \left(\boldsymbol{y} - \boldsymbol{X}^\top(\boldsymbol{u} \odot \boldsymbol{v}) \right) = \boldsymbol{v} \odot \nabla_{\boldsymbol{\theta}} \hat{R}(\boldsymbol{u} \odot \boldsymbol{v}) \\
\nabla_{\boldsymbol{v}} L(\boldsymbol{u}, \boldsymbol{v}) &= -\frac{1}{n}\boldsymbol{u} \odot \boldsymbol{X}^\top \left(\boldsymbol{y} - \boldsymbol{X}^\top(\boldsymbol{u} \odot \boldsymbol{v}) \right) = \boldsymbol{u} \odot \nabla_{\boldsymbol{\theta}} \hat{R}(\boldsymbol{u} \odot \boldsymbol{v})
\end{aligned} \tag{7.3.9}$$

To have a critical point, we need $\nabla_{\boldsymbol{u}} L = \nabla_{\boldsymbol{v}} L = \boldsymbol{0}$. An obvious choice is $(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{0}, \boldsymbol{0})$. For $(\boldsymbol{u}, \boldsymbol{v}) \neq (\boldsymbol{0}, \boldsymbol{0})$, since a subset of the coordinates is zero, this is automatically satisfied for these coordinates. In the remaining coordinates, we effectively have an OLS problem. To formalise this, define the support of a vector:

$$\operatorname{supp}(\boldsymbol{\theta}) = \{j \in [d] : \theta_j \neq 0\}. \tag{7.3.10}$$

Then, the previous condition can be written $\nabla_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta})_j = 0$ for $j \notin \operatorname{supp}(\boldsymbol{\theta}_c)$. This allow us to characterise the saddle-points of L directly in terms of $\hat{R}_n$ as:

$$\boldsymbol{\theta}_c \in \underset{\theta_j = 0 \text{ for } j \notin \operatorname{supp}(\theta_c)}{\arg\min} \hat{R}_n(\boldsymbol{\theta}) \tag{7.3.11}$$

⚠ Note that due to the symmetry eq. (7.3.2), all points in the orbit of a saddle-point $(\boldsymbol{u}_c, \boldsymbol{v}_c) \in \mathbb{R}^{2d}$ are also saddle points.

We can summarise the properties of the loss landscape in the following proposition.

Proposition 11. The critical points of the empirical risk eq. (7.3.3) in the underspecified regime $n > d$ are characterised as follows.

- The only extremisers of $L(\boldsymbol{u}, \boldsymbol{v})$ are global minima, which can be written as:

$$\underset{(\boldsymbol{u},\boldsymbol{v})\in\mathbb{R}^{2d}}{\arg\min}\ L(\boldsymbol{u}, \boldsymbol{v}) = \left\{ \left(\operatorname{sign}(\boldsymbol{\theta}_\star)\sqrt{|\boldsymbol{\theta}_\star|} \odot \boldsymbol{b}, \sqrt{|\boldsymbol{\theta}_\star|} \odot \boldsymbol{b}^{-1} \right) : \boldsymbol{\theta}_\star \in \mathcal{I}, \boldsymbol{b} \in \mathbb{R}^d, \boldsymbol{b} \neq \boldsymbol{0} \right\}. \tag{7.3.12}$$

  where $\mathcal{I} = \{\boldsymbol{\theta} \in \mathbb{R}^d : \boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{y}\}$ are the set of OLS interpolators.

- All the other critical points are saddle-points, given by $\boldsymbol{\theta}_c = \boldsymbol{\theta}(\boldsymbol{u}_c, \boldsymbol{v}_c)$ such that:

$$\boldsymbol{\theta}_c \in \underset{\theta_j = 0 \text{ for } j \notin \operatorname{supp}(\theta_c)}{\arg\min} \hat{R}_n(\boldsymbol{\theta}) \tag{7.3.13}$$

7.3.2 The implicit bias of gradient flow

We now consider gradient flow for the linear diagonal network:

$$\dot{\boldsymbol{u}}(t) = -\nabla_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{n}\boldsymbol{v}(t) \odot \boldsymbol{X}^\top \boldsymbol{r}(t) \tag{7.3.14}$$

$$\dot{\boldsymbol{v}}(t) = -\nabla_{\boldsymbol{v}} L(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{n}\boldsymbol{u}(t) \odot \boldsymbol{X}^\top \boldsymbol{r}(t) \tag{7.3.15}$$

where for notational convenience we defined the displacement vector $\boldsymbol{r}(t) = \boldsymbol{y} - \boldsymbol{X}^\top(\boldsymbol{u}(t) \odot \boldsymbol{v}(t))$. In particular, we would like to compare how this dynamics differs from the gradient flow on $\hat{R}_n$ in eq. (7.2.8). For that, we can attempt to reconstruct the trajectory $\boldsymbol{\theta}(t) = \boldsymbol{u}(t) \odot \boldsymbol{v}(t)$ from the above:

$$\begin{aligned} \dot{\boldsymbol{\theta}} &= \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{u} \odot \boldsymbol{v}) = \dot{u} \odot \boldsymbol{v} + \boldsymbol{u} \odot \dot{\boldsymbol{v}} \\ &= \frac{1}{n}\boldsymbol{X}^\top \boldsymbol{r}(t) \odot (\boldsymbol{v}^2 + \boldsymbol{u}^2) \end{aligned} \tag{7.3.16}$$

where the squares are understood entry-wise. In principle, it is not clear how to close this equation in $\boldsymbol{\theta}$. In order to achieve this, we note that the following quantity:

$$\boldsymbol{I}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^2 - \boldsymbol{v}^2}{2} \tag{7.3.17}$$

is an integral of motion, i.e.:

$$\dot{\boldsymbol{I}} = \boldsymbol{u} \odot \dot{\boldsymbol{u}} - \boldsymbol{v} \odot \dot{\boldsymbol{v}} = 0 \tag{7.3.18}$$

it is therefore conserved along the flow. With this, we can rewrite:

$$\boldsymbol{u}^2 + \boldsymbol{v}^2 = \sqrt{\boldsymbol{\theta}^2 + \boldsymbol{I}^2} \tag{7.3.19}$$

where we remind the reader the square-root is understood component-wise. Therefore:

$$\begin{aligned} \dot{\boldsymbol{\theta}}(t) &= -\frac{2}{n}\boldsymbol{X}^\top \boldsymbol{r}(t) \odot \sqrt{\boldsymbol{\theta}(t)^2 + \boldsymbol{I}^2} \\ &= -2\sqrt{\boldsymbol{\theta}(t)^2 + \boldsymbol{I}^2} \odot \nabla_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}) \end{aligned} \tag{7.3.20}$$

which is remarkably different from the simple gradient flow $\dot{\boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta})$. Unfortunately, solving this non-convex flow explicitly is hard. In order to characterise the implicit bias, we will make the following rewriting. Define the potential function:

$$\phi_{\boldsymbol{I}}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{j=1}^{d}\left(\theta_j \sinh^{-1}\left(\frac{\theta_j}{I_j}\right) - \sqrt{\theta_j^2 + I_j^2} + I_j\right) \tag{7.3.21}$$

Noting that:

$$\partial_{\theta_j}\phi = \sinh^{-1}\left(\frac{\theta_j}{I_j}\right) \tag{7.3.22}$$

It is easy to check that we can rewrite the flow in eq. (7.3.20) as:

$$\frac{\mathrm{d}\nabla_{\boldsymbol{\theta}}\phi}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}} \hat{R}_n(\boldsymbol{\theta}) \tag{7.3.23}$$

The reader who is familiar with convex optimisation will recognise this is a *Mirror descent flow*, the zero learning rate limit of the *Mirror descent* algorithm. Mirror descent is an algorithm for constrained optimisation that generalises the Euclidean projection into the constraint set to other geometries, in this case implicitly defined by the Bregman divergence of a potential function ϕ:

$$D_\phi(\boldsymbol{\theta}, \boldsymbol{\theta}') = \phi(\boldsymbol{\theta}) - \phi(\boldsymbol{\theta}') - \langle \nabla\phi(\boldsymbol{\theta}), \boldsymbol{\theta} - \boldsymbol{\theta}' \rangle \tag{7.3.24}$$

Note that the Euclidean projection is recovered with the quadratic potential $\phi(\boldsymbol{\theta}) = {}^1\!/\!_2||\boldsymbol{\theta}||_2^2$. The advantage of writing this as a mirror flow is that, in the case in which the potential is a strictly convex function, which is the case here:

$$\partial_{\theta_j}\partial_{\theta_k}\phi_{\boldsymbol{I}}(\boldsymbol{\theta}) = \frac{1}{2\sqrt{\theta_j^2 + I_j^2}}\delta_{jk} > 0, \tag{7.3.25}$$

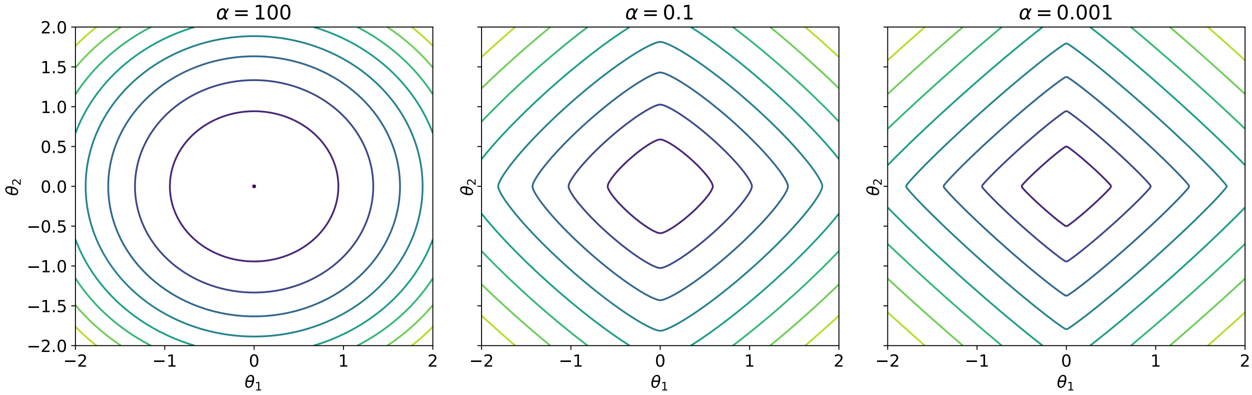

Figure 7.5: Contour plot of the potential ϕ_α defined in eq. (7.3.21) with $\boldsymbol{I} = \alpha^2 \mathbf{1}_d$ in the (θ_1, θ_2)-plane for $\alpha \in \{10^{-3}, 0.1, 100\}$

strong guarantees from convex optimisation apply. For instance, it can be proved that the flow in eq. (7.3.23) converge $\boldsymbol{\theta} \to \boldsymbol{\theta}_\infty$, and that in the regime $d > n$ this must necessarily be an interpolator $\boldsymbol{\theta}_\infty \in \mathcal{I}$ of $\hat{R}_n$. By the same argument as in section 7.2.2, since $\nabla_{\boldsymbol{\theta}} \hat{R}_n \in \operatorname{span}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$, eq. (7.3.23) implies that for any $t > 0$:

$$\nabla_{\boldsymbol{\theta}} \phi_{\boldsymbol{I}}(\boldsymbol{\theta}(t)) \in \nabla_{\boldsymbol{\theta}} \phi_{\boldsymbol{I}}(\boldsymbol{\theta}_0) + \operatorname{span}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \tag{7.3.26}$$

Moreover, we can repeat the argument in eq. (7.2.22) with the Bregman divergence instead of the Euclidean norm. Indeed, for any interpolator $\hat{\boldsymbol{\theta}} \in \mathcal{I}$

$$\begin{aligned} D_\phi(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_0) &= D_\phi(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}_\infty) + D_\phi(\boldsymbol{\theta}_\infty, \boldsymbol{\theta}_0) + \langle \nabla \phi_{\boldsymbol{I}}(\boldsymbol{\theta}_\infty) - \underbrace{\nabla \phi_{\boldsymbol{I}}(\boldsymbol{\theta}_0), \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_\infty \rangle}_{=0} \\ &\geq D_\phi(\boldsymbol{\theta}_\infty, \boldsymbol{\theta}_0) \end{aligned} \tag{7.3.27}$$

which implies that:

$$\boldsymbol{\theta}_\infty = \underset{\boldsymbol{\theta} \in \mathcal{I}}{\arg\min}\, D_\phi(\boldsymbol{\theta}, \boldsymbol{\theta}_0) \tag{7.3.28}$$

Or in words: GD converges to the interpolator which has minimal Bregman divergence to the initial condition. Note that, as expected, this result recovers eq. (7.2.23) when $\phi(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_2^2$. Interpolators of the type eq. (7.3.28) can be quite different from the ones in eq. (7.2.23) — we now discuss these differences in detail.

Role of the initialisation — To study the differences in a concrete setting, we consider the initial condition $\boldsymbol{\theta}_0 = \mathbf{0}$. Recall from section 7.2.2 that in this case, gradient flow on the square loss converges to $\hat{\boldsymbol{\theta}}_{\text{ols}} = \boldsymbol{X}^\top (\boldsymbol{X}\boldsymbol{X}^\top)^{-1} \boldsymbol{y}$, the minimum ℓ_2-norm interpolator. What about the diagonal neural network flow? The initial condition $\boldsymbol{\theta}_0 = \boldsymbol{u}_0 \odot \boldsymbol{v}_0 = \mathbf{0}$ corresponds to a full family of initial conditions on $(\boldsymbol{u}_0, \boldsymbol{v}_0)$. Again, for concreteness we focus our attention on a particular subclass of solutions parametrised by a single scalar parameter $\alpha \geq 0$:

$$\boldsymbol{u}_0 = \sqrt{2}\alpha \mathbf{1}_d, \qquad \boldsymbol{v}_0 = \mathbf{0} \tag{7.3.29}$$

In this case, the integral of motion is given by:

$$\boldsymbol{I}(\boldsymbol{u}_0, \boldsymbol{v}_0) = \alpha^2 \mathbf{1}_d \tag{7.3.30}$$

A simple asymptotic expansion allow us to see how the potential $\phi_\alpha \equiv \phi_{\alpha^2 \mathbf{1}_d}$ looks like when α is small or large:

$$\phi_\alpha(\boldsymbol{\theta}) \underset{\alpha\to 0^+}{\asymp} \log(^1/_\alpha)\cdot ||\boldsymbol{\theta}||_1, \qquad \phi_\alpha(\boldsymbol{\theta}) \underset{\alpha\to\infty}{\asymp} \frac{1}{4\alpha^2}||\boldsymbol{\theta}||_2^2. \tag{7.3.31}$$

In other words, α continuously interpolate $\boldsymbol{\phi}_\alpha$ between the ℓ_1 and the ℓ_2 norm! See fig. 7.5 for a contour plot of ϕ_α at different values of α. Consequently, from eq. (7.3.28) it can be shown that the solution $\boldsymbol{\theta}_\infty(\alpha)$ will inherit a similar implicit bias:

$$\boldsymbol{\theta}_\infty(\alpha) \underset{\alpha\to 0^+}{\to} \underset{\boldsymbol{\theta}\in\mathcal{I}}{\arg\min}\, ||\boldsymbol{\theta}||_1, \qquad \boldsymbol{\theta}_\infty(\alpha) \underset{\alpha\to\infty}{\to} \underset{\boldsymbol{\theta}\in\mathcal{I}}{\arg\min}\, ||\boldsymbol{\theta}||_2 \tag{7.3.32}$$

⚠ Proving this result requires showing uniform convergence of ϕ_α as $\alpha \to 0/\infty$ on a compact of $\mathbb{R}^d$ and that $\boldsymbol{\theta}_\infty(\alpha)$ is bounded for all $\alpha \geq 0$. A detailed proof can be found in Proposition 6 of Pesme (2024).

Remark 33 (Relationship to lazy training)**.** Recall from Lecture 4 our discussion on how the scale of initialisation determines whether our network behaves as a kernel method or whether it learn features. Kernel methods are just linear models on Hilbert space, and therefore the implicit bias of GF/GF/SGD for kernels is the same as least-squares, which for vanishing initialisation $\boldsymbol{\theta}_0 = \mathbf{0}$ is the minimum-ℓ_2 norm. Indeed, this is coherent with the $\alpha \to \infty$ limit of our diagonal linear network.

Interestingly, this analogy also hold in the opposite, $\alpha \to 0^+$ limit. Indeed, this limit is akin to the mean-field limit discussed in Lectures 3 & 4, where the wide network can be seen as an integral over a limiting probability measure over the hidden units. The implicit bias of gradient flow on the square loss corresponds in this case to the minimum Barron norm interpolator, which is akin to a ℓ_1-penalty on the weights.

Remark 34 (Relation to generalisation)**.** It is important to stress that, whether ℓ_1, ℓ_2 or in between, the implicit regularisation above is a property of the architecture and training algorithm, and hence is independent of generalisation. Indeed, we have made no assumption on the data distribution, and whether a particular regularisation is "good" in terms of generalisation will crucially depend on the properties of the target function. For instance, if the underlying predictor is a linear function $y_i = \langle \boldsymbol{\theta}_\star, \boldsymbol{x}_i\rangle + \varepsilon_i$ with sparse weights $||\boldsymbol{\theta}_\star||_0 \ll d$, we expect the minimum-$\ell_1$ norm predictor to generalise better than the minimum-ℓ_2 norm predictor. But the converse can be true if $||\boldsymbol{\theta}_\star||_2^2 = d$ instead.

7.4 To go further

7.4.1 Benefits of noise

As we discussed in section 7.2.3, in the OLS problem the implicit bias is the same whether we use gradient flow, GD or SGD — a consequence of the fact that in these three algorithms problems the gradient lives in the span of the covariates. Interestingly, the situation is very different for linear diagonal neural networks. As shown in (Pesme et al., 2021), adding a stochastic term to gradient flow leads to an algorithm with similar potential, but with an effective initialisation scale $\alpha_{\text{sgf}} < \alpha_{\text{gf}}$. This implies that for a fixed choice of initialisation scale α, these two algorithms will converge to different interpolators with different implicit biases — a situation closer to fig. 7.1. In particular, since $\alpha_{\text{sgf}} < \alpha_{\text{gf}}$, $\boldsymbol{\theta}_\infty^{\text{sgf}}$ will tend to be sparser than $\boldsymbol{\theta}_\infty^{\text{gf}}$. The role of the step-size (which quantifies the difference between GF and GD) has also been studied in (Nacson et al., 2022).

7.4.2 Implicit bias in binary classification

Our discussion in this lecture has focused on results for the square loss function. There is a similar line of work characterising the implicit bias of algorithms for for binary classification with the logistic loss. Soudry et al. (2018) has shown that, when data is linearly separable (hence interpolators exist), GD or SGD for logistic regression converge to the minimum margin interpolator, independently of the initial condition.

Appendix A

Mathematics checklist

A.1 Linear algebra

A.1.1 Important notions

Definition 7 (Column and row space)**.** Let $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ denote a real-valued rectangular matrix with entries $a_{ij} \in \mathbb{R}$. Define the families of vectors $\boldsymbol{a}_i \in \mathbb{R}^d$, $i \in [n]$ and $\boldsymbol{A}_j \in \mathbb{R}^n$, $j \in [d]$ given by the rows and columns of $\boldsymbol{A}$, respectively. We define the *row* and *column* spaces of $\boldsymbol{A}$ as the vector spaces spanned by these families:

$$\begin{aligned} \mathrm{row}(\boldsymbol{A}) &= \mathrm{span}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n) \subset \mathbb{R}^d \\ \mathrm{col}(\boldsymbol{A}) &= \mathrm{span}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_d) \subset \mathbb{R}^n \end{aligned} \tag{A.1.1}$$

Note that seen as a linear transformation $\boldsymbol{A} : \mathbb{R}^d \to \mathbb{R}^n$, the column space is simply its image $\mathrm{col}(\boldsymbol{A}) = \mathrm{Im}(\boldsymbol{A})$.

⚠ For any $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, $\mathrm{col}(\boldsymbol{A}) = \mathrm{row}(\boldsymbol{A}^\top)$.

Definition 8 (Rank)**.** The *rank* of a real-valued rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ is the dimension of its column space.

$$\mathrm{rank}(\boldsymbol{A}) = \dim(\mathrm{col}(\boldsymbol{A})) \tag{A.1.2}$$

In other words, it is the number of linearly independent columns of $\boldsymbol{A}$.

From the definition above, one might wonder why defining the rank as the dimension of the column space and not the row space. Actually, an important result is that these two potential notions are the same.

Theorem 14. For any real-valued rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$, the dimension of the column space is the same as the dimension of the row space:

$$\dim(\mathrm{col}(\boldsymbol{A})) = \dim(\mathrm{row}(\boldsymbol{A})) \tag{A.1.3}$$

Therefore, we have:

$$\mathrm{rank}(\boldsymbol{A}) \leq \min(n, d) \tag{A.1.4}$$

Definition 9 (Full-rank matrix)**.** A real-valued rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ is said to be *full-rank* if:

$$\mathrm{rank}(\boldsymbol{A}) = \min(n, d) \tag{A.1.5}$$

The most important result in linear algebra for the purpose of this course is the singular-value decomposition.

Theorem 15 (Singular value decomposition)**.** Any real-valued rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ can be decomposed as:

$$\boldsymbol{A} = \sum_{i=1}^{\mathrm{rank}(\boldsymbol{A})} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \tag{A.1.6}$$

where $\sigma_i \geq 0$ are non-negative real numbers known as the *singular values* and $\boldsymbol{u}_i \in \mathbb{R}^n$, $\boldsymbol{v}_i \in \mathbb{R}^d$ are known as the left and right *singular vectors*. Moreover, the singular vectors form an orthonormal family with respect to the Euclidean scalar product: $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$, $\boldsymbol{v}_i^\top \boldsymbol{v}_j = \delta_{ij}$.

Remark 35. Without loss of generality we can (and will) assume the singular values $\sigma_i(\boldsymbol{A})$ are non-increasing: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$ where $r = \mathrm{rank}(\boldsymbol{A})$. Often, the SVD is written as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ where $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{d\times r}$ are the orthogonal matrices with columns $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ and $\boldsymbol{D} \in \mathbb{R}^{r\times r}$ is a diagonal matrix of singular values $d_{ij} = \sigma_i \delta_{ij}$.

⚠ Sometimes in the literature you will find $\boldsymbol{A} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}^\top$ with $\tilde{\boldsymbol{U}} \in \mathtt{O}(n)$, $\tilde{\boldsymbol{V}} \in \mathtt{O}(d)$ and $\tilde{D} \in \mathbb{R}^{n\times d}$ obtained by completing $\boldsymbol{U}, \boldsymbol{V}$ with an orthonormal basis of $\mathbb{R}^n$ and $\mathbb{R}^d$, respectively. In this case, $\tilde{\boldsymbol{D}} \in \mathbb{R}^{n\times d}$ is a rectangular matrix with a block given by $\boldsymbol{D}$ and zero elsewhere.

A.1.2 Important classes of matrices

There are a few classes of real valued square matrices which will often appear in the lectures. Here we review the most important ones.

- A square matrix $\boldsymbol{O} \in \mathbb{R}^{n\times n}$ is said to be **orthogonal** if:

  $$\boldsymbol{O}^\top \boldsymbol{O} = \boldsymbol{O}\boldsymbol{O}^\top = \boldsymbol{I}_n \tag{A.1.7}$$

  Note that orthogonal matrices are always invertible $\boldsymbol{O}^\top = \boldsymbol{O}^{-1}$, and as linear transformations they define isometries, i.e. they preserve the Euclidean norm of vectors:

  $$||\boldsymbol{O}\boldsymbol{v}||_2 = ||\boldsymbol{v}||_2 \tag{A.1.8}$$

  for any $\boldsymbol{v} \in \mathbb{R}^n$. The set of orthogonal matrices define a group, known as the orthogonal group $\boldsymbol{O}(n) = \{\boldsymbol{O} \in \mathbb{R}^{n\times n} : \boldsymbol{O}^\top\boldsymbol{O} = \boldsymbol{I}_n\}$. From eq. (A.1.7), it is immediate to show that $\det(\boldsymbol{O}) \in \{-1, +1\}$. Orthogonal matrices such that $\det(\boldsymbol{O}) = +1$ are also known as *rotations*, while orthogonal matrices with $\det(\boldsymbol{O}) = -1$ are known as *reflections*. The set of rotations $\mathtt{SO}(n) = \{\boldsymbol{O} \in \mathbb{R}^{n\times n} : \boldsymbol{O}^\top\boldsymbol{O} = \boldsymbol{I}_n \text{ and } \det \boldsymbol{O} = 1\} \subset \mathtt{O}(n)$ defines a subgroup of $\mathtt{O}(n)$ known as the *special orthogonal group*. Orthogonal matrices have complex eigenvalues $\lambda_i \in \mathbb{C}$ with modulus $|\lambda_i| = 1$.

- A square matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is said to be **symmetric** if:

  $$\boldsymbol{M}^\top = \boldsymbol{M} \tag{A.1.9}$$

  Every real symmetric matrix can be diagonalised over the real numbers:

  $$\boldsymbol{M} = \boldsymbol{O}\boldsymbol{D}\boldsymbol{O}^\top \tag{A.1.10}$$

  where $\boldsymbol{O} \in \mathtt{O}(n)$ is an orthogonal matrix and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with entries $d_{ij} = \lambda_i(\boldsymbol{M})\delta_{ij}$. Note that the rows (or columns) of $\boldsymbol{O}$ are precisely the normalised eigenvectors of $\boldsymbol{M}$. A symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is said to be **positive semi-definite** $\boldsymbol{M} \succeq 0$ if its spectrum is non-negative $\lambda_i(\boldsymbol{M}) \geq 0$ for all $i \in [n]$, and is said to be **positive-definite** $\boldsymbol{M} \succ 0$ if its spectrum is positive $\lambda_i(\boldsymbol{M}) > 0$ for all $i \in [n]$.

⚠ A symmetric matrix can have zero eigenvalues, so it might not be invertible. However, a positive-definite symmetric matrix $\boldsymbol{M} \succ 0$ is always invertible.

Example 9. For any real valued rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$, the square matrices $\boldsymbol{A}^\top \boldsymbol{A} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{A}\boldsymbol{A}^\top \in \mathbb{R}^{n\times n}$ are symmetric positive semi-definite matrices.

- A square matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is said to be a **projection** if it is idempotent:

$$\boldsymbol{P}^2 = \boldsymbol{P} \tag{A.1.11}$$

From this, it follows that a projection matrix can only have eigenvalues 0 or 1: $\lambda_i(\boldsymbol{P}) \in \{0, 1\}$. Therefore, a projection matrix can always be written as:

$$\boldsymbol{P} = \sum_{i=1}^{\text{rank}(\boldsymbol{P})} \boldsymbol{v}_i \boldsymbol{v}_i^\top \tag{A.1.12}$$

As the name suggests, projection matrices $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ geometrically define projections into a linear subspace of $\text{Im}(\boldsymbol{P}) \subset \mathbb{R}^n$ of dimension $\text{rank}(\boldsymbol{P})$. More explicitly, this subspace is precisely the span of the eigenvectors corresponding to the non-zero eigenvalues $V = \text{span}(\boldsymbol{v}_i)$. An **orthogonal projection** $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is a projection which is also orthogonal $\boldsymbol{P} \in \texttt{SO}(n)$, and correspond to the case where the eigenvalues $\boldsymbol{v}_i$ are orthonormal vectors. Finally, every orthogonal projection defines an orthogonal decomposition $\mathbb{R}^n = \text{Im}(\boldsymbol{P}) \oplus \text{Ker}(\boldsymbol{P})$, for which we can associate another orthogonal projection matrix $\boldsymbol{P}_\perp \in \mathbb{R}^{n\times n}$, which is the projection on its orthogonal complement $\text{Ker}(\boldsymbol{P})$.

⚠ With the exception of the identity $\boldsymbol{I}_n$, a projection matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is never invertible.

Example 10. Let $\boldsymbol{v} \in \mathbb{R}^n$ denote a unit-norm vector $||\boldsymbol{v}||_2 = 1$. Then:

$$\boldsymbol{P} = \boldsymbol{v}\boldsymbol{v}^\top, \qquad \boldsymbol{P} = \boldsymbol{I}_n - \boldsymbol{v}\boldsymbol{v}^\top \tag{A.1.13}$$

define a orthogonal projection in the line $L = \{\alpha \boldsymbol{v} \in \mathbb{R}^n : \alpha \in \mathbb{R}\}$ and its orthogonal complement.

A.1.3 Matrix norms

Just as for vectors, there are different useful notions of norm for matrices. Here we discuss the most relevant for the lectures. Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ denote a real-valued rectangular matrix with singular values $(\sigma_j(\boldsymbol{A}))_{j\in[r]}$, where $r := \text{rank}(\boldsymbol{A})$. Without loss of generality, we assume $\sigma_j(\boldsymbol{A})) \geq 0$ are non-decreasing. We define the following matrix norms:

- The **Frobenius norm** of $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is defined as:

$$||\boldsymbol{A}||_{\texttt{F}} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d} A_{ij}^2} = \sqrt{\text{Tr}\,\boldsymbol{A}^\top \boldsymbol{A}} = \sqrt{\sum_{i=1}^{r} \sigma_i(\boldsymbol{A})^2} \tag{A.1.14}$$

- The **operator norm** of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is defined as:

$$||\boldsymbol{A}||_{\texttt{op}} = \sup_{\boldsymbol{v}\in\mathbb{S}^{d-1}} ||\boldsymbol{A}\boldsymbol{v}||_2 = \sigma_1(\boldsymbol{A}) \tag{A.1.15}$$

where we recall $\sigma_1(\boldsymbol{A})$ is the top singular value of $\boldsymbol{A}$.

- The **nuclear norm** of a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ is defined as:

$$||A||_* = \operatorname{Tr}\left(\sqrt{\boldsymbol{A}\boldsymbol{A}^\top}\right) = \sum_{i=1}^{r} \sigma_i(\boldsymbol{A}) \tag{A.1.16}$$

Remark 36. All the norms above are a particular case of a more general class of norms known as Schatten norms:

$$||\boldsymbol{A}||_p = \left(\sum_{i=1}^{r} \sigma_i(\boldsymbol{A})^p\right)^{1/p}. \tag{A.1.17}$$

More precisely, they correspond to the case $p = 1, 2, \infty$.

Lemma 7. For any real valued matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we have:

$$||\boldsymbol{A}||_{\mathtt{op}} \leq ||\boldsymbol{A}||_{\mathtt{F}} \leq ||\boldsymbol{A}||_* \tag{A.1.18}$$

Proof. Since the norms are positive, it is equivalent to show:

$$||\boldsymbol{A}||_{\mathtt{op}}^2 \leq ||\boldsymbol{A}||_{\mathtt{F}}^2 \leq ||\boldsymbol{A}||_*^2 \tag{A.1.19}$$

The first inequality is immediate: since $\sigma_i(\boldsymbol{A}) \geq 0$, the sum can only be larger than any of the terms:

$$||\boldsymbol{A}||_{\mathtt{F}}^2 = \sum_{i=1}^{r} \sigma_i(\boldsymbol{A})^2 \geq \sigma_i(\boldsymbol{A}) \text{ for all } i \in [r]. \tag{A.1.20}$$

The second inequality follow from noting that:

$$\begin{aligned} ||\boldsymbol{A}||_*^2 = \left(\sum_{i=1}^{r} \sigma_i\right)^2 &= \sum_{i,j=1}^{r} \sigma_i \sigma_j = \sum_{i=1}^{r} \sigma_i^2 + \sum_{i \neq j} \sigma_i \sigma_j \\ &= ||\boldsymbol{A}||_{\mathtt{F}}^2 + \sum_{i \neq j} \sigma_i \sigma_j \geq ||\boldsymbol{A}||_{\mathtt{F}}^2 \end{aligned} \tag{A.1.21}$$

since $\sigma_i(\boldsymbol{A}) \geq 0$. □

Lemma 8. All the norms above are equivalent since:

- $||\boldsymbol{A}||_{\mathtt{F}} \leq ||\boldsymbol{A}||_* \leq \sqrt{r}||\boldsymbol{A}||_{\mathtt{F}}$
- $||\boldsymbol{A}||_{\mathtt{op}} \leq ||\boldsymbol{A}||_* \leq r||\boldsymbol{A}||_{\mathtt{op}}$
- $||\boldsymbol{A}||_{\mathtt{op}} \leq ||\boldsymbol{A}||_{\mathtt{F}} \leq \sqrt{r}||\boldsymbol{A}||_{\mathtt{op}}$

A.1.4 Matrix identities

Let $\boldsymbol{U} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{V} \in \mathbb{R}^{d \times n}$ be two rectangular matrices. We have the following useful identities:

- The traces of the resolvent and co-resolvent are related as:

$$\operatorname{Tr}(\boldsymbol{U}\boldsymbol{V} - z\boldsymbol{I}_n)^{-1} = \operatorname{Tr}(\boldsymbol{V}\boldsymbol{U} - z\boldsymbol{I}_d)^{-1} - \frac{n-d}{z} \tag{A.1.22}$$

  Taking the derivative with respect to z on both sides, this also implies:

$$\operatorname{Tr}(\boldsymbol{U}\boldsymbol{V} - z\boldsymbol{I}_n)^{-2} = \operatorname{Tr}(\boldsymbol{V}\boldsymbol{U} - z\boldsymbol{I}_d)^{-2} - \frac{n-d}{z^2} \tag{A.1.23}$$

- Push-through identity:

$$(\boldsymbol{UV} - z\boldsymbol{I}_n)^{-1}\boldsymbol{U} = \boldsymbol{U}(\boldsymbol{VU} - z\boldsymbol{I}_d)^{-1} \tag{A.1.24}$$

- Block inversion formula:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\boldsymbol{A} - \boldsymbol{BD}^{-1}\boldsymbol{C})^{-1} & -(\boldsymbol{A} - \boldsymbol{BD}^{-1}\boldsymbol{C})^{-1}\boldsymbol{BD}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{BD}^{-1}\boldsymbol{C})^{-1} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{BD}^{-1}\boldsymbol{C})^{-1}\boldsymbol{BD}^{-1} \end{bmatrix} \tag{A.1.25}$$

  where $\boldsymbol{A}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{D} \in \mathbb{R}^{m\times m}$.

- Sherman-Morrison lemma: Let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ denote an invertible matrix and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{d\times d}$ two vectors such that $\boldsymbol{v}^\top \boldsymbol{A}^{-1}\boldsymbol{u} \neq -1$. Then:

$$(\boldsymbol{A} + \boldsymbol{uv})^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{uv}^\top\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^\top\boldsymbol{A}^{-1}\boldsymbol{u}} \tag{A.1.26}$$

A.2 Probability

A few good references to catch up:

- Roman Vershynin's book "*High-dimensional probability: an introduction with applications in data science*", freely available online.
- Chapter 1 of Philippe Rigollet and Jan-Christian Hütte lecture notes on "*High-Dimensional Statistics*", freely available online.

A.2.1 Geometry of random variables

Definition 10 (L^p norm of a r.v.)**.** Let X denote a random variable. The L^p norm of X is given by:

$$||X||_{L^p} = \left(\mathbb{E}[|X^p|]\right)^{1/p}, p \in [1, \infty) \tag{A.2.1}$$

This can be extended to $p = \infty$ by defining:

$$||X||_{L^\infty} = \text{ess sup}|X| \tag{A.2.2}$$

It can be shown that indeed this defines a norm, and therefore the linear space:

$$L^p = \{X : ||X||_{L^p} \leq \infty\} \tag{A.2.3}$$

defines a Banach space.

Remark 37. Definition 10 still makes sense for $p \in (0, 1)$. However, in this case $||\cdot||_{L^p}$ is not a norm.

The space L^2 is also a Hilbert space, with inner product defined as:

$$\langle X, Y\rangle_{L^2} = \mathbb{E}[|XY|] \tag{A.2.4}$$

Note that $||X||_{L^p}$ is an increasing function of p. This implies the following inclusion of L^p spaces:

$$L^\infty \subset \cdots \subset L^2 \subset L^1 \tag{A.2.5}$$

This is quite intuitive: a bounded random variable has all moments, a random variable with p moments has all $p-1$ moments, and so on. However, having finite p moments for all p does not imply X is almost surely bounded. The Gaussian distribution is an example.

Lemma 9. Let $G \sim \mathcal{N}(0,1)$, then for all $p \in [1, \infty)$:

$$||G||_{L^p} \leq \sqrt{p} \tag{A.2.6}$$

and $||G||_{L^\infty} = \infty$ since Gaussian variables are unbounded.

L^p spaces are a particular case of a more general geometry of random variables, known as *Orlicz spaces.*

Definition 11 (Orlicz spaces). Let ψ denote a convex increasing function such that:

$$\lim_{x \to 0} \psi(x) = 0, \qquad \lim_{x \to \infty} \psi(x) = \infty. \tag{A.2.7}$$

For any random variable X, we define the *Orlicz norm* of X as:

$$||X||_\psi = \inf\{k > 0 : \mathbb{E}[\psi(^{|X|}/_k)] \leq 1\} \tag{A.2.8}$$

Further, we define the *Orlicz space* associated to ψ as:

$$L_\psi = \{X : ||X||_\psi < \infty\} \tag{A.2.9}$$

Note that for $\psi(x) = x^p$, we retrieve $L_\psi = L^p$. However, Orlicz spaces allow us to defined a more refined geometry of random variables, that allow us to distinguish different classes of random variables that have all moments but are not necessarily bounded. For instance, $\psi_p(x) = e^{x^p} - 1$ defines a family of Orlicz spaces L_{ψ_p} that sit exactly in between L^p and L^∞. For instance, note that $L_{\psi_1} \subset L^p$ since exponentials grow faster than polynomials. However, $L^\infty \subset L_{\psi_1}$ since the expectation of the exponential of a bounded random variable is finite. Therefore, $L^\infty \subset L_{\psi_1} \subset L^p$ for any $p > 1$. More generally, L_{ψ_p} define a hierarchy of Orlicz spaces based on the tails of the distributions, with tails which are lighter as p increases. Two important examples are sub-exponential and sub-Gaussian random variables.

Definition 12 (Sub-Gaussian r.v.). A random variable X is sub-Gaussian if:

$$||X||_{\psi_2} = \inf\left\{C > 0 : \mathbb{E}\left[\exp\left(\frac{X^2}{C^2}\right)\right] \leq 2\right\} \leq \infty \tag{A.2.10}$$

In other words, it is the Orlicz space L_{ψ_2} with $\psi_2(x) = e^{x^2} - 1$. The following are equivalent characterisations:

- **Gaussian tails:** $\exists c_1$ such that for all $t > 0$:

$$\mathbb{P}(|X| \geq t) \leq 2e^{-\frac{t^2}{c_1^2}} \tag{A.2.11}$$

- **Moments:** $\exists c_2$ such that for all $p > 1$:

$$||X||_{L^p} \leq c_2\sqrt{p} \tag{A.2.12}$$

- **Moment generating function:** $\exists c_3$ such that if $\mathbb{E}[X] = 0$:

$$\mathbb{E}[e^{tX}] \leq e^{c_3^2 t^2}, \qquad t \in \mathbb{R} \tag{A.2.13}$$

Example 11. Some popular examples of sub-Gaussian random variables are:

- Gaussian random variables are sub-Gaussian. In particular, if we have $G \sim \mathcal{N}(0, \sigma^2)$, then:

$$||G||_{\psi_2} \leq C\sigma \tag{A.2.14}$$

- Bernouilli random variables $X \sim \mathrm{Ber}({}^1\!/\!{}_2)$ are sub-Gaussian random variables. In particular, we have:

$$||X||_{\psi_2} \leq \frac{1}{\sqrt{\log 2}} \tag{A.2.15}$$

- Bounded random variables are sub-Gaussian random variables. In particular, we have:

$$||X||_{\psi_2} \leq \frac{||X||_{L^\infty}}{\sqrt{\log 2}} \tag{A.2.16}$$

Intuitively, sub-Gaussian variables are variables that have the same tail as Gaussian random variables. We can define sub-exponential random variables similarly.

Definition 13 (Sub-Exponential r.v.)**.** A random variable X is *sub-exponential* if:

$$||X||_{\psi_1} = \inf\left\{C > 0 : \mathbb{E}\left[\exp\left(\frac{X}{C}\right)\right] \leq 2\right\} \leq \infty \tag{A.2.17}$$

In other words, it is the Orlicz space L_{ψ_2} with $\psi_2(x) = e^x - 1$. The following are equivalent characterisations:

- **Exponential tails:** $\exists c_1$ such that for all $t > 0$:

$$\mathbb{P}(|X| \geq t) \leq 2e^{-\frac{t}{c_1^2}} \tag{A.2.18}$$

- **Moments:** $\exists c_2$ such that for all $p > 1$:

$$||X||_{L^p} \leq c_2 p \tag{A.2.19}$$

- **Moment generating function:** $\exists c_3$ such that if $\mathbb{E}[X] = 0$:

$$\mathbb{E}[e^{tX}] \leq e^{c_3^2 t^2}, \qquad |t| \leq {}^1\!/\!{}_{c_3} \tag{A.2.20}$$

Remark 38. Note that from the perspective of the MGF, the only difference between sub-exponential and sub-Gaussian random variables is that the former holds for t bounded. Therefore, we can informally view sub-Gaussian random variables as the class sub-exponential random variables with $c_3 \to 0$.

Proposition 12. A random variable X is sub-Gaussian if and only if X^2 is sub-exponential. Moreover:

$$||X^2||_{\psi_1} = ||X||_{\psi_2}^2 \tag{A.2.21}$$

A.2.2 Classical inequalities

In this section, we review some classical inequalities in probability.

Proposition 13 (Jensen's inequality)**.** Let X denote a real-valued random variable. Then, for any convex function φ:

$$\varphi(\mathbb{E}[X]) \leq \mathbb{E}[\varphi(X)] \tag{A.2.22}$$

⚠ It is a common mistake to inverse the direction of Jensen's inequality.

Proposition 14 (Holder's inequality)**.** For any random variables $X \in L^p$ and $Y \in L^q$ where $p, q \in [1, \infty]$ are conjugate variables ${}^1\!/\!{}_p + {}^1\!/\!{}_q = 1$:

$$|\mathbb{E}[XY]| \leq ||X||_p ||Y||_q \tag{A.2.23}$$

Remark 39. The case $p = q = 2$ is known as the Cauchy-Schwarz inequality:

$$|\mathbb{E}[XY]| \leq ||X||_{L^2} ||Y||_{L^2} \tag{A.2.24}$$

Proposition 15 (Minkowski's inequality)**.** For any random variables $X, Y \in L^p$ and $p \in [1, \infty]$:

$$||X + Y||_p \leq ||X||_p + ||Y||_p \tag{A.2.25}$$

A.2.3 Tail inequalities

Proposition 16 (Markov's inequality)**.** Let X denote a non-negative random variable. Then for all $t > 0$:

$$\mathbb{P}(X \geq t) \leq \frac{\mathbb{E}[X]}{t} \tag{A.2.26}$$

i.e. the probability that X is at least t is at most the expectation divided by t. Note that when $\mathbb{E}[X] > 0$ we can equivalently write:

$$\mathbb{P}(X \geq t\,\mathbb{E}[X]) \leq \frac{1}{t} \tag{A.2.27}$$

Proposition 17 (Chebyshev's inequality)**.** Let X denote a random variable with mean $\mathbb{E}[X] = \mu$ and $\mathrm{Var}(X) = \sigma^2$. Then, for all $t > 0$

$$\mathbb{P}(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2} \tag{A.2.28}$$

Proposition 18 (Chernoff's inequality)**.** Let X denote a real random variable. Then, for all $a \in \mathbb{R}$ and $t > 0$:

$$\mathbb{P}(X \geq a) \leq \mathbb{E}[e^{t(X-a)}] \tag{A.2.29}$$

Note that this holds for all $t > 0$, it is also common to take the infimum over t:

$$\mathbb{P}(X \geq a) \leq \inf_{t \geq 0} \mathbb{E}[e^{t(X-a)}] \tag{A.2.30}$$

A.2.4 Concentration inequalities for the sum of random variables

Proposition 19 (Hoeffding's inequality)**.** Let $X_1, \ldots, X_n$ denote independent, zero mean sub-Gaussian random variables. Then, for every $t > 0$, we have:

$$\mathbb{P}\left(\left|\sum_{i=1}^{n} X_i\right| \geq t\right) \leq 2\exp\left(-\frac{ct^2}{\sum\limits_{i=1}^{n} ||X_i||_{\psi_2}^2}\right) \tag{A.2.31}$$

where $||\cdot||_{\psi_2}$ is the sub-Gaussian norm from definition 12.

Remark 40 (Particular cases of Hoeffding's inequality)**.** The following particular cases of Hoeffding's inequality are useful.

- **Bernouilli:** Let $X_1, \ldots, X_n$ denote independent symmetric Bernouilli random variables, i.e. $X_i \in \{-1, +1\}$ with:

  $$\mathbb{P}(X_i = -1) = \mathbb{P}(X_i = +1) = \frac{1}{2}. \tag{A.2.32}$$

  In this case, applying proposition 19 leads to the following tail bound:

  $$\mathbb{P}\left(\left|\sum_{i=1}^{n} X_i\right| \geq t\right) \leq 2e^{-\frac{t^2}{2n}} \tag{A.2.33}$$

- **Bounded:** Let $X_1, \dots, X_n$ denote independent random variables which are bounded almost surely, i.e. $X_i \in [a_i, b_i]$ a.s. Then, for all $t > 0$ their sum $S_n = \sum_{i=1}^{n} X_n$ satisfy:

$$\mathbb{P}\left(S_n - \mathbb{E}S_n \geq t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right) \tag{A.2.34}$$

Proposition 20 (Bernstein's inequality)**.** Let $X_1, \dots, X_n$ note independent sub-exponential random variables with $\mathbb{E}[X_i] = 0$. Then, there exists a constant c such that for all $t > 0$ the sum $S_n = \sum_{i=1}^{n} X_i$ satisfy:

$$\mathbb{P}(|S_n| \geq t) \leq 2\exp\left(-c\min\left(\frac{t^2}{\sigma^2}, \frac{t}{k}\right)\right) \tag{A.2.35}$$

where $c > 0$ is an absolute constant and:

$$\sigma^2 = \sum_{i=1}^{n} ||X_i||_{\psi_1}, \qquad k = \max_{i\in[n]} ||X_i||_{\psi_1} \tag{A.2.36}$$

where $|| \cdot ||_{\psi_1}$ is the sub-exponential norm from definition 13.

A.2.5 Convergence of random variables

In this appendix, we review the different notions of convergence for random variables. Let $(\Omega, \mathcal{F}, \mathbb{P})$ denote a probability space. We start with one of the strongest forms of convergence:

Definition 14 (Almost sure convergence)**.** We say a sequence of random variables $(X_n)_{n\geq1}$ converges *almost surely* (a.s.) to a random variable X and denote $X_n \overset{a.s.}{\to} X$ if there exists a measurable set $\Omega' \in \mathcal{F}$ such that:

- $\mathbb{P}(\Omega') = 1$.
- For all $\omega \in \Omega'$, $\lim_{n\to\infty} X_n(\omega) = X(\omega)$

Intuitively, almost sure convergence means that $X_n \to X$ just as for deterministic variables, excepts perhaps for exceptional events that have probability zero as $n \to \infty$ (hence the "almost").

Definition 15 (Convergence in probability)**.** We say a sequence of random variables $(X_n)_{n\geq1}$ converges *in probability* to a random variable X and denote $X_n \overset{P}{\to} X$ if for every $\epsilon > 0$:

$$\lim_{n\to\infty} \mathbb{P}(|X_n - X| > \epsilon) = 0 \tag{A.2.37}$$

While almost sure convergence is a statement about the convergence of values taken by a random variable, convergence in probability is a statement about the convergence of probabilities. A particularly intuitive case is when the limiting random variable X is deterministic, i.e. $X = x$ with probability 1. In this case, we can visualise the convergence in probability as the distribution of X_n getting more and more peaked around $X = x$ as $n \to \infty$.

Example 12. Let $X_n \sim \text{Unif}([-\frac{1}{n}, \frac{1}{n}])$ denote a sequence of uniform random variables. We have $X_n \overset{P}{\to} 0$.

Almost sure convergence implies convergence in probability (see Grimmett and Stirzaker (2020) for a proof), but the converse is not true. A standard example is the following:

Example 13. Consider a sequence of binary random variables $X_n \in \{0, 1\}$ such that:

$$\mathbb{P}(X_n = 1) = \frac{1}{n}, \qquad \mathbb{P}(X_n = 0) = 1 - \frac{1}{n} \tag{A.2.38}$$

Then, we have $X_n \xrightarrow{P} 0$ since:

$$\lim \mathbb{P}(X_n = 1) = 0, \qquad \lim \mathbb{P}(X_n = 0) = 1 \tag{A.2.39}$$

However, X_n does not converge almost surely to 0. To see this, consider the event that X_n takes the value 1: $E_n = \{X_n = 1\}$. We have:

$$\sum_{n=1}^{\infty} \mathbb{P}(E_n) = \sum_{n=1}^{\infty} \frac{1}{n} = \infty \tag{A.2.40}$$

By the Borel-Cantelli lemma, a sequence of independent events with probability that sum to ∞ must happen infinitely often.

Definition 16 (Convergence in L^p)**.** We say a sequence of random variables $(X_n)_{n\geq 1}$ converges *in* L^p (or p-th mean) to a random variable X and denote $X_n \xrightarrow{L^p} X$ if:

$$\lim_{n\to\infty} \mathbb{E}[|X_n - X|^p] = 0 \tag{A.2.41}$$

Note that this is equivalent to convergence in $L^p(\mathbb{P})$ norm. Convergence in L^p implies convergence in probability, but the converse if not true. Note that convergence in L^p does not implied and does not imply almost sure convergence: in general these are unrelated.

Example 14. Let $U \sim \text{Unif}([0, 1])$, and define:

$$X_n = \sqrt{n}\, \mathbf{1}_{(0, 1/n)}(U) = \begin{cases} \sqrt{n} & \text{if } U \in (0, 1/n) \\ 0 & \text{otherwise} \end{cases} \tag{A.2.42}$$

Then, X_n convergences in probability to 0 since for all $0 < \epsilon < 1$:

$$\mathbb{P}(|X_n| > \epsilon) = \mathbb{P}\left(\sqrt{n}\mathbf{1}_{(0,1/n)}(U) > \epsilon\right) = \mathbb{P}\left(0 \leq U \leq \frac{1}{n}\right) = \frac{1}{n} \tag{A.2.43}$$

which goes to zero as $n \to \infty$. However, X_n does not converge to zero in L^2 since:

$$\mathbb{E}[X_n^2] = n \int_0^{n/2} \mathrm{d}t = 1 \tag{A.2.44}$$

Note that all the notions so far easily generalise to random vectors or matrices by simply taking an adapted norm. Finally, the last common notion of convergence is convergence in distribution, which we first define for real valued variables:

Definition 17 (Convergence in distribution)**.** We say a sequence of random variables $(X_n)_{n\geq 1}$ converges *in distribution* to a random variable X and denote $X_n \xrightarrow{d} X$ if:

$$\lim_{n\to\infty} \mathbb{P}(X_n \leq t) = \mathbb{P}(X \leq t) \tag{A.2.45}$$

for all t for which the c.d.f. $\mathbb{P}(X \leq t)$ is continuous.

Convergence in distribution is the weakest form of convergence discussed here. Indeed, it is implied by convergence in probability (and hence by both almost sure and L^p convergence). Note that the condition "for all t for which the c.d.f. $\mathbb{P}(X \leq t)$ is continuous" is important, as highlighted by the following example:

Example 15. Consider a sequence of Gaussian random variables with decreasing variance $X_n \sim \mathcal{N}(0, {}^1\!/\!{}_n)$. We have:

$$\lim_{n\to\infty} \mathbb{P}(X_n \leq x) = \lim_{n\to\infty} \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2n}}\right)\right] = \begin{cases} 1 & \text{if } x > 0 \\ \frac{1}{2} & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{A.2.46}$$

Therefore, $X_n \xrightarrow{d} X$ with $\mathbb{P}(X = 0) = 1$ which has c.d.f. $\mathbb{P}(X \leq x) = \Theta(x)$, since the discontinuity point $x = 0$ can be ignored.

Since this definition of convergence in distribution relies on the c.d.f., if does not straightforwardly generalise to random vectors. A more adapted and equivalent notion is known in this context as *weak convergence*:

Definition 18 (Weak convergence)**.** We say a sequence of random vectors $(X_n)_{n\geq 1}$ in $\mathbb{R}^d$ *weakly converges* to a random vector X and denote $X_n \xrightarrow{d} X$ if for any bounded continuous function $f : \mathbb{R}^d \to \mathbb{R}$:

$$\lim_{n\to\infty} \mathbb{E}[f(X_n)] = \mathbb{E}[f(X)] \tag{A.2.47}$$

This definition can be easily extended to any metric space. Note that "bounded continuous" f can also be exchanged for "bounded Lipschitz". Several equivalent characterisations of weak convergence are given by the Portmanteau lemma.

Summary

We can summarise the discussion in this Appendix in Figure A.1. Note that several converse results under stronger assumptions exist. We refer the reader to Chapter 7 of Grimmett and Stirzaker (2020) for a full discussion. Finally, we state the following result which is useful in the context of statistics:

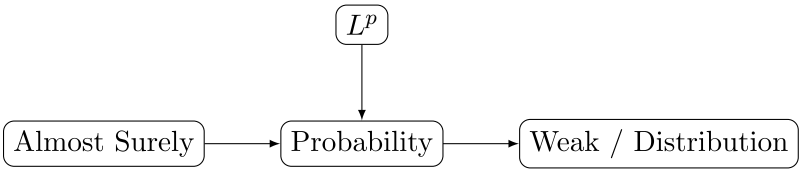

Figure A.1: Different notions of convergence for random variables, with the respective implications.

Lemma 10. Let X_n be an unbiased estimate of $\alpha \in \mathbb{R}$. Then, if $\operatorname{Var}(X_n) \to 0$ as $n \to \infty$, $X_n \xrightarrow{L^2} \alpha$ (and hence also in probability).

Proof. By definition, we have $\mathbb{E}[X_n] = \alpha$. Therefore:

$$\mathbb{E}[|X_n - \alpha|^2] = \mathbb{E}[|X_n - \mathbb{E}[X_n]|^2] = \operatorname{Var}(X_n) \to 0 \text{ as } n \to \infty \tag{A.2.48}$$

which implies convergence in squared-mean. □

A.2.6 Limit theorems

Theorem 16 (Strong law of large numbers)**.** Let $X_1, \dots, X_n$ be a sequence of i.i.d. random variables with mean $\mathbb{E}[X_i] = \mu$, and consider the empirical mean:

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{A.2.49}$$

Then, as $n \to \infty$:

$$\bar{X}_n \xrightarrow{a.s.} \mu \tag{A.2.50}$$

Theorem 17 (Lindeberg-Lévy central limit theorem)**.** Let $X_1, \dots, X_n$ be a sequence of i.i.d. random variables with mean $\mathbb{E}[X_i] = \mu$ and variance $\mathrm{Var}(X_i) = \sigma^2 < \infty$, and consider the empirical mean:

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{A.2.51}$$

Then, as $n \to \infty$:

$$\sqrt{n}(\bar{X}_n - \mu) \xrightarrow{d} \mathcal{N}(0, \sigma^2) \tag{A.2.52}$$

In other words, letting $Z_n = \sqrt{n}/\sigma(\bar{X}_n - \mu)$, for any $t \in \mathbb{R}$:

$$\mathbb{P}(|Z_n| \geq t) \to \mathbb{P}(|G| \geq t) = \int_t^{\infty} \frac{\mathrm{d}x}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{A.2.53}$$

point-wise as $n \to \infty$.

Corollary 1. Let $X_1, \dots, X_n$ be a sequence of i.i.d. random variables with mean $\mathbb{E}[X_i] = \mu$ finite variance. Then:

$$\mathbb{E}[|\bar{X}_n - \mu|] = O(1/\sqrt{d}), \qquad \text{as } n \to \infty \tag{A.2.54}$$

A.3 Analysis

A.3.1 Lipschitz functions

In the course, we will often need to control the regularity of function. A particularly useful notion of regularity is how the slope/derivative of the function changes point-wise. Functions which have a "gentle" change of the slope are more regular than "spiky" functions for which the slope can vary abruptly. This notion is formalised by Lipschitz function.

Definition 19 (Lipschitz function)**.** Let (X, d_X) and (Y, d_Y) denote metric spaces. A function $f : X \to Y$ is called L-Lipschitz if there exists $L \in \mathbb{R}$ such that for all $x, y \in X$:

$$d_Y(f(x), f(y)) \leq L \cdot d_X(x, y) \tag{A.3.1}$$

The constant L is known as the Lipschitz constant of f, and the infimum over all L defines a norm, known as the Lipschitz norm of f:

$$||f||_{\mathrm{Lip}} = \inf \ \{L \in \mathbb{R} : d_Y(f(x), f(y)) \leq L \cdot d_X(x, y) \text{ for all } x, y \in X\} \tag{A.3.2}$$

Lipschitz functions with $||f||_{\mathrm{Lip}} < 1$ are also known as *contractions.*

A particular example of interest for the lectures is the case of functions in a normed vector space, where $X = \mathbb{R}^n$ and $Y = \mathbb{R}$ and eq. (A.3.1) reads:

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq L||\boldsymbol{x} - \boldsymbol{y}|| \tag{A.3.3}$$

Proposition 21 (Properties of Lipschitz functions)**.** Lipschitz functions satisfy the following properties:

1. Every Lipschitz function is uniformly continuous.
2. Every differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz, and:

$$||f||_{\text{Lip}} \leq \sup_{x\in\mathbb{R}^n} ||\nabla f(\boldsymbol{x})||_2 \tag{A.3.4}$$

3. The composition of two Lipschitz maps is Lipschitz, with:

$$||f \circ g||_{\text{Lip}} = ||f||_{\text{Lip}}||g||_{\text{Lip}} \tag{A.3.5}$$

⚠ The converse is not true: there are functions which are not everywhere differentiable but are still Lipschitz, for example $f(x) = |x|$ is a 1-Lipschitz function since $||x| - |y|| \leq |x - y|$ for all $x, y \in \mathbb{R}$.

Example 16. Some useful examples of Lipschitz functions.

- For a fixed vector $\boldsymbol{\theta} \in \mathbb{R}^n$, the inner product:

$$f(\boldsymbol{x}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle \tag{A.3.6}$$

  is a Lipschitz function on $\mathbb{R}^n$ with:

$$||f||_{\text{Lip}} = ||\boldsymbol{\theta}||_2 \tag{A.3.7}$$

- More generally, any matrix $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ acting as a linear operator:

$$\begin{aligned} \boldsymbol{A} : \mathbb{R}^d &\to \mathbb{R}^n \\ \boldsymbol{x} &\mapsto \boldsymbol{A}\boldsymbol{x} \end{aligned}$$

  Is a Lipschitz function with:

$$||\boldsymbol{A}||_{\text{Lip}} = ||\boldsymbol{A}||_{\mathtt{op}} \tag{A.3.8}$$

- Any norm $f(\boldsymbol{x}) = ||\boldsymbol{x}||$ on $\mathbb{R}^n$ is a Lipschitz function. The Lipschitz norm of f is the smallest f that satisfies:

$$||\boldsymbol{x}|| \leq L||\boldsymbol{x}||_2, \text{ for all } \boldsymbol{x} \in \mathbb{R}^n \tag{A.3.9}$$

  For example, the L^1 norm:

$$f(\boldsymbol{x}) = ||\boldsymbol{x}||_1 = \sum_{i=1}^{n} |x_i| \tag{A.3.10}$$

  is a Lipschitz function with Lipschitz constant $L = \sqrt{n}$. More generally, the L^p norms have Lipschitz constant $L = n^{\max(0, 1/2 - 1/p)}$.

- The rectified linear unit $f(x) = \max(0, x)$ is a 1-Lipschitz function.

- The Logistic loss $\ell(x) = \log(1 + e^{-x})$ is a 1-Lipschitz function.
- The Hinge loss $\ell(x) = \max(0, 1 - x)$ is a 1-Lipschitz function.

It is also useful to have in mind examples of functions which are not Lipschitz (and why). The most common features of non-Lipschitz functions are: (a) unbounded derivative/slope; (b) Discontinuities; (c) Infinite oscillations. In some cases, a Lipschitz function can be defined by restricting the domain of non-Lipschitz functions to exclude the singularities. Below, we give a few useful examples:

Example 17. The following functions are not Lipschitz everywhere in their domain.

- The logarithm $f(x) = \log x$ is not a Lipschitz function in $\mathbb{R}_+$ since $f'(x) = \frac{1}{x}$. However, it is a Lipschitz function in any domain $[a, \infty)$ with $a > 0$, with Lipschitz constant $L = {}^1\!/_a$.
- The quadratic function $f(x) = x^2$ is not Lipschitz in $\mathbb{R}$ since its derivative $f'(x) = x$ is unbounded. However, the truncated quadratic $f(x) = \min(1, x^2)$ is a Lipschitz function with constant $L = 2$.
- The square root function $f(x) = \sqrt{x}$ is not a Lipschitz function in $\mathbb{R}_+$ since $f'(x) = {}^1\!/_{2\sqrt{x}}$ grows unbounded as $x \to 0^+$. However, it is a Lipschitz function in any interval $[a, \infty)$ with $a > 0$ with Lipschitz constant $L = {}^1\!/_{2\sqrt{x}}$.
- The exponential function $f(x) = e^x$ is not Lipschitz.
- The Heavyside step function:

$$\Theta(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases} \tag{A.3.11}$$

  is not Lipschitz because of the discontinuity at $x = 0$.
- The function $f(x) = \sin({}^1\!/_x)$ is not Lipschitz on $(0, \infty)$, due to the fast oscillations close to 0^+. One can define a Lipschitz function by restricting it to an interval $[a, \infty)$ with $a > 0$, with Lipschitz constant $L = {}^1\!/_{a^2}$

A.4 Big-O notation

In these notes, we often employ the so-called *Big-O* notation, a handy way of comparing the order of magnitude or limiting behaviour of functions. In this appendix, we give a formal definition and discuss some intuition.

Definition 20 (Big-O notation)**.** Let $f, g : \mathbb{R} \to \mathbb{R}$ denote two real-valued functions. We say:

- "$f(x)$ is big-O of $g(x)$" and write $f(x) = O(g(x))$ as $x \to \infty$ if there exists $M > 0$ and $x_0 \in \mathbb{R}$ such that:

$$|f(x)| < M|g(x)| \text{ for all } x > x_0 \tag{A.4.1}$$

  Intuitively, $f(x) = O(g(x))$ means $f(x)$ is "at most" $g(x)$, meaning that one can make it $f(x)$ as large as g by multiplying by a constant (with respect to x_0). It is used to denote asymptotic upper bounds. If $g(x)$ is non-zero beyond a certain point, this is equivalent to:

$$\limsup_{x \to \infty} \frac{f(x)}{g(x)} < \infty \tag{A.4.2}$$

- "$f(x)$ is little-O of $g(x)$" and write $f(x) = o(g(x))$ as $x \to \infty$ if for every $\varepsilon > 0$, there exists constant $x_0 \in \mathbb{R}$ such that:

  $$|f(x)| < \varepsilon |g(x)| \text{ for all } x > x_0 \tag{A.4.3}$$

  Intuitively, $f(x) = o(g(x))$ means that $g(x)$ grows much faster than $f(x)$, or equivalently that $f(x)$ is of lower order than $g(x)$. If $g(x)$ is non-zero beyond a certain point, this is equivalent to:

  $$\lim_{x \to \infty} \frac{f(x)}{g(x)} = 0 \tag{A.4.4}$$

Note that both notions can be easily generalised for other limits than infinity.

Remark 41. Although it is widespread to write $f(x) = O(g(x))$ and $f(x) = o(g(x))$, the use of the equality is an abuse of notation, since this is not a symmetric statement. For instance, $O(x) = O(x^2)$ but $O(x^2) \neq O(x)$. The equality here should be understood in the same sense as we use in English: "*Aristotle is a man, but a man is not necessarily Aristotle*". A more precise notation would be to saw $f(x) < O(g(x))$ or $f(x) \in O(g(x))$, with $O(g(x))$ thought as a class of functions h satisfying eq. (A.4.6).

Properties 3. The following important properties hold:

- Multiplicative constants are irrelevant: if $f(x) = O(g(x))$, then $100f(x) = O(g(x))$.
- When adding two functions, we only care about the larger one. For example $x^3 + 100x^2 = O(x^3)$.
- For all $a, b > 0$, we have $x^a = O(x^b)$ if and only if $a \leq b$ and $x^a = o(x^b)$ if and only if $a < b$.
- Polynomials are always smaller than exponentials: $x^a = o(e^{x^\epsilon})$ for every $a, \epsilon > 0$, even if ϵ is much smaller than a. For example, $x^{100} = o(e^{\sqrt{x}})$.
- Logarithms are always smaller than polynomials: $(\log x)^a = o(x^\epsilon)$ for all $a, \epsilon > 0$, even if ϵ is much smaller than a. For example, $100x^2 \log x = o(x^3)$.

An useful and related notion is the big-Theta:

Definition 21 (Big-Θ)**.** Let $f, g : \mathbb{R} \to \mathbb{R}$ denote two real-valued functions. We say "f is theta of g" and write $f(x) = \Theta(g(x))$ as $x \to \infty$ if both $f(x) = O(g(x))$ and $g(x) = O(f(x))$ as $x \to \infty$. In order words, there exists constants $m, M > 0$ and $x_0 \in \mathbb{R}$ such that for $x > x_0$:

$$mg(x) < f(x) < Mg(x) \tag{A.4.5}$$

Intuitively, $f(x) = \Theta(g(x))$ means that f is of the same order as g. It is also common to see the notation $f(x) \asymp g(x)$ and to say f and g are asymptotically equivalent.

A complementary notion, often used in the context of computer science is the big-Ω.

Definition 22 (Big-Ω notation)**.** Let $f, g : \mathbb{R} \to \mathbb{R}$ denote two real-valued functions. We say:

- "$f(x)$ is big-Ω of $g(x)$" and write $f(x) = \Omega(g(x))$ as $x \to \infty$ if $g(x) = O(f(x))$ as $x \to \infty$. More precisely, there exists $M > 0$ and $x_0 \in \mathbb{R}$ such that:

  $$|f(x)| > M|g(x)| \text{ for all } x > x_0 \tag{A.4.6}$$

  Intuitively, $f(x) = O(g(x))$ means $f(x)$ is "at least" $g(x)$. It is used to denote asymptotic lower bounds. If $g(x)$ is non-zero beyond a certain point, this is equivalent to:

  $$\liminf_{x \to \infty} \frac{f(x)}{g(x)} > 0 \tag{A.4.7}$$

- "$f(x)$ is little-ω of $g(x)$" and write $f(x) = \omega(g(x))$ as $x \to \infty$ if $g(x) = o(f(x))$ as $x \to \infty$. More precisely, for every $\varepsilon > 0$, there exists constant $x_0 \in \mathbb{R}$ such that:

  $$|f(x)| > \varepsilon |g(x)| \text{ for all } x > x_0 \tag{A.4.8}$$

  Intuitively, $f(x) = \omega(g(x))$ means that $f(x)$ dominates $g(x)$ asymptotically. If $g(x)$ is non-zero beyond a certain point, this is equivalent to:

  $$\lim_{x \to \infty} \frac{f(x)}{g(x)} = \infty \tag{A.4.9}$$

Example 18. Some examples with Big-Ω:

- $4x^2 - 3x + 2 = \Omega(x^2)$
- $x^5 = \omega(x^4)$
- $e^x = \omega(x^a)$ for any $a > 0$.

Bibliography

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

Francis Bach. High-dimensional analysis of double descent for linear regression with random projections. *SIAM Journal on Mathematics of Data Science*, 6(1):26–50, 2024.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

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(32):15849–15854, 2019.

Avrim Blum and Ronald Rivest. Training a 3-node neural network is np-complete. *Advances in neural information processing systems*, 1, 1988.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pages 1024–1034. PMLR, 2020.

S Boucheron, G Lugosi, and P Massart. Concentration inequalities: A nonasymptotic theory of independence. univ. press, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.

Chen Cheng and Andrea Montanari. Dimension free ridge regression. *The Annals of Statistics*, 52(6): 2879–2912, 2024.

Lenaic 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.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Romain Couillet and Zhenyu Liao. *Random matrix methods for machine learning*. Cambridge University Press, 2022.

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*, 34:10131–10143, 2021.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Leonardo Defilippis, Bruno Loureiro, and Theodor Misiakiewicz. Dimension-free deterministic equivalents and scaling laws for random feature regression. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 104630–104693. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/bd18189308a4c45c7d71ca83acf3deaa-Paper-Conference.pdf.

David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eK3i09YQ.

László Erdős, Benjamin Schlein, and Horng-Tzer Yau. Local semicircle law and complete delocalization for wigner random matrices. *Communications in Mathematical Physics*, 287(2):641–655, 2009.

László Erdős, Antti Knowles, Horng-Tzer Yau, and Jun Yin. Spectral statistics of erdős-rényi graphs ii: Eigenvalue spacing and the extreme eigenvalues. *Communications in Mathematical Physics*, 314 (3):587–640, 2012.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Federica Gerace, Bruno Loureiro, Florent Krzakala, Marc Mézard, and Lenka Zdeborová. Generalisation error in learning with random features and the hidden manifold model. In *International Conference on Machine Learning*, pages 3452–3462. PMLR, 2020.

Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová. Modeling the influence of data structure on learning in neural networks: The hidden manifold model. *Physical Review X*, 10 (4):041044, 2020.

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. In *Mathematical and Scientific Machine Learning*, pages 426–471. PMLR, 2022.

Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford university press, 2020.

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.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *Annals of statistics*, 50(2):949, 2022.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Hong Hu and Yue M. Lu. Universality laws for high-dimensional learning with random features. *IEEE Transactions on Information Theory, in press*, 2022.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

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.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Chaoyue Liu, Libin Zhu, and Misha Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33:15954–15964, 2020a.

Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. Bad global minima exist and sgd can reach them. *Advances in Neural Information Processing Systems*, 33:8543–8552, 2020b.

Marco Loog, Tom Viering, Alexander Mey, Jesse H Krijthe, and David MJ Tax. A brief prehistory of double descent. *Proceedings of the National Academy of Sciences*, 117(20):10625–10626, 2020.

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*, 75 (4):667–766, 2022.

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(33):E7665–E7671, 2018.

Theodor Misiakiewicz and Andrea Montanari. Six lectures on linearized neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2024(10):104006, 2024.

Theodor Misiakiewicz and Basil Saeed. A non-asymptotic theory of kernel ridge regression: deterministic equivalents, test error, and gcv estimator. *arXiv preprint arXiv:2403.08938*, 2024.

Andrea Montanari and Basil N Saeed. Universality of empirical risk minimization. In *Conference on Learning Theory*, pages 4310–4312. PMLR, 2022.

Mor Shpigel Nacson, Kavya Ravichandran, Nathan Srebro, and Daniel Soudry. Implicit bias of the step size in linear diagonal neural networks. In *International Conference on Machine Learning*, pages 16270–16295. PMLR, 2022.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Radford M Neal. Priors for infinite networks. *Bayesian learning for neural networks*, pages 29–53, 1996.

Manfred Opper, W Kinzel, J Kleinz, and R Nehl. On the ability of the optimal perceptron to generalise. *Journal of Physics A: Mathematical and General*, 23(11):L581, 1990.

LA Pastur and VA Martchenko. The distribution of eigenvalues in certain sets of random matrices. *Math. USSR-Sbornik*, 1(4):457–483, 1967.

Scott Pesme. *Deep Learning Theory Through the Lens of Diagonal Linear Networks*. PhD thesis, Lausanne, 2024. URL https://infoscience.epfl.ch/handle/20.500.14299/208225.

Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. *Advances in Neural Information Processing Systems*, 34:29218–29230, 2021.

Loucas Pillaud-Vivien and Scott Pesme. Rethinking sgd's noise - ii: Implicit bias, Sep 2022. URL https://francisbach.com/implicit-bias-sgd/.

Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

Grant Rotskoff and Eric Vanden-Eijnden. Trainability and accuracy of artificial neural networks: An interacting particle system approach. *Communications on Pure and Applied Mathematics*, 75(9): 1889–1935, 2022.

Jack W Silverstein. Strong convergence of the empirical distribution of eigenvalues of large dimensional random matrices. *Journal of Multivariate Analysis*, 55(2):331–339, 1995.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19(70):1–57, 2018.

S Spigler, M Geiger, S d'Ascoli, L Sagun, G Biroli, and M Wyart. A jamming transition from under- to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, oct 2019. doi: 10.1088/1751-8121/ab4c8b. URL https://dx.doi.org/10.1088/1751-8121/ab4c8b.

Terence Tao and Van Vu. Random matrices: Universality of local eigenvalue statistics. *Acta Mathematica*, 206(1):127–204, 2011. doi: 10.1007/s11511-011-0061-3. URL https://doi.org/10.1007/s11511-011-0061-3.

Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

Matus Telgarsky. Benefits of depth in neural networks. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL https://proceedings.mlr.press/v49/telgarsky16.html.

Craig A Tracy and Harold Widom. *The distribution of the largest eigenvalue in the Gaussian ensembles: β= 1, 2, 4*. Springer, 2000.

Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, 2008. ISBN 9780387790527. URL https://books.google.fr/books?id=mwB8rUBsbqoC.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, nov 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Eugene P. Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *Annals of Mathematics*, 62(3):548–564, 1955. ISSN 0003486X, 19398980. URL http://www.jstor.org/stable/1970079.

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. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020.

Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural computation*, 17(9):2077–2098, 2005.