Chapter 12

Over-parameterized models

Chapter summary

- A model is said to be over-parameterized when it has sufficiently many parameters to fit the training data perfectly. While many overparameterized models can significantly overfit the data, the ones learned by (stochastic) gradient descent typically do not.
- Implicit regularization of gradient descent: for linear models, when there are several minimizers (typically for over-parameterized models), gradient descent techniques tend to converge to the one with minimum Euclidean norm.
- Double descent: for unregularized models learned with gradient descent techniques, when the number of parameters grows and when gradient descent is used to fit the model, the performance can exhibit a second descent after the test error blows up when the number of parameters goes beyond the number of observations.
- Global convergence of gradient descent for two-layer neural networks: in the infinite width (and thus strongly over-parameterized) limit, gradient descent exhibits some globally convergent behavior for a non-convex problem, which can be analyzed for simple architectures.

In this chapter, we will cover three recent topics within learning theory, all related to high-dimensional models (such as neural networks) in the "over-parameterized" regime, where the number of parameters is larger than the number of observations. When regularization is added to the estimation procedures, we have seen in Chapters 7, 8, and 9, that estimation can be numerically and statistically efficient by adding penalties to the empirical risk. In this section, we consider primarily non-penalized problems, with a

regularization coming from the choice of optimization algorithm (here, gradient descent).

The number of parameters is not what generally characterizes the generalization capabilities of regularized learning methods. See Section 3.6 and Section 9.2.3.

12.1 Implicit bias of gradient descent

Given an optimization problem whose aim is to minimize some function $F(\theta)$ over with respect to a parameter $\theta \in \mathbb{R}^d$, if there is a unique global minimizer θ_*, then the goal of optimization algorithms is to find this minimizer, that is, we want that the t-th iterate θ_t converges to θ_*. When there are multiple minimizers (thus for a function which cannot be strongly convex), we showed only that $F(\theta_t) - \inf_{\theta \in \mathbb{R}^d} F(\theta)$ is converging to zero for convex functions F (and only if a minimizer exists, see Chapter 5).

With some extra assumptions, it can be shown that the algorithm converges to one of the multiple minimizers of F (Bolte et al., 2010) (note that when F is convex, this set is also convex). But which one? This is referred to as the implicit regularization properties of optimization algorithms, and here, gradient descent and its variants.

This is interesting in machine learning because, when $F(\theta)$ is the empirical loss on n observations, d is much larger than n, and no regularization is used, there are multiple minimizers. An arbitrary empirical risk minimizer is not expected to work well on unseen data, and a classical solution is to use explicit regularization (e.g., ℓ_2-norms like in Chapter 3 and Chapter 7, or ℓ_1-norms like in Chapters 8 and 9). In this section, we show that optimization algorithms have a similar regularizing effect. In a nutshell, gradient descent usually leads to minimum ℓ_2-norm solutions, in a similar way that boosting algorithms were related to ℓ_1-norm regularization in Section 10.3. This shows that the chosen empirical risk minimizer is not arbitrary.

This will be explicitly shown for the quadratic loss and partially only for the logistic loss. These results will be used in subsequent sections.

12.1.1 Least-squares

We consider the least-squares objective function[1] $F(\theta) = \frac{1}{2n}\|y - X\theta\|_2^2$ from Chapter 3, with $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times d}$ such that $d > n$ and (for simplicity) $XX^\top \in \mathbb{R}^{n \times n}$ invertible (this is the kernel matrix). There are thus infinitely many (a whole affine subspace) solutions such that $y = X\theta$, since the column space of X is the entire space $\mathbb{R}^n$ and θ has dimension $d > n$. We apply gradient descent with step-size $\gamma \leqslant \frac{1}{L} = \lambda_{\max}(\frac{1}{n}X^\top X)^{-1}$, which is equal to $\lambda_{\max}(\frac{1}{n}XX^\top)^{-1}$, starting from $\theta_0 = 0$, leading to $\theta_t = \theta_{t-1} - \frac{\gamma}{n}X^\top(X\theta_{t-1} - y)$, and

[1] We use X as a notation for the design matrix to highlight that in this section we will consider predictions that are also linear in x.

thus we have

$$X\theta_t - y = X\theta_{t-1} - y - \frac{\gamma}{n} XX^\top (X\theta_{t-1} - y) = \big(I - \frac{\gamma}{n} XX^\top\big)(X\theta_{t-1} - y),$$

leading to, by recursion,

$$X\theta_t - y = \big(I - \frac{\gamma}{n} XX^\top\big)^t (X\theta_0 - y) = \big(I - \frac{\gamma}{n} XX^\top\big)^t (-y). \tag{12.1}$$

We thus get $\|X\theta_t - y\|_2^2 \leqslant \big(1 - \frac{\gamma}{n}\lambda_{\max}(XX^\top)\big)^{2t} \|y\|_2^2$ and hence linear convergence of $X\theta_t$ towards y, with a convergence rate depending on the condition number of the kernel matrix $XX^\top$.

Moreover, when started at $\theta_0 = 0$, gradient descent techniques (stochastic or not) will always have iterates θ_t that are linear combinations of rows of X, that is, of the form $\theta_t = X^\top \alpha_t$ for some $\alpha_t \in \mathbb{R}^n$. This is an alternative algorithmic version of the representer theorem from Chapter 7.

Since $X\theta_t$ converges to y, $X\theta_t = XX^\top \alpha_t$ converges to y. Since $K = XX^\top$ is invertible, this means that α_t is converges to $K^{-1}y$, and thus $\theta_t = X^\top \alpha_t$ converges to $X^\top K^{-1} y$. It turns out that this is exactly the minimum ℓ_2-norm solution as by standard Lagrangian duality (Boyd and Vandenberghe, 2004):

$$\begin{aligned}
\inf_{\theta \in \mathbb{R}^d} \frac{1}{2}\|\theta\|_2^2 \text{ such that } y = X\theta &= \inf_{\theta \in \mathbb{R}^d} \sup_{\alpha \in \mathbb{R}^n} \frac{1}{2}\|\theta\|_2^2 + \alpha^\top (y - X\theta) \\
&= \sup_{\alpha \in \mathbb{R}^n} \alpha^\top y - \frac{1}{2}\|X^\top \alpha\|_2^2 \text{ with } \theta = X^\top \alpha \text{ at optimum}, \\
&= \sup_{\alpha \in \mathbb{R}^n} \alpha^\top y - \frac{1}{2}\alpha^\top K \alpha.
\end{aligned}$$

The last problem is exactly solved for $\alpha = K^{-1}y$, with at optimum $\theta = X^\top \alpha$. Note that in Chapter 7, we used this formula for function interpolation to compare different RKHSs (see Prop. 7.2).

Lojasiewicz's inequality (♦). It turns out that the linear convergence obtained from Eq. (12.1) can be obtained directly for any L-smooth function, for which we have the so-called Lojasiewicz's inequality:

$$\forall \theta \in \mathbb{R}^d, \ F(\theta) - F(\theta_*) \leqslant \frac{1}{2\mu}\|F'(\theta)\|_2^2, \tag{12.2}$$

for some $\mu > 0$.

In Chapter 5, we have seen that this is a consequence of μ-strong-convexity (Lemma 5.1), but this can be satisfied without strong convexity. For example, for the least-squares example, we have, for any minimizer θ_*:

$$\begin{aligned}
\|F'(\theta)\|_2^2 &= \Big\|\frac{1}{n} X^\top X(\theta - \theta_*)\Big\|_2^2 = \frac{1}{n^2}(\theta - \theta_*)^\top X^\top XX^\top X(\theta - \theta_*) \\
&\geqslant \frac{\lambda_{\min}^+(XX^\top)}{n^2}(\theta - \theta_*)^\top X^\top X(\theta - \theta_*),
\end{aligned}$$

where $\lambda_{\min}^+(XX^\top) = \lambda_{\min}^+(X^\top X)$ is the smallest *non-zero* eigenvalue of $XX^\top$ (which is also the one of $X^\top X$). Thus, we have

$$\|F'(\theta)\|_2^2 \geqslant \frac{\lambda_{\min}^+(K)}{n^2}\|X(\theta-\theta_*)\|_2^2 = \frac{2\lambda_{\min}^+(K)}{n}[F(\theta)-F(\theta_*)].$$

Thus, Eq. (12.2) is satisfied with $\mu = \frac{1}{n}\lambda_{\min}^+(K)$. Note that this includes also the strongly-convex case since $\lambda_{\min}^+(X^\top X) \geqslant \lambda_{\min}(X^\top X)$.

When Eq. (12.2) is satisfied, we have for the t-th iterate of gradient descent with step-size $\gamma = 1/L$, following the analysis of Chapter 5 (Prop. 5.3):

$$F(\theta_t) - F(\theta_*) \leqslant F(\theta_{t-1}) - F(\theta_*) - \frac{1}{2L}\|F'(\theta_{t-1})\|_2^2 \leqslant \big(1-\frac{\mu}{L}\big)\big[F(\theta_{t-1})-F(\theta_*)\big].$$

Moreover, we can then show that the iterates x_t are also converging to a minimizer of F (see Bolte et al., 2010; Karimi et al., 2016, for more details).

12.1.2 Separable classification

We now consider logistic regression, that is, for $y_i \in \{-1,1\}$, $i = 1,\dots,n$,

$$F(\theta) = \frac{1}{n}\sum_{i=1}^n \log(1+\exp(-y_i x_i^\top \theta)), \tag{12.3}$$

with $X \in \mathbb{R}^{n\times d}$ the design matrix (with rows equal to the input vectors $x_1,\dots,x_n$) such that $d > n$ and the kernel matrix $XX^\top \in \mathbb{R}^{n\times n}$ is invertible. In the regression setting, interpolation corresponds to $X\theta = y$; in the classification setting, we predict perfectly if and only if $\operatorname{sign}(X\theta) = y$, which happens when $y \circ (X\theta)$ (where $\circ$ is the component-wise product) has strictly positive components. Such an interpolator always exists (for example, the one for regression on y).

Maximum margin classifier. Since $XX^\top$ is invertible, there exists $\eta \in \mathbb{R}^d$ of unit norm such that $\forall i \in \{1,\dots,n\}$, $y_i x_i^\top \eta > 0$ (e.g., as mentioned earlier, $y = X^\top(XX^\top)^{-1}y$). We denote by η_* the one such that

$$\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$$

is maximal (and thus strictly positive). We denote by $\frac{1}{\rho} > 0$ its value. This η_* solves the following problem, which can be rewritten as, using Lagrange duality:

$$\begin{aligned}\frac{1}{\rho} = \sup_{\|\eta\|_2\leqslant 1}\min_{i\in\{1,\dots,n\}} y_i x_i^\top\eta &= \sup_{\|\eta\|_2\leqslant 1, t\in\mathbb{R}} t \text{ such that } \forall i\in\{1,\dots,n\},\ y_i x_i^\top\eta \geqslant t \\ &= \inf_{\alpha\in\mathbb{R}_+^n}\sup_{\|\eta\|_2\leqslant 1, t\in\mathbb{R}} t + \sum_{i=1}^n \alpha_i(y_i x_i^\top\eta - t) \\ &= \inf_{\alpha\in\mathbb{R}_+^n}\Big\|\sum_{i=1}^n \alpha_i y_i x_i\Big\|_2 \text{ such that } \sum_{i=1}^n \alpha_i = 1,\end{aligned} \tag{12.4}$$

with $\eta \propto \sum_{i=1}^n \alpha_i y_i x_i$ at optimum. Moreover, by complementary slackness, a non-negative α_i is non zero only for i attaining the minimum in $\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$.

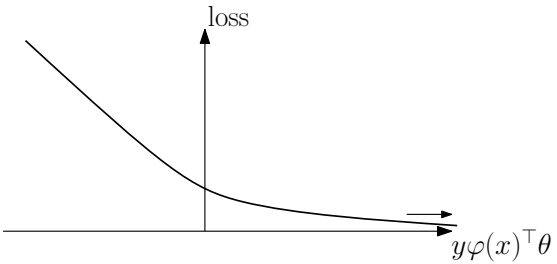
Reformulation as an SVM. Because of positive homogeneity, we aim to have the quantity $\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$ large and $\|\eta\|_2$ small, and we can decide to constrain the first and minimize the second one. In other words, we can see η_* as the unit-norm direction of the solution θ_* of the following optimization problem (with now non-negative Lagrange multipliers $\alpha_1, \dots, \alpha_n$):

$$\begin{aligned}\inf_{\theta\in\mathbb{R}^d} \frac{1}{2}\|\theta\|_2^2 \text{ such that } \operatorname{Diag}(y)X\theta \geqslant 1_n &= \inf_{\theta\in\mathbb{R}^d} \sup_{\alpha\in\mathbb{R}^n_+} \frac{1}{2}\|\theta\|_2^2 + \alpha^\top(1_n - \operatorname{Diag}(y)X\theta) \\ &= \sup_{\alpha\in\mathbb{R}^n_+} \alpha^\top 1_n - \frac{1}{2}\|X^\top \operatorname{Diag}(y)\alpha\|_2^2 \\ &\qquad \text{with } \theta = X^\top \operatorname{Diag}(y)\alpha \text{ at optimum.}\end{aligned}$$

Note that above, $\operatorname{Diag}(y)X\theta \geqslant 1_n$ is the compact formulation of the inequality constraints $\forall i \in \{1,\dots,n\}$, $y_i x_i^\top \theta \geqslant 1$. Given η, θ is equal to $\eta / \min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta$, so that the optimal value of the problem above is $\frac{1}{2}\rho^2$.

The vector θ_* above is the solution of the separable SVM from Section 4.1.2 with vanishing regularization parameter, that is, of $\frac{1}{2}\|\theta\|_2^2 + C\sum_{i=1}^n (1 - y_i x_i^\top \theta)_+$ for C large enough. See Section 4.1.2 for an illustration.

Divergence and convergence of directions. Because the logistic loss plotted below is strictly positive and tends to zero at infinity, the function F in Eq. (12.3) has an infimum equal to zero, which is not attained. However, for any sequence θ_t such that all $y_i x_i^\top \theta_t$, $i = 1, \dots, n$, tend to $+\infty$, we have $F(\theta_t) \to \inf_{\theta\in\mathbb{R}^d} F(\theta) = 0$.



In such a situation, gradient descent cannot converge to a point, and it has to diverge to achieve small values of F. It turns out that it diverges along a direction, that is, $\|\theta_t\|_2 \to +\infty$, with $\frac{1}{\|\theta_t\|_2}\theta_t \to \eta$ for some $\eta \in \mathbb{R}^d$ of unit ℓ_2-norm. That direction η has to lead to perfect classification (that is $y_i x_i^\top \eta > 0$ for all $i \in \{1,\dots,n\}$). Among all of them, the direction η is exactly the maximum margin one, that is, which maximizes $\min_{i\in\{1,\dots,n\}} y_i x_i^\top \eta > 0$. See Gunasekar et al. (2018) for a detailed proof. Here, we just give a simple argument on a slightly modified problem.

Gradient flow on the exponential loss (♦). We consider instead the logarithm of the empirical risk associated with the exponential loss $G(\theta) = \log\left[\frac{1}{n}\sum_{i=1}^n \exp(-y_i x_i^\top \theta)\right]$, which is asymptotically equivalent to the logistic loss for $y_i x_i^\top \theta$ tending to infinity for all $i \in \{1, \dots, n\}$ (which is the case when gradient descent diverges). Moreover, we replace the gradient descent recursion $\theta_t = \theta_{t-1} - \gamma G'(\theta_{t-1})$ by the gradient flow

$$\xi'(\tau) = -G'(\xi(\tau)). \tag{12.5}$$

This ordinary differential equation approximates gradient descent for vanishing step-sizes, as $\xi(\gamma t) \approx \theta_t$ for γ tending to zero. The use of gradient flows instead of gradient descent is a standard theoretical simplification that allows the use of differential calculus (see, e.g., Scieur et al., 2017, and references therein).

We have, for all $\theta \in \mathbb{R}^d$:

$$G'(\theta) = \frac{-\sum_{i=1}^n y_i x_i \exp(-y_i x_i^\top \theta)}{\sum_{i=1}^n \exp(-y_i x_i^\top \theta)} = -\sum_{i=1}^n \alpha_i y_i x_i,$$

for $\alpha_i = \frac{\exp(-y_i x_i^\top \theta)}{\sum_{j=1}^n \exp(-y_j x_j^\top \theta)}$, for $i \in \{1, \dots, n\}$, leading to α in the simplex (with non-negative components and summing to one). Thus, from Eq. (12.4) defining the maximum margin hyperplane, we get:

$$\|G'(\theta)\| \geqslant \frac{1}{\rho}. \tag{12.6}$$

Moreover, comparing maxima and soft-maxima,[2], we get:

$$-\log(n) - \min_{i \in \{1,\dots,n\}} y_i x_i^\top \theta \leqslant G(\theta) \leqslant - \min_{i \in \{1,\dots,n\}} y_i x_i^\top \theta.$$

We thus have a flow $\tau \mapsto \xi(\tau)$ that cannot converge as by Eq. (12.5) and Eq. (12.6), $\|\xi'(\tau)\|_2 \geqslant 1/\rho$. Moreover, it maximizes a function that is a constant away from the margin. Therefore, it has to diverge along a direction that maximizes this margin. We now make this reasoning precise.

By integration by part, using $\frac{d}{d\tau} G(\xi(\tau)) = G'(\xi(\tau))^\top \xi'(\tau) = -\|G'(\xi(\tau))\|_2^2$ and Eq. (12.6) twice:

$$\begin{aligned}
\min_{i \in \{1,\dots,n\}} y_i x_i^\top \xi(\tau) &\geqslant -G(\xi(\tau)) - \log(n) \\
&= -G(\xi(0)) + \int_0^\tau \|G'(\xi(u))\|_2^2 du - \log(n) \\
&\geqslant -G(\xi(0)) + \frac{1}{\rho}\int_0^\tau \|G'(\xi(u))\|_2 du - \log(n) \qquad (12.7) \\
&\geqslant -G(\xi(0)) + \frac{1}{\rho^2}\tau - \log(n) \qquad (12.8)
\end{aligned}$$

[2]We use $0 \geqslant \log\left(\frac{1}{n}\sum_{i=1}^n e^{z_i}\right) - \max_{i \in \{1,\dots,n\}} z_i = \log\left(\frac{1}{n}\sum_{i=1}^n e^{z_i - \max_{j \in \{1,\dots,n\}} z_j}\right) \geqslant \log(1/n)$.

(note that Eq. (12.7) is not needed to derive Eq. (12.8), but will be used later). Thus, from Eq. (12.8), for $\tau \geqslant \rho^2\big[\log(n) + G(\xi(0))\big]$, we have a non-negative lower bound. Moreover the derivative of $\tau \mapsto \|\xi(\tau)\|_2$ is $\tau \mapsto -G'(\xi(\tau))^\top\big(\frac{\xi(\tau)}{\|\xi(\tau)\|_2}\big)$, and its magnitude is less than $\|G'(\xi(\tau))\|_2$. This implies by integration that

$$\|\xi(\tau)\|_2 \leqslant \|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du.$$

We thus get from Eq. (12.7) and Eq. (12.6):

$$\begin{aligned}
\min_{i\in\{1,\dots,n\}} y_i x_i^\top\big(\tfrac{\xi(\tau)}{\|\xi(\tau)\|_2}\big) &\geqslant \frac{-G(\xi(0)) + \frac{1}{\rho}\int_0^\tau \|G'(\xi(u))\|_2 du - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du} \\
&= \frac{-G(\xi(0)) + \frac{1}{\rho}\big(\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du\big) - \frac{1}{\rho}\|\xi(0)\|_2 - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du} \\
&= \frac{1}{\rho} + \frac{-G(\xi(0)) - \frac{1}{\rho}\|\xi(0)\|_2 - \log(n)}{\|\xi(0)\|_2 + \int_0^\tau \|G'(\xi(u))\|_2 du} \\
&\geqslant \frac{1}{\rho} - \frac{G(\xi(0)) + \frac{1}{\rho}\|\xi(0)\|_2 + \log(n)}{\|\xi(0)\|_2 + \tau/\rho}, \text{ since } \int_0^\tau \|G'(\xi(u))\|_2 du \geqslant \frac{\tau}{\rho}.
\end{aligned}$$

The lower bound above tends to $\frac{1}{\rho}$ when τ tends to infinity, which is the maximal value. We thus get convergence to the maximum margin hyperplane.

Alternative proof (♦). We provide another informal derivation based on gradients. The gradient $F'(\theta)$ of the original objective function based on the logistic loss is equal to $F'(\theta) = -\frac{1}{n}\sum_{i=1}^n \frac{\exp(-y_i x_i^\top \theta)}{1+\exp(-y_i x_i^\top \theta)} y_i x_i$.

Asymptotically, θ_t will behave as $\|\theta_t\|\eta$, with $\|\theta_t\|_2$ tending to infinity. Thus, because we have a sum of exponentials with arguments that go to infinity, the dominant term in $F'(\theta_t)$ corresponds to the indices i for which $-y_i x_i^\top \eta$ is the largest. Moreover, all of these values must be negative (indeed, we can only attain zero loss for well-classified training data). We denote by I this set. Thus, asymptotically,

$$F'(\theta_t) \sim -\frac{1}{n}\sum_{i\in I} y_i \exp(-\|\theta_t\|_2 y_i x_i^\top \eta) x_i.$$

Moreover, if we admit for simplicity that $F'(\theta_t)$ diverges in the direction $-\eta$, thus η has to be proportional to a vector $\sum_{i\in I}\alpha_i y_i x_i$, where $\alpha \geqslant 0$, and $\alpha_i = 0$ as soon as i is not among the minimizers of $y_i x_i^\top \eta$. This is exactly the optimality condition for η_* in Eq. (12.4). Thus $\eta = \eta_*$.

Summary. Overall, we obtain a classifier corresponding to a minimum ℓ_2-norm separating hyperplane. See examples in two dimensions in Figure 12.1. Note that gradient descent on the logistic regression problem may not be the most efficient way to obtain a maximum margin hyperplane. See convergence rates by Soudry et al. (2018); Ji and Telgarsky (2018) and a simpler subgradient algorithm below.

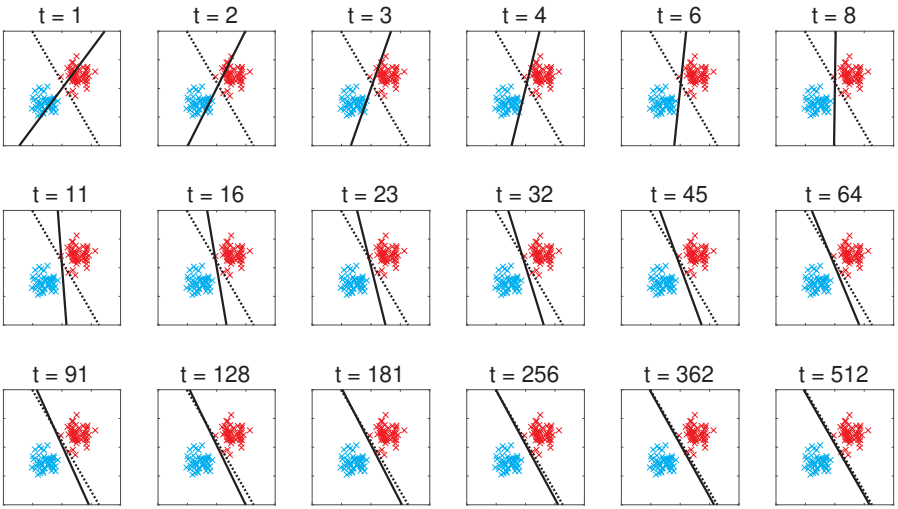


Figure 12.1: Logistic regression on separable data estimated with gradient descent on the unregularized empirical risk, at various numbers of iterations t. This is implemented by minimizing the logistic loss function with data $\binom{x_i}{1} \in \mathbb{R}^3$. The dotted line represents the maximum margin hyperplane, while the plain line represents the current classification hyperplane.

Subgradient method for the hinge loss and perceptron (♦). For linearly separable data, dedicated algorithms exist that are explicitly or implicitly based on the hinge loss. If we consider the "margin" $\rho > 0$ defined as

$$\rho^2 = \inf_{\theta \in \mathbb{R}^d} \|\theta\|_2^2 \text{ such that } \operatorname{Diag}(y) X \theta \geqslant 1_n. \tag{12.9}$$

To obtain a linear separator, one can use the subgradient method from Section 5.3 applied to the cost function

$$F(\theta) = \max_{i \in \{1,\dots,n\}} (1 - y_i x_i^\top \theta)_+.$$

The iteration is

$$\theta_t = \theta_{t-1} + \gamma 1_{y_{i_t} x_{i_t}^\top \theta_{t-1} < 1} y_{i_t} x_{i_t}, \tag{12.10}$$

where $i_t \in \arg\min_{i \in \{1,\dots,n\}} y_i x_i^\top \theta_{t-1}$, and γ is the step-size. With θ_* being the minimizer in Eq. (12.9), we have $F(\theta_*) = 0 = \min_{\theta \in \mathbb{R}^d} F(\theta)$, and after t steps, following the analysis of Prop. 5.6, we get:

$$\min_{u \leqslant t} F(\theta_u) \leqslant \frac{\gamma R^2}{2} + \frac{\rho^2}{2\gamma t}.$$

The quantity above is less than ε as soon as $\frac{\gamma R^2}{2} + \frac{\rho^2}{2\gamma t} \leqslant \varepsilon$, which can be achieved by $\gamma = \frac{\varepsilon}{R^2}$ and $t = \frac{\rho^2}{\gamma \varepsilon} = \frac{\rho^2 R^2}{\varepsilon^2}$, thus an objective function less than $\frac{\rho R}{\sqrt{t}}$. If $F(\theta_t) < 1$, then, following Section 4.1), we have linearly separated the data, which happens as soon

as we have $t > (\rho R)^2$. The iteration in Eq. (12.10) is a variation on the perceptron algorithm (Rosenblatt, 1958; Novikoff, 1962), presented in the exercise below.

Exercise 12.1 *Extend the analysis above to the stochastic gradient algorithm for the objective function $F(\theta) = \frac{1}{n}\sum_{i=1}^n (1 - y_i x_i^\top \theta)_+$. What can be concluded when the data are i.i.d. and a single pass over the data is made?*

Exercise 12.2 (perceptron) *In the same set-up as above, we consider the iteration, started at θ_0, that at time t looks for an $i_t \in \{1, \dots, n\}$ such that $y_{i_t} x_{i_t}^\top \theta_{t-1} \leqslant 0$ and, if found, implements the update $\theta_t = \theta_{t-1} + y_{i_t} x_{i_t}$. Show that all points are well-classified if the number of iterations is greater than $(\rho R)^2$.*

12.1.3 Beyond convex problems (♦)

The implicit bias of gradient descent can be observed and analyzed in various models, beyond linear ones. In this section, we focus on diagonal linear networks, where the analysis of gradient flows is reasonably simple. We highlight the potential difference in implicit biases depending on the chosen learning algorithm.

We consider our traditional least-squares model with design matrix $X \in \mathbb{R}^{n \times d}$ and response vector $y \in \mathbb{R}^n$, with the least-squares objective function: $F(\theta) = \frac{1}{2n}\|y - X\theta\|_2^2$, where $d > n$, and with an invertible kernel matrix $XX^\top \in \mathbb{R}^{n \times n}$, leading to infinitely many minimizers. We consider different learning dynamics, which we study in continuous time for simplicity.

From gradient flow to mirror flow. The gradient flow dynamics on θ is the ordinary differential equation (ODE):

$$\frac{d}{dt}\theta(t) = -F'(\theta(t)) = -\frac{1}{n} X^\top (X\theta(t) - y).$$

As shown in Section 12.1.1, $\theta(t)$ converges exponentially fast to the minimum ℓ_2-norm interpolator. This can extended to the continuous-time limit of mirror descent presented in Section 11.1.3. If we consider a μ-strongly convex mirror map $\Phi : \mathbb{R}^d \to \mathbb{R}$, the mirror descent recursion is defined as $\Phi'(\tilde{\theta}_{k+1}) = \Phi'(\tilde{\theta}_k) - \gamma F'(\tilde{\theta}_k)$, and we obtain the continuous-time limit by setting $\theta(\gamma k) = \tilde{\theta}_k$, leading to the ODE:

$$\frac{d}{dt}\Phi'(\theta(t)) = -F'(\theta(t)), \tag{12.11}$$

which is equivalent to $\Phi''(\theta(t))\frac{d}{dt}\theta(t) = -F'(\theta(t)) = -\frac{1}{n} X^\top (X\theta(t) - y)$. This leads to

$$\frac{d}{dt}\big[X\theta(t) - y\big] = -\frac{1}{n} X \Phi''(\theta(t))^{-1} X^\top \big[X\theta(t) - y\big],$$

which in turn leads to

$$\begin{aligned}\frac{d}{dt}\big[\|X\theta(t)-y\|_2^2\big] &= -\frac{2}{n}\big[X\theta(t)-y\big]^\top X\Phi''(\theta(t))^{-1}X^\top\big[X\theta(t)-y\big]\\ &\leqslant -\frac{2\lambda_{\min}(XX^\top)}{n\mu}\|X\theta(t)-y\|_2^2.\end{aligned}$$

Thus, like for the gradient flow dynamics, $X\theta(t)$ converges exponentially fast to y. Since from Eq. (12.11), $\Phi'(\theta)$ is of the form $\Phi'(\theta_0)+X^\top\alpha$ for some $\alpha\in\mathbb{R}^n$, the corresponding limit α_∞ (with the corresponding θ_∞ such that $\Phi'(\theta_\infty)=\Phi'(\theta_0)+X^\top\alpha_\infty$) of $\alpha(t)$ when t tends to infinity is such $X\theta_\infty=y$ and $\Phi'(\theta_\infty)=\Phi'(\theta_0)+X^\top\alpha_\infty$, which is exactly the interpolator of the data with minimum value of the Bregman divergence $D_\Phi(\theta,\theta_0)=\Phi(\theta)-\Phi(\theta_0)-\Phi'(\theta_0)^\top(\theta-\theta_0)$.[3]

Diagonal linear networks. Following Woodworth et al. (2020), we consider "diagonal linear networks", which are simple hidden layer neural networks defining a prediction function of the form

$$f(x)=(u\circ v)^\top x=\sum_{j=1}^d u_jv_jx_j,$$

for $u,v\in\mathbb{R}^d$, and $u\circ v$ denoting the pointwise product. This is thus an alternative (non-linear) way of defining a linear model defined by $\theta=u\circ v$. We study the gradient flow dynamics for the objective function $G(u,v)=F(u\circ v)$, that is,

$$\begin{aligned}\frac{d}{dt}u(t) &= -\frac{\partial G}{\partial u}(u(t),v(t))=-F'(u(t)\circ v(t))\circ v(t)\\ \frac{d}{dt}v(t) &= -\frac{\partial G}{\partial v}(u(t),v(t))=-F'(u(t)\circ v(t))\circ u(t).\end{aligned}$$

We thus have $\frac{d}{dt}[u\circ u(t)-v\circ v(t)]=0$, and therefore $u\circ u-v\circ v$ is a constant function. If we initialize $v=0$ and u the constant vector equal to $\alpha\in\mathbb{R}$, we have $u\circ u(t)-v\circ v(t)=\alpha^2 1_d$ for all $t\geqslant 0$. Thus, for $\theta(t)=u\circ v(t)$, we have:

$$\frac{d}{dt}\theta(t)=u(t)\circ\frac{d}{dt}v(t)+v(t)\circ\frac{d}{dt}u(t)=-F'(\theta(t))\circ(u\circ u(t)+v\circ v(t)).$$

Moreover, we have $\theta\circ\theta(t)=(u\circ u)\circ(v\circ v)(t)=\frac{1}{4}[u\circ u(t)+v\circ v(t)]^2-\frac{1}{4}[u\circ u(t)-v\circ v(t)]^2$. Thus, we obtain the following ODE for each component θ_j of θ:

$$\frac{d}{dt}\theta_j(t)=-F'(\theta(t))_j\sqrt{4\theta_j(t)^2+\alpha^4}.$$

It can be exactly cast as a mirror flow with mirror map $\Phi(\theta)=\sum_{j=1}^d q(\theta_j)$ for q a convex function such that $q''(\eta)=(4\eta^2+\alpha^4)^{-1/2}$. By integrating twice, one obtains,

[3] By duality, we have $\inf_{X\theta=y}\Phi(\theta)-\Phi'(\theta_0)^\top\theta=\sup_{\alpha\in\mathbb{R}^n}\inf_{\theta\in\mathbb{R}^d}\Phi(\theta)-\Phi'(\theta_0)^\top\theta+\alpha^\top(y-X\theta)$, which is equal to $\sup_{\alpha\in\mathbb{R}^n}\alpha^\top y-\Phi^*(\Psi'(\theta_0)+X^\top\alpha)$, with optimality conditions $\Phi'(\theta)=\Phi'(\theta_0)+X^\top\alpha$ and $X\theta=y$.

with imposing $q'(0) = 0$:

$$q'(\eta) = \frac{1}{2} \arg\sinh(2\eta/\alpha^2) = -\log\alpha + \frac{1}{2}\log\big[2\eta + \sqrt{\alpha^4 + 4\eta^2}\big],$$

and then, imposing $q(0) = 0$, we get an even function of η defined as:

$$\begin{aligned} q(\eta) &= \frac{\eta}{2}\arg\sinh(2\eta/\alpha^2) + \frac{\alpha^2}{4}\big[1 - \sqrt{4\eta^2/\alpha^4 + 1}\big] \\ &= \frac{\eta}{2}\log\big[2\eta + \sqrt{\alpha^4 + 4\eta^2}\big] - \frac{1}{4}\sqrt{4\eta^2 + \alpha^4} + \frac{\alpha^2}{4} - \frac{\eta}{2}\log\alpha^2. \end{aligned}$$

When α tends to $+\infty$, we get $q(\eta) \sim \frac{\eta^2}{\alpha^2}$, and thus the implicit bias converges to the Euclidean norm, and we recover the traditional geometry of gradient descent directly on θ.

However, when α tends to zero, we get $q(\eta) \sim |\eta| \log \frac{1}{\alpha}$, and thus the implicit bias corresponds to the ℓ_1-norm, showing how non-convex optimization can lead to a different implicit bias. See more details by Woodworth et al. (2020), and an analysis of the extra regularizing effect of stochastic gradient descent by Pesme et al. (2021).[4] Note that the analysis above for diagonal networks explicitly shows quantitative global convergence of the gradient flow for a non-convex objective; we consider in Section 12.3 qualitative results which apply more generally.

Beyond linear networks. Characterizing the implicit bias of gradient descent can be done in more complex situations. For example, Chizat and Bach (2020) show that with a neural network with ReLu activations and infinitely many neurons estimated by gradient descent on the empirical logistic loss, then in the infinite width limit, we get a predictor that interpolates the data, with a minimum specific norm, for norms which are exactly the ones obtained in Section 9.3.[5]

12.2 Double descent

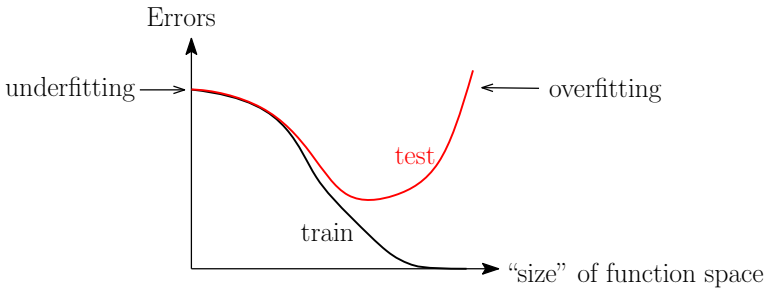
In this section, we consider a recent and interesting phenomenon described in several recent works (Belkin et al., 2019; Mei and Montanari, 2022; Geiger et al., 2020; Hastie et al., 2019), which shows a particular behavior for overparameterized models learned by gradient descent.

12.2.1 The double descent phenomenon

As seen in Chapter 2 and Chapter 4, typical learning curves look like the one below.

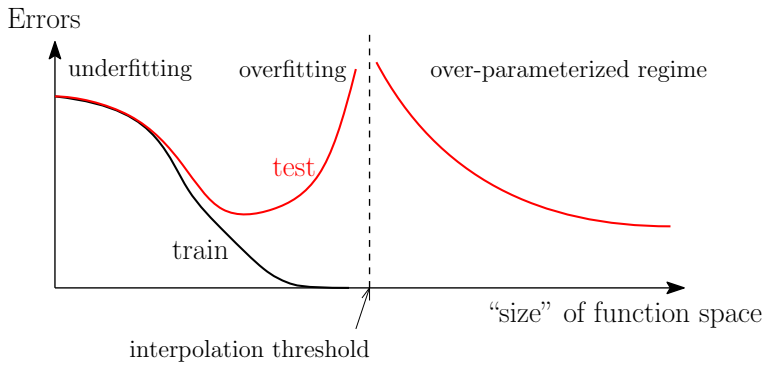
[4] See https://francisbach.com/implicit-bias-sgd/.

[5] See https://www.di.ens.fr/~fbach/ltfp/wide_implicit_bias.html for more details.



Typically, the "capacity" of the space of functions $\mathcal{H}$ used to estimate the prediction function is controlled either by the number of parameters or by some norms of its parameters. In particular, when there is zero training error at the extreme right of the curve, the testing error may be arbitrarily bad. The bound that we have used in Chapter 4, such as Rademacher averages for $\mathcal{H}$ controlled by the ℓ_2-norm of some parameters (with a bound D), grows as $D/\sqrt{n}$, which can typically be quite large. These bounds were true for *all* empirical risk minimizers. In this section, we will focus on a particular one, namely the one obtained by unconstrained gradient descent.

When the model is over-parameterized (in other words, the capacity gets very large), that is, when the number of parameters is large, or the norm constraint allows for exact fitting, a new phenomenon occurs, where after the test error explodes as the capacity grows, it goes down again, as illustrated below.



This section aims to understand why, starting from empirical evidence.

There may be no double descent phenomenon if other empirical risk minimizers are used instead of the one obtained by (stochastic) gradient descent.

12.2.2 Empirical evidence

Toy example with random features. We consider a random feature models like in Chapter 7 and Chapter 9, with features $(v^\top x)_+$, for neurons $v \in \mathbb{R}^d$ sampled uniformly

on the unit sphere. We consider $n = 200$, $d = 5$ with input data distributed uniformly on the unit sphere, and we consider outputs $y = \big(\frac{1}{4} + (v_*^\top x)^2\big)^{-1} + \mathcal{N}(0, \sigma^2)$, with $\sigma = 2$, for some random v_*.

We sample m random features $v_1, \dots, v_m \in \mathbb{R}^d$ uniformly at random on the sphere, and we learn parameters $\theta \in \mathbb{R}^m$ by minimizing

$$\frac{1}{n} \sum_{i=1}^{n} \Big(y_i - \sum_{j=1}^{m} \theta_j (v_j^\top x_i)_+ \Big)^2 + \lambda \|\theta\|_2^2. \tag{12.12}$$

We report in Figure 12.2 train and test errors after learning with gradient descent until convergence: (left) varying m with $\lambda = 0$, (right) varying λ with $m = +\infty$ (we can perform estimation for $m = +\infty$ efficiently because we can compute the corresponding positive-definite kernel $k(x, x') = \mathbb{E}_v[(v^\top x)_+ (v^\top x')_+]$, see Section 9.5).

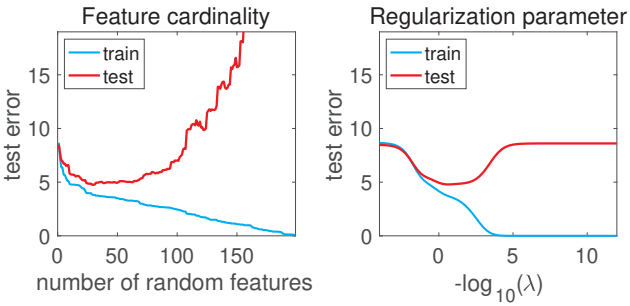


Figure 12.2: Classical learning curve: (left) train and test errors as functions of the number of random features, always less than the number of observations, (right) train and test errors for ridge regression with the same features (i.e., using ℓ_2-regularization).

In the left plot of Figure 12.2, the number of random features m is less than n as the test error diverges. But, when this number m is allowed to grow past n, we see the double descent phenomenon in Figure 12.3. Similar experiments are shown by Belkin et al. (2019); Geiger et al. (2020); Mei and Montanari (2022), in particular for neural networks.

No phenomenon when using regularization. When an extra regularizer is used, that is $\lambda \neq 0$ in Eq. (12.12), then the double descent phenomenon is reduced. In particular, if the regularization parameter λ is adapted for each m, then the phenomenon totally disappears (see Mei and Montanari, 2022, for more details).

12.2.3 Linear regression with Gaussian projections (♦)

To provide some theoretical justification for the double descent phenomenon, we consider a linear regression model in the random design setting, with Gaussian inputs and Gaussian noise.

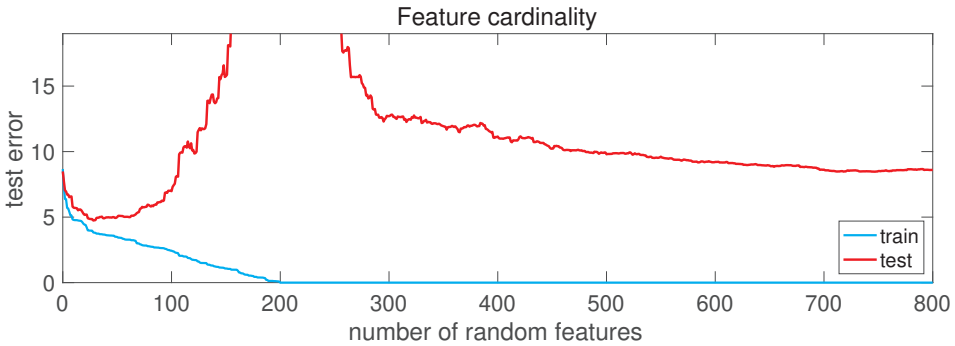


Figure 12.3: Double descent curve: train and test errors as functions of the number of random features. For $m \leqslant n = 200$, this is exactly the curves from Figure 12.2 (left).

That is, we consider a Gaussian random variable with mean 0 and covariance matrix Σ the identity matrix, with n observations $x_1, \dots, x_n$, and responses $y_i = x_i^\top \theta_* + \varepsilon_i$, with ε_i normal with mean zero and variance $\sigma^2 I$.

To have a unique prediction problem with a varying number of features, we consider additional random projections, that is, like in Section 10.2.2,[6] we consider a random matrix $S \in \mathbb{R}^{d \times m}$ with independent components all sampled from a standard Gaussian distribution (mean 0 and variance 1). The main differences are that (a) we will perform an analysis in the random design setting, and (b) we will also need to tackle the overparameterized regime $m > n$.

We will compute the expectation of the risk of the minimum norm empirical risk minimizer (as detailed in Section 12.1.1), which is the one gradient descent converges to. See Bach (2023a) for further more precise asymptotic results using random matrix theory.

We denote by $X \in \mathbb{R}^{n \times d}$ the design matrix, and $\hat{\Sigma} = \frac{1}{n} X^\top X$ the non-centered covariance matrix, and by $K = XX^\top \in \mathbb{R}^{n \times n}$ the kernel matrix. We will need to compute expectations with respect to the data X, ε *and* the random projection matrix S. The estimator $\hat{\theta}$ is equal to $S\hat{\eta}$ with $\hat{\eta} \in \mathbb{R}^m$ a minimizer of $\|y - XS\eta\|_2^2$.

The excess risk is denoted $\mathcal{R}(\hat{\theta}) = (\hat{\theta} - \theta_*)\Sigma(\hat{\theta} - \theta_*)$, and we now consider the two regimes $m > n$ (underparameterized) and $m > n$ (overparameterized). In both cases, as already seen in Chapter 3, the expectation of the excess risk will be composed of two terms: a (squared) "bias term" $\mathcal{R}^{(\text{bias})}(\hat{\theta})$ corresponding to $\sigma = 0$, and a "variance term" $\mathcal{R}^{(\text{var})}(\hat{\theta})$ corresponding to $\theta_* = 0$.

Underparameterized regime. In the under-parameterized regime where $n > d$, the minimum norm empirical risk minimizer is simply the ordinary least-squares estimator. We denote by $\eta_* = (S^\top \Sigma S)^{-1} S^\top \Sigma \theta_* \in \mathbb{R}^m$ the minimizer of $(\theta_* - S\eta)^\top \Sigma (\theta_* - S\eta)$. We have $S\eta_* = \Pi_S \theta_*$ with $\Pi_S = S(S^\top \Sigma S)^{-1} S^\top \Sigma \in \mathbb{R}^{d \times d}$, which is a projection matrix such that $\Pi_S S = S$, $\Pi_S^2 = \Pi_S$, and $\Pi_S^\top \Sigma \Pi_S = \Sigma \Pi_S$.

[6] For an analysis without random projections, see Hastie et al. (2019).

If $m \leqslant n$, the estimator is obtained from the normal equations $S^\top X^\top X S \hat{\eta} = S^\top X^\top y$, and can be expanded using θ_* as follows:

$$\begin{aligned}\hat{\theta} &= S\hat{\eta} = S(S^\top X^\top X S)^{-1} S^\top X^\top y \\ &= S(S^\top X^\top X S)^{-1} S^\top X^\top X \theta_* + S(S^\top X^\top X S)^{-1} S^\top X^\top \varepsilon \quad \text{using } y = X\theta_* + \varepsilon, \\ &= N\theta_* + S(S^\top X^\top X S)^{-1} S^\top X^\top \varepsilon,\end{aligned}$$

with $N = S(S^\top X^\top X S)^{-1} S^\top X^\top X$. Conditioned on S and X, the expected excess risk is equal to:

$$\begin{aligned}\mathbb{E}_\varepsilon\big[\mathcal{R}(\hat{\theta})\big] &= \sigma^2 \operatorname{tr}\big(XS(S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top\big) + \big\|\Sigma^{1/2}\big(N\theta_* - \theta_*\big)\big\|_2^2 \\ &= \sigma^2 \operatorname{tr}\big(S^\top \Sigma S (S^\top X^\top X S)^{-1}\big) + \operatorname{tr}\Big(\big(N\theta_* - \theta_*\big)^\top \Sigma \big(N\theta_* - \theta_*\big)\Big).\end{aligned}$$

For the variance term (first term above), for S fixed, since X has a Gaussian distribution, the matrix $S^\top X^\top X S$ is distributed as a Wishart distribution with parameter $S^\top \Sigma S$ and n degrees of freedom (see, e.g., Haff, 1979, for computations of moments of the Wishart distribution). Thus, if $n > m + 1$, we have:

$$\mathbb{E}_X\big[(S^\top X^\top X S)^{-1}\big] = \frac{1}{n-m-1}(S^\top \Sigma S)^{-1},$$

which in turn implies $\mathbb{E}_{S,X,\varepsilon}\big[\mathcal{R}^{(\mathrm{var})}(\hat{\theta})\big] = \mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{var})}(\hat{\theta})\big] = \dfrac{\sigma^2 m}{n-m-1}$, independently of the choice of the sketching matrix S.

For the bias term, the computation is more involved. We expand

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{bias})}(\hat{\theta})\big] &= \mathbb{E}_X\Big[\operatorname{tr}\Big(\big(N\theta_* - \theta_*\big)^\top \Sigma \big(N\theta_* - \theta_*\big)\Big)\Big] \\ &= \theta_*^\top \Sigma \theta_* + 2\theta_*^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \theta_* \\ &\qquad + \theta_*^\top X^\top X S (S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \theta_*.\end{aligned}$$

To compute the expectation, we will first condition on XS and use the Gaussian conditioning formulas from Section 1.1.3, which leads to, for any matrices A and B of appropriate sizes (proof left as an exercise):

$$\begin{aligned}\mathbb{E}\big[X|XS\big] &= XS(S^\top \Sigma S)^{-1} S^\top \Sigma = X\Pi_S \\ \mathbb{E}\big[\operatorname{tr}(AX^\top BX)|XS\big] &= \operatorname{tr}\big(A\Pi_S^\top X^\top B X \Pi_S\big) + \operatorname{tr}(B)\operatorname{tr}(A\Sigma(I - \Pi_S)).\end{aligned}$$

This leads to, with S assumed fixed, using the identities above:

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\mathrm{bias})}(\hat{\theta})\big] &= \theta_*^\top \Sigma \theta_* + \mathbb{E}_X\big[2\theta_*^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \Pi_S \theta_*\big] \\ &+ \mathbb{E}_X\big[\theta_*^\top \Pi_S^\top X^\top X S (S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top X \Pi_S \theta_*\big] \\ &+ \mathbb{E}_X\Big[\operatorname{tr}\big(XS(S^\top X^\top X S)^{-1} S^\top \Sigma S (S^\top X^\top X S)^{-1} S^\top X^\top\big) \cdot \operatorname{tr}\big(\theta_* \theta_*^\top \Sigma (I - \Pi_S)\big)\Big].\end{aligned}$$

Now, using properties of Π_S, we get:

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma\theta_* + \mathbb{E}_X\big[2\theta_*^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\theta_*\big]\\ &\quad+\mathbb{E}_X\big[\theta_*^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\theta_*\big]\\ &\quad+\mathbb{E}_X\big[\operatorname{tr}\big(S^\top\Sigma S(S^\top X^\top XS)^{-1}\big)\cdot\operatorname{tr}\big(\theta_*\theta_*^\top\Sigma(I-S(S^\top\Sigma S)^{-1}S^\top\Sigma)\big)\big].\end{aligned}$$

We can now group some terms and use $\Pi_S^\top\Sigma\Pi_S = \Sigma\Pi_S$ to get:

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\Big[1+\mathbb{E}_X\big[\operatorname{tr}\big(S^\top\Sigma S(S^\top X^\top XS)^{-1}\big)\big]\Big]\\ &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\Big(1+\operatorname{tr}\Big(\frac{1}{n-m-1}(S^\top\Sigma S)^{-1}S^\top\Sigma S\Big)\Big),\end{aligned}$$

using expectations of Wishart random variables. Overall, we get:

$$\begin{aligned}\mathbb{E}_{X,\varepsilon}\big[\mathcal{R}^{(\text{bias})}(\hat\theta)\big] &= \theta_*^\top\Sigma(I-\Pi_S)\theta_*\cdot\frac{n-1}{n-m-1}\\ &= \theta_*^\top\big(\Sigma-\Sigma S(S^\top\Sigma S)^{-1}S^\top\Sigma\big)\theta_*\cdot\frac{n-1}{n-m-1}.\end{aligned}$$

We can further bound the bias term; we have, for S Gaussian, following the same reasoning as in Section 10.2.2:

$$\begin{aligned}&\mathbb{E}_S\big[\theta_*^\top(I-\Pi_S)^\top\Sigma(I-\Pi_S)\theta_*\big]\\ =\;&\mathbb{E}_S\big[\min_{\eta\in\mathbb{R}^m}(\theta_*-S\eta)^\top\Sigma(\theta_*-S\eta)\big]\text{ by definition of }\Pi_S,\\ \leqslant\;&\mathbb{E}_S\big[\min_{\xi\in\mathbb{R}^d}(\theta_*-SS^\top\xi)^\top\Sigma(\theta_*-SS^\top\xi)\big]\text{ with }\eta\text{ constrained in the column space of }S^\top,\\ \leqslant\;&\min_{\xi\in\mathbb{R}^d}\mathbb{E}_S\big[(\theta_*-SS^\top\xi)^\top\Sigma(\theta_*-SS^\top\xi)\big]\text{ by swapping minimum and expectation},\\ =\;&\min_{\xi\in\mathbb{R}^d}\Big(\theta_*^\top\Sigma\theta_*-2\xi^\top\mathbb{E}\big[SS^\top\big]\Sigma\theta_*+\xi^\top\mathbb{E}\big[SS^\top\Sigma SS^\top\big]\xi\Big)\text{ by developing},\\ =\;&\theta_*^\top\Big(\Sigma-\Sigma\mathbb{E}\big[SS^\top\big]\big(\mathbb{E}\big[SS^\top\Sigma SS^\top\big]\big)^{-1}\mathbb{E}\big[SS^\top\big]\Sigma\Big)\theta_*\text{ by minimizing in closed form},\\ =\;&\theta_*^\top\Big(\Sigma-m\Sigma\big((m+1)\Sigma+\operatorname{tr}(\Sigma)I\big)^{-1}\Sigma\Big)\theta_*\text{ using Wishart expectations},\\ =\;&\theta_*^\top(\Sigma+\operatorname{tr}(\Sigma)I)\big((m+1)\Sigma+\operatorname{tr}(\Sigma)I\big)^{-1}\Sigma\theta_*\\ \leqslant\;&\frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\theta_*^\top\Big(\Sigma+\frac{\operatorname{tr}(\Sigma)}{m+1}I\Big)^{-1}\Sigma\theta_*\leqslant\frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\|\theta_*\|_2^2,\text{ using }\Sigma\preccurlyeq\operatorname{tr}[\Sigma]\cdot I.\end{aligned}$$

Overall, for the underparameterized regime, we obtain an upper-bound equal to $\frac{1}{1-m/n}$ times $\frac{\sigma^2 m}{n}+\frac{2\operatorname{tr}(\Sigma)}{m+1}\cdot\|\theta_*\|_2^2$, which a similar excess risk bound as for ridge regression from Section 3.6 and Section 7.6.4, with $\operatorname{tr}(\Sigma)/m$ playing the role of the regularization parameter, but with the extra term $1/(1-m/n)$ due to the random design setting and the lack of regularization. This leads to a classical bias-variance trade-off with a U-shaped curve. See Bach (2023a) for sharper results.

Overparameterized regime. In the overparameterized regime, when $m \geqslant n$, then the kernel matrix is almost surely invertible, and the minimum ℓ_2-norm interpolator $\hat{\theta}$ is equal to (using the formulas from Section 12.1.1)

$$\hat{\theta} = S\hat{\eta} = SS^\top X^\top (XSS^\top X^\top)^{-1}(X\theta_* + \varepsilon).$$

We can decompose the expectation with respect to ε of the excess risk $\mathcal{R}(\hat{\theta})$ as follows:

$$\begin{aligned}\mathbb{E}_\varepsilon\big[\mathcal{R}(\hat{\theta})\big] &= \sigma^2 \operatorname{tr}\big((XSS^\top X^\top)^{-1} XSS^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1}\big) \\ &\quad + \big\|\Sigma^{1/2}\big(SS^\top X^\top (XSS^\top X^\top)^{-1} X - I\big)\theta_*\big\|_2^2.\end{aligned}$$

We can now use the same reasoning as in the under-parameterized regime, but now taking expectation with respect to S (with X fixed). We have for any symmetric matrices A and B of compatible sizes (proof left as an exercise):

$$\begin{aligned}\mathbb{E}\big[\operatorname{tr}(ASBS^\top)|S^\top X^\top\big] &= \operatorname{tr}\big(X^\top (XX^\top)^{-1} XAX^\top (XX^\top)^{-1} XSBS^\top\big) \\ &\quad + \operatorname{tr}(B)\operatorname{tr}\big[A(I - X^\top (XX^\top)^{-1}X)\big] \\ \mathbb{E}\big[S|S^\top X^\top\big] &= X^\top (XX^\top)^{-1} XS.\end{aligned}$$

Therefore, for the variance term proportional to σ^2, for which we take $A = \Sigma$ and $B = S^\top X^\top (XSS^\top X^\top)^{-2} XS$, we obtain two parts from the identities above. The second part of the variance term becomes:

$$\begin{aligned}&\operatorname{tr}\big[S^\top X^\top (XSS^\top X^\top)^{-2} XS\big] \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1}X)\big] \\ = \;& \operatorname{tr}\big[(XSS^\top X^\top)^{-1}\big] \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1}X)\big].\end{aligned}$$

The first part of the variance term is:

$$\begin{aligned}&\operatorname{tr}\big(X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} XSS^\top X^\top (XSS^\top X^\top)^{-2} XSS^\top\big) \\ = \;& \operatorname{tr}\big((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\big).\end{aligned}$$

Thus, using the expectation of the inverse Wishart distribution, the variance term is

$$\begin{aligned}\mathbb{E}_{\varepsilon,S}[\mathcal{R}^{(\mathrm{var})}(\hat{\theta})] &= \sigma^2 \operatorname{tr}\big((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\big) \\ &\quad + \sigma^2 \frac{\operatorname{tr}\big((XX^\top)^{-1}\big)}{m - n - 1} \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1}X)\big].\end{aligned}$$

For the bias term, we have:

$$\begin{aligned}&\big\|\Sigma^{1/2}\big(SS^\top X^\top (XSS^\top X^\top)^{-1} X - I\big)\theta_*\big\|_2^2 \\ = \;& \theta_*^\top \Sigma \theta_* + \theta_*^\top X^\top (XSS^\top X^\top)^{-1} XSS^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\ &\qquad - 2\theta_*^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\ = \;& \theta_*^\top \Sigma \theta_* + \operatorname{tr}(ASBS^\top) - 2\theta_*^\top \Sigma SS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*,\end{aligned}$$

with $A = \Sigma$, and $B = S^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \theta_*^\top X^\top (XSS^\top X^\top)^{-1} XS$, with conditional expectation given XS, simplifying the product $XSS^\top X^\top (XSS^\top X^\top)^{-1} = I$, and using $XSBS^\top X^\top = X\theta_*\theta_*^\top X^\top$:

$$
\begin{aligned}
& \theta_*^\top \Sigma \theta_* \\
& + \operatorname{tr}\big(X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XSS^\top\big) \\
& + \operatorname{tr}(S^\top X^\top (XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top (XSS^\top X^\top)^{-1} XS) \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big] \\
& - 2\theta_*^\top \Sigma X^\top (XX^\top)^{-1} XSS^\top X^\top (XSS^\top X^\top)^{-1} X\theta_* \\
= \; & \theta_*^\top \Sigma \theta_* \\
& + \theta_*^\top X^\top (XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1} X\theta_* \\
& + \operatorname{tr}((XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top) \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big] \\
& - 2\theta_*^\top \Sigma X^\top (XX^\top)^{-1} X\theta_* \text{ by simplifying,} \\
= \; & \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma (I - X^\top (XX^\top)^{-1} X)\theta_* \\
& + \operatorname{tr}((XSS^\top X^\top)^{-1} X\theta_*\theta_*^\top X^\top) \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big],
\end{aligned}
$$

by grouping terms. This leads to, by rearranging terms,

$$
\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}^{(\text{bias})}(\hat\theta)] \; = \; & \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma (I - X^\top (XX^\top)^{-1} X)\theta_* \\
& + \frac{1}{m-n-1} \theta_*^\top X^\top (XX^\top)^{-1} X\theta_* \cdot \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big].
\end{aligned}
$$

Pulling together bias and variance, when m tends to infinity, we get the performance:

$$
\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}_\infty(\hat\theta)] \; = \; & \theta_*^\top (I - X^\top (XX^\top)^{-1} X)\Sigma (I - X^\top (XX^\top)^{-1} X)\theta_* \\
& + \sigma^2 \operatorname{tr}\big((XX^\top)^{-1} X\Sigma X^\top (XX^\top)^{-1}\big).
\end{aligned}
$$

Overall, we get:

$$
\begin{aligned}
\mathbb{E}_{\varepsilon,S}[\mathcal{R}(\hat\theta)] \; = \; & \mathbb{E}_{\varepsilon,S}[\mathcal{R}_\infty(\hat\theta)] + \frac{1}{m-n-1} \operatorname{tr}\big[\Sigma(I - X^\top (XX^\top)^{-1} X)\big] \\
& \cdot \big(\theta_*^\top X^\top (XX^\top)^{-1} X\theta_* + \sigma^2 \operatorname{tr}((XX^\top)^{-1})\big).
\end{aligned}
$$

Thus, as a function of m, we get a descent curve on the right of $m = n$. See Bach (2023a) for a detailed expression obtained after taking the expectation with respect to X. While the limiting bias term typically has a better value than for the under-parameterized regime, for the variance term, the limit when m tends to $+\infty$ does not always go to zero when n tends to infinity. See Bartlett et al. (2020) for conditions under which the end of the double descent curve can lead to good performance when $\sigma^2 > 0$. See illustration in Figure 12.4.

12.3 Global convergence of gradient descent

In Section 9.2.1, we alluded to the property of gradient descent for overparameterized neural networks, which converges to a global minimum of the objective function despite being non-convex. We present more formal arguments in this section, with a general result without proof, as well as a detailed proof for linear neural networks.

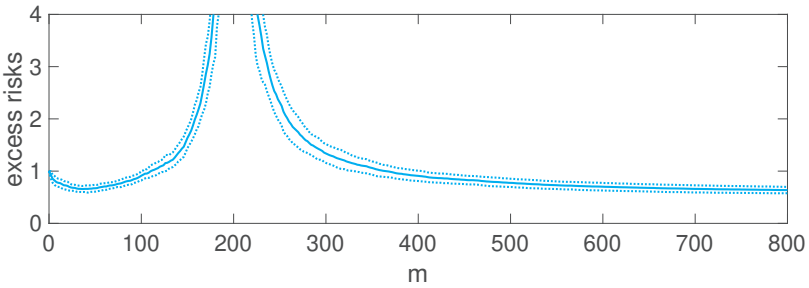


Figure 12.4: Example of a double descent curve, for linear regression with random projections with $n = 200$ observations, in dimension $d = 400$ and a non-isotropic covariance matrix. The data are normalized so that predicting zero leads to an excess risk of 1 and the noise so that the optimal expected risk is $1/4$. The empirical estimate is obtained by sampling 20 datasets and 20 different random projections from the same distribution and averaging the corresponding excess risks

12.3.1 Mean field limits

In this section, we present results from Chizat and Bach (2018), following closely the exposition from Bach and Chizat (2022).[7] More precisely, we consider neural networks with one infinitely wide hidden layer, and we first rescale the prediction function by $1/m$ (which can be obtained by rescaling all output weights by $1/m$) and express it explicitly as an empirical average as

$$f(x) = \frac{1}{m} \sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j)$$

where $\eta_j \in \mathbb{R}$ is the output weight associated to the j-th neuron, and $(w_j, b_j) \in \mathbb{R}^{d+1}$ the corresponding vector of input weights. The key observation is that the prediction function h is the average of m prediction functions $x \mapsto \eta_j \sigma(w_j^\top x + b_j)$, for $j = 1, \dots, m$, with *no sharing of the parameters* (which is not true if extra layers of hidden neurons are added).

To highlight this parameter separability, we define $v_j = \big[\eta_j, w_j^\top, b_j\big]^\top \in \mathbb{R}^{d+2}$ the set of weights associated to the hidden neuron $j \in \{1, \dots, m\}$, and we define the function $\Psi(v) = \Psi(\eta, w^\top, b) : x \mapsto \eta\sigma(x^\top w + b)$, so that the prediction function h is parameterized by $v_1, \dots, v_m \in \mathbb{R}^{d+2}$, which is now

$$h = \frac{1}{m} \sum_{j=1}^{m} \Psi(v_j). \tag{12.13}$$

The expected risk is of the form

$$\mathcal{R}(h) = \mathbb{E}\big[\ell(y, h(x))\big],$$

[7] See also https://www.di.ens.fr/~fbach/ltfp/wide_convergence.html.

which is convex in h for convex loss functions (which is the case throughout the book, even for neural networks, such as the logistic or square loss), but typically non convex in $V = (v_1, \dots, v_m)$. Note that the resulting problem of minimizing a convex function $\mathcal{R}(h)$ for $h = \frac{1}{m}\sum_{j=1}^m \Psi(v_j)$ applies beyond neural networks, for example, for sparse deconvolution (Chizat, 2021).

Reformulation with probability measures. We now define by $\mathcal{P}(\mathcal{V})$ the set of probability measures on $\mathcal{V} = \mathbb{R}^{d+2}$. We can rewrite Eq. (12.13) as

$$h = h(\cdot, v_1, \dots, v_m) = \int_{\mathcal{V}} \Psi(v) d\mu(v),$$

where $\mu = \frac{1}{m}\sum_{j=1}^m \delta_{v_j}$ is an average of Dirac measures at each $v_1, \dots, v_m \in \mathcal{V}$. Following a physics analogy, we will refer to each w_j as a *particle*. When the number m of particles grows, then the empirical measure $\frac{1}{m}\sum_{j=1}^m \delta_{w_j}$ may converge in distribution to a probability measure with a density, often referred to as a *mean field* limit. Our main reformulation will thus consider an optimization problem over probability measures.

The optimization problem we are faced with is equivalent to

$$\inf_{\mu \in \mathcal{P}(\mathcal{V})} \mathcal{R}\Big(\int_{\mathcal{V}} \Psi(v) d\mu(v) \Big), \tag{12.14}$$

with the constraint that μ is an average of m Dirac measures. We now follow a long line of work in statistics and signal processing (Barron, 1993; Kurková and Sanguineti, 2001), and consider the optimization problem *without this constraint*, and relate optimization algorithms for finite but large m (thus acting on $V = (v_1, \dots, v_m)$ in $\mathcal{V}^m$) to a well-defined algorithm in $\mathcal{P}(\mathcal{V})$, like we already did in Section 9.3.2.

Note that we now have a convex optimization problem, with a convex objective in μ over a convex set (all probability measures). However, it is still an infinite-dimensional space that requires dedicated finite-dimensional algorithms. In this section, we focus on gradient descent on w, corresponding to standard practice in neural networks (e.g., back-propagation). For algorithms based on classical convex optimization algorithms such as the Frank-Wolfe algorithm, see Section 9.3.6.

From gradient descent to gradient flow. Our general goal is to study the gradient descent recursion on $V = (v_1, \dots, v_m) \in \mathcal{V}^m$, defined as

$$V_k = V_{k-1} - \gamma m G'(V_{k-1}), \tag{12.15}$$

with

$$G(V) = \mathcal{R}\big(h(\cdot, v_1, \dots, v_m)\big) = \mathcal{R}\Big(\frac{1}{m}\sum_{j=1}^m \Psi(v_j)\Big).$$

In the context of neural networks, this is exactly the back-propagation algorithm. We include the factor m in the step-size to obtain a well-defined limit when m tends to infinity (see below).

For convenience in the analysis, we look at the limit when the step-size γ goes to zero. If we consider a function $W : \mathbb{R} \to \mathcal{V}^m$, with values $W(k\gamma) = V_k$ at $t = k\gamma$, and we interpolate linearly between these points, then we obtain exactly the standard Euler discretization of the ordinary differential equation (ODE) (Suli and Mayers, 2003):

$$\dot{W} = -mG'(W). \tag{12.16}$$

This gradient flow will be our main focus in this paper. As highlighted above, and with extra regularity assumptions, it is the limit of the gradient recursion in Eq. (12.15) for vanishing step-sizes γ. Moreover, under appropriate conditions, stochastic gradient descent, where we only observe an unbiased noisy version of the gradient, also leads in the limit $\gamma \to 0$ to the same ODE (Kushner and Yin, 2003). This allows us to apply our results to probability distributions of the data (x, y), which are not the observed empirical distribution but the unseen test distribution, where, for a single over the data, the stochastic gradients come from the gradient of the loss from a single observation.

Wasserstein gradient flow. Above, we have described a general framework where we want to minimize a function F defined on probability measures:

$$F(\mu) = \mathcal{R}\Big(\int_{\mathcal{V}} \Psi(v) d\mu(v)\Big), \tag{12.17}$$

with an algorithm minimizing $G(v_1, \dots, v_m) = \mathcal{R}\big(\frac{1}{m}\sum_{j=1}^m \Psi(v_j)\big)$ through the gradient flow $\dot{V} = -mG'(V)$, with $V = (v_1, \dots, v_m)$.

As shown in a series of works concerned with the infinite width limit of two-layer neural networks (Nitanda and Suzuki, 2017; Chizat and Bach, 2018; Mei et al., 2018; Sirignano and Spiliopoulos, 2020; Rotskoff and Vanden-Eijnden, 2018), this converges to a well-defined mathematical object called a Wasserstein gradient flow (Ambrosio et al., 2008). This is a gradient flow derived from the Wasserstein metric on the set of probability measures, which is defined as (Santambrogio, 2015),

$$W_2(\mu, \nu)^2 = \inf_{\gamma \in \Pi(\mu,\nu)} \int \|v - w\|_2^2 d\gamma(v, w),$$

where $\Pi(\mu, \nu)$ is the set of probability measures on $\mathcal{V} \times \mathcal{V}$ with marginals μ and ν. In a nutshell, the gradient flow is defined as the limit when γ tends to zero of the extension of the following discrete-time dynamics:

$$\mu(t + \gamma) = \inf_{\nu \in \mathcal{P}(\mathcal{V})} F(\nu) + \frac{1}{2\gamma} W_2(\mu(t), \nu)^2.$$

When applying such a definition in a Euclidean space with the Euclidean metric, we recover the usual gradient flow $\dot{\mu} = -F'(\mu)$, but here with the Wasserstein metric, this defines a specific flow on the set of measures. When the initial measure is a weighted sum of Diracs, this is precisely asymptotically (when $\gamma \to 0$) equivalent to backpropagation. When initialized with an arbitrary probability measure, we obtain a partial differential

equation (PDE), satisfied in the sense of distributions. Moreover, when the sum of Diracs converges in distribution to some measure, the flow converges to the solution of the PDE. More precisely, assuming $\Psi : \mathbb{R}^{d+1} \to \mathcal{H}$ a Hilbert space (in our neural network example, $\mathcal{H}$ is the space of square-integrable functions on $\mathbb{R}^d$), and $\mathcal{R}'(h) \in \mathcal{H}$ the gradient of $\mathcal{R}$, we consider the *mean potential*

$$J(v|\mu) = \left\langle \Psi(v), \mathcal{R}'\Big(\int_{\mathcal{V}} \Psi(w) d\mu(w) \Big) \right\rangle, \tag{12.18}$$

so that the gradient flow equation on each particle becomes $\dot{v}_j = -J'(v_j|\mu)$ (where μ is the time-dependent aggregation of all particles).

The PDE is then the continuity equation (see, e.g., Evans, 2022):

$$\partial_t \mu_t(v) = \operatorname{div}(\mu_t(v) J'(v|\mu_t)), \tag{12.19}$$

which is understood in the sense of distributions. The following result formalizes this behavior (see Chizat and Bach (2018) for details and a more general statement).

Theorem 12.1 *Assume that $\mathcal{R} : \mathcal{H} \to [0, +\infty[$ and $\Psi : \mathcal{V} = \mathbb{R}^{d+2} \to \mathcal{H}$ are (Fréchet) differentiable with Lipschitz differentials, and that R is Lipschitz on its sublevel sets. Consider a sequence of initial weights $(v_j(0))_{j \geq 1}$ contained in a compact subset of $\mathcal{V}$ and let $\mu_{t,m} = \frac{1}{m} \sum_{j=1}^m v_j(t)$ where $(v_1(t), \dots, v_m(t))$ solves the ODE (12.16). If $\mu_{0,m}$ weakly converges to some $\mu_0 \in \mathcal{P}(\mathcal{V})$ then $\mu_{t,m}$ weakly converges to μ_t where $(\mu_t)_{t \geq 0}$ is the unique weakly continuous solution to (12.19) initialized with μ_0.*

In the following paragraph, we will study the solution of this PDE (i.e., the Wasserstein gradient flow), interpreting it as the limit of the gradient flow in Eq. (12.16) when the number of particles m tends to infinity.

Global convergence. We consider the Wasserstein gradient flow defined above, which leads to the PDE in Eq. (12.19). We aim to understand when we can expect that when $t \to \infty$, μ_t converges to a global minimum of F defined in Eq. (12.17). Obtaining a global convergence result is not out of the question because F is a convex functional defined on the convex set of probability measures. However, it is nontrivial because, with our choice of the Wasserstein geometry on measures, which allows an approximation through particles, the flow has some stationary points that are not the global optimum.

We only consider an informal general result without technical assumptions before referring to Bach and Chizat (2022) for a formal simplified result and Chizat and Bach (2018) for the general result.

Theorem 12.2 (Informal) *If the support of the initial distribution includes all directions in $\mathbb{R}^{d+1}$, and if the function Ψ is positively 2-homogeneous, then if the Wasserstein gradient flow weakly converges to a distribution, it can only be to a global optimum of F.*

Chizat and Bach (2018) present another version of this result that allows for *partial* homogeneity (e.g., with respect to a subset of variables) of degree 1 is proven, at the cost of a more technical assumption on the initialization. For neural networks, we have $\Psi(\eta, w^\top, b)(x) = m\eta\sigma(w^\top x + b)$, and this more general version applies. For the

classical ReLU activation function $u \mapsto \max\{0, u\}$, we get a positively 2-homogeneous function, as required in the previous statement. A simple way to spread all directions is to initialize neural network weights from Gaussian distributions, which is standard in applications (Goodfellow et al., 2016).

From qualitative to quantitative results? Our result states that for infinitely many particles, we can only converge to a global optimum (note that we cannot show that the flow always converges). However, it is only a qualitative result in comparison with what is known for convex optimization problems in Chapter 5:

- This is only for $m = +\infty$, and we cannot provide an estimation of the number of particles needed to approximate the mean-field regime that is not exponential in t (see such results, e.g., by Mei et al., 2019).
- We cannot provide an estimation of the performance as a function of time that would give an upper bound on the running time complexity.

Moreover, our result does not apply beyond a single hidden layer, and understanding the non-linear infinite width limits for deeper networks is an important research area (Nguyen and Pham, 2020; Araújo et al., 2019; Fang et al., 2021; Hanin and Nica, 2019; Sirignano and Spiliopoulos, 2021; E and Wojtowytsch, 2020; Yang and Hu, 2020).

In the remainder of this section, to present a simpler analysis, we focus on linear neural networks and first reformulate them as optimizing over positive definite matrices.

12.3.2 From linear networks to positive definite matrices

We consider "linear" neural networks, that is, neural networks with no activation function. For example, for $x \in \mathbb{R}^d$, we consider $f(x) = UV^\top x \in \mathbb{R}^k$, where $U \in \mathbb{R}^{k \times m}$ and $V \in \mathbb{R}^{d \times m}$. This is a linear function $f(x) = \Theta x$, with Θ of the form $\Theta = UV^\top \in \mathbb{R}^{k \times d}$. Assuming that we minimize some smooth convex risk function $G : \mathbb{R}^{k \times d} \to \mathbb{R}$, we aim to minimize $G(UV^\top)$.

It can be rewritten as the function G applied to a linear projection of the matrix $\binom{U}{V}\binom{U}{V}^\top = \left(\begin{smallmatrix} UU^\top & UV^\top \\ VU^\top & VV^\top \end{smallmatrix}\right)$, which is of the form $WW^\top$ with $W = \binom{U}{V} \in \mathbb{R}^{(k+d) \times m}$. Thus, we can analyze instead the minimization of functions of the form $G(WW^\top)$ for $W \in \mathbb{R}^{d \times m}$, where G is a smooth convex function defined on positive semidefinite matrices of size d.

The goal of this section is now to minimize a convex function G over positive-semidefinite (PSD) matrices, using plain gradient descent techniques on a non-linear parameterization of such matrices. This is done to illustrate optimization for neural networks, noting that faster algorithms based on projected gradient descent presented in Chapter 5 could also be used. We already studied a special case in Section 12.1.3, where the ordinary differential equation could be integrated in closed form, while, here, we rely on more qualitative arguments.

12.3.3 Global convergence for positive definite matrices

We consider a twice continuously differentiable convex function $G : \mathbb{R}^{d\times d} \to \mathbb{R}$ (which only needs to be defined on symmetric matrices). We consider m vectors $w_1, \dots, w_m \in \mathbb{R}^d$ put into a matrix $W = (w_1, \dots, w_m) \in \mathbb{R}^{d\times m}$, and the cost function $F(W) = G\big(WW^\top\big)$, where we have $WW^\top = \sum_{j=1}^m w_j w_j^\top$. This is thus an instance of the framework developed in Section 12.3.1, but, since the space of quadratic functions is finite-dimensional, there is no need to let m tend to infinity, and we can keep $m \leqslant d$.

We consider the gradient flow $\dot W = -\frac{1}{2}F'(W)$, that is, $W'(t) = -\frac{1}{2}F'(W(t))$, where the factor $\frac{1}{2}$ was added so simplify formulas later. Since F is twice differentiable, this ordinary differential equation (ODE) is defined for all $t \geqslant 0$. To compute the gradient of F, we perform an asymptotic expansion as follows:

$$\begin{aligned} F(W+\Delta) &= G\big(WW^\top + \Delta W^\top + W\Delta^\top + o(\|\Delta\|_2)\big) \\ &= F(W) + \operatorname{tr}\big[G'\big(WW^\top\big)\big(\Delta W^\top + W\Delta^\top\big)\big] + o(\|\Delta\|_2) \\ &= F(W) + 2\operatorname{tr}\big[\Delta^\top G'\big(WW^\top\big)W\big] + o(\|\Delta\|_2), \end{aligned}$$

so that $F'(W) = 2G'\big(WW^\top\big)W$, and the flow becomes $\dot W = -G'\big(WW^\top\big)W$. By projecting onto each of the m columns of W, this leads to the following flow for each "particle" $w_j \in \mathbb{R}^d$:

$$\dot w_j = -G'\big(WW^\top\big)w_j,$$

which is a linear ODE, but with a time-dependent matrix $G'\big(WW^\top\big)$ which depends on the aggregation of all particles since $WW^\top = \sum_{j=1}^m w_j w_j^\top$.

We denote $M = WW^\top$ and $A = G'(M)$, which are functions of time defined for all time $t \geqslant 0$. We then have:

$$\dot M = \dot W W^\top + W\dot W^\top = -G'(M)M - MG'(M) = -AM - MA.$$

Preservation of rank. If at time zero $M = WW^\top$ has full rank, then the rank is preserved throughout the flow. This is a simple consequence of the ODE for $r(M) = \log\det(M)$, equal to

$$\dot r = \operatorname{tr}\big[M^{-1}\dot M\big] = \operatorname{tr}\big[M^{-1}(-AM - MA)\big] = -2\operatorname{tr}(A).$$

Thus, since A is continuous for all positive times, the log determinant is finite for all times as soon as it exists at initialization, and we thus obtain a full rank matrix. If $m \geqslant d$, which corresponds to an over-parameterized situation, and the columns of W are initialized randomly (e.g., from a standard Gaussian random variable), then $WW^\top$ indeed has full rank.

Exercise 12.3 *(♦) Show that if at initialization, $M = WW^\top$ has rank $r \leqslant \min\{d, m\}$, then M has rank r at all times.*

Global optimality conditions. The problem of minimizing $G(M)$ over PSD matrices has the following optimality condition: (a) $\mathrm{tr}[MG'(M)] = 0$ and (b) $G'(M) \succcurlyeq 0$. Note that once (b) is satisfied, (a) is equivalent to $MG'(M) = 0$.[8]

- *Necessary conditions (no need for convexity).* If M is optimal, then for all Δ such that $M + \Delta \succcurlyeq 0$, $G(M + \Delta) - G(M) \geqslant 0$. When Δ is small, this leads to $\mathrm{tr}[\Delta G'(M)] \geqslant 0$.

  Taking Δ small along $-M$ or M, we get: $\mathrm{tr}[MG'(M)] = 0$ as necessary condition.

  Taking $\Delta = uu^\top$ for all $u \in \mathbb{R}^d$, we get $G'(M) \succcurlyeq 0$ as a necessary condition.
- *Sufficient conditions.* If the conditions are met, then for any matrix $N \succcurlyeq 0$, we get from the subgradient inequality for the convex function R:

$$G(N) \geqslant G(M) + \mathrm{tr}\left[G'(M)(N - M)\right].$$

  Using condition (a), we get : $\mathrm{tr}\left[G'(M)M\right] = 0$, while condition (b) ensures that $\mathrm{tr}\left[G'(M)N\right] \geqslant 0$. Thus, M is a global optimum.

If M is invertible, the optimality conditions simplify to $G'(M) = 0$.

Global convergence. (♦) If the flow in M is initialized with a full-rank matrix and converges to some M_∞,[9] we now show that it satisfies the two optimality conditions above (and thus, it has to be a global optimum). Note that while we know that M is invertible for all time $t \geqslant 0$, it is often not the case for M_∞ (see examples below).

The first condition (a) is a direct consequence of $-G'(M_\infty)M_\infty - M_\infty G'(M_\infty) = 0$ (by taking the trace), which is satisfied at convergence (this is the stationary condition, stating that all particles stop). The difficult part is to show the second condition (b), which can be interpreted as ensuring that no other potential particles could enter and increasing the rank of M while reducing the cost function.

We now assume that $A_\infty = G'(M_\infty)$ is not PSD, that is, $\lambda_{\min}(A_\infty) < 0$. We choose $\eta > 0$ such that $\lambda_{\min}(A_\infty) < -\eta$, and $-\eta$ is not an eigenvalue of A_∞ (which is possible because there are at most d distinct eigenvalues). This implies that for u such that $\|u\|_2 = 1$ and $u^\top A_\infty u = -\eta$,

$$\eta = -u^\top A_\infty u < \|u\|_2 \|A_\infty u\|_2 = \|A_\infty u\|_2$$

by Cauchy-Schwarz inequality and the impossibility of having $A_\infty u = -\eta u$ (which is the equality condition for Cauchy-Schwarz inequality). We denote by $\beta > \eta$ the minimal value of such $\|A_\infty u\|_2$ (for all u that satisfies $\|u\|_2 = 1$ and $u^\top A_\infty u = -\eta$).

The idea is to show that sufficiently close to convergence, once a particle has a direction in

$$K = \left\{u \in \mathbb{R}^d,\ \|u\|_2 = 1,\ u^\top A_\infty u < -\eta\right\},$$

[8] For two PSD matrices A and B of the same sizes, $AB = 0 \Leftrightarrow \mathrm{tr}(AB) = 0$.

[9] It does under basic assumptions on R, such as piecewise analyticity, see Bolte et al. (2006).

its direction never gets out of K, and that it leads to a contradiction (the set K is not empty because $\lambda_{\min}(A_\infty) < -\eta$).

We now introduce the time dependence explicitly.

Choice of particle close to convergence (⧫). We have $M(t) \to M_\infty$. Thus there exists t_0 such that $\|A(t) - A_\infty\|_{\rm op} \leqslant \varepsilon$, for all $t \geqslant t_0$, with ε well chosen (small enough).

Let $y_0 \in \mathbb{R}_+ K$, $y_0 \neq 0$ (it exists since K is not empty). Since $W(t_0) \in \mathbb{R}^{d\times m}$ has full rank equal to d, then there exists $\alpha_0 \in \mathbb{R}^m$ such that $y_0 = W(t_0)\alpha_0$.

We then consider a particle $z(t) = W(t)\alpha_0 \in \mathbb{R}^d$. By construction, $z'(t) = \dot{W}(t)\alpha_0 = -A(t)W(t)\alpha_0 = -A(t)z(t)$, and $z(t_0) = y_0 \in \mathbb{R}_+ K$. We now show by contradiction that we must have $z(t) \in \mathbb{R}_+ K$ for all $t \geqslant t_0$. If t_1 is the smallest $t \geqslant t_0$ such that $z(t) \notin \mathbb{R}_+ K$ (which is assumed to exist by contradiction), then by continuity, $z(t_1) \in \mathbb{R}_+ \partial K$, that is, $z(t_1)^\top A_\infty z(t_1) = -\eta z(t_1)^\top z(t_1)$. We then have, with $z_1 = z(t_1)$, and using that $z'(t_1) = -A(t_1)z(t_1)$:

$$\begin{aligned}
\frac{d}{dt}\frac{z(t)^\top A_\infty z(t)}{z(t)^\top z(t)}\Big|_{t=t_1} &= 2\frac{z(t_1)^\top A_\infty z'(t_1)}{z(t_1)^\top z(t_1)} - 2\frac{z(t_1)^\top A_\infty z(t_1)}{z(t_1)^\top z(t_1)}\frac{z'(t_1)^\top z(t_1)}{z(t_1)^\top z(t_1)} \\
&= -2\frac{z_1^\top A_\infty A(t_1) z_1}{z_1^\top z_1} + 2\frac{z_1^\top A_\infty z_1}{z_1^\top z_1}\frac{z_1^\top A(t_1) z_1}{z_1^\top z_1} \\
&= -2\frac{z_1^\top A_\infty^2 z_1}{z_1^\top z_1} + 2\frac{z_1^\top A_\infty (A_\infty - A(t_1)) z_1}{z_1^\top z_1} + 2\frac{z_1^\top A_\infty z_1}{z_1^\top z_1}\frac{z_1^\top A(t_1) z_1}{z_1^\top z_1}.
\end{aligned}$$

Using that $\|A(t_1) - A_\infty\|_{\rm op} \leqslant \varepsilon$ and $z(t_1)^\top A_\infty z(t_1) = -\eta z(t_1)^\top z(t_1)$, this leads to

$$\begin{aligned}
\frac{d}{dt}\frac{z(t)^\top A_\infty z(t)}{z(t)^\top z(t)}\Big|_{t=t_1} &\leqslant -2\frac{z_1^\top A_\infty^2 z_1}{z_1^\top z_1} + 2\frac{\|A_\infty z_1\|_2 \varepsilon}{\|z_1\|_2} + 2\eta^2 + 2\eta\varepsilon \\
&\leqslant -2\beta^2 + 2\eta^2 + 2\|A_\infty\|_{\rm op}\varepsilon + 2\eta\varepsilon,
\end{aligned}$$

which is strictly negative for ε small enough, which is a contradiction because it would imply that for t just above t_1, $\frac{z(t)^\top A_\infty z(t)}{z(t)^\top z(t)} < \frac{z(t_1)^\top A_\infty z(t_1)}{z(t_1)^\top z(t_1)} = -\eta$, and thus, $z(t) \in \mathbb{R}_+ K$.

Final contradiction. (⧫) We now have that the particule $z(t)$ is in $\mathbb{R}_+ K$ for all $t \geqslant t_0$. We then have for all $t \geqslant t_0$,

$$\frac{d}{dt} z(t)^\top z(t) = -2z(t)^\top A(t) z(t) \geqslant 2\big(- z(t)^\top A_\infty z(t) - \|z(t)\|_2^2 \varepsilon\big) \geqslant 2(\eta - \varepsilon)\|z\|_2^2,$$

leading to, after integration, $\|z(t)\|_2^2 \geqslant \|z(t_0)\|_2^2 \exp(2(\eta-\varepsilon)(t-t_0))$, and thus a divergence. This contradicts the convergence of $z(t) = W(t)\alpha_0$.

12.3.4 Special case of Oja flow

As an illustration of the convergence results above, we consider the function

$$G(M) = \frac{1}{2}\|M - C\|_F^2,$$

for a symmetric matrix $C \in \mathbb{R}^{d \times d}$, for which the flow can integrated in closed form. We have $G'(M) = M - C$, and thus the following gradient flows:

$$\dot{W} = -G'(WW^\top)W = CW - WW^\top W \quad \text{and} \quad \dot{M} = CM + MC - 2M^2.$$

If we initialize $W(0) = V \in \mathbb{R}^{d \times m}$, we obtain a solution in closed-form (as can be checked by taking derivatives and showing that $\dot{M} = CM + MC - 2M^2$), as

$$M = WW^\top = \exp(Ct)V\big(I + V^\top C^{-1}(\exp(2Ct) - 1)V\big)^{-1}V^\top \exp(Ct).$$

This is the Oja flow, up to a change of variable (Yan et al., 1994). It is interesting to note that if we use $m \leqslant d$ particles, the rank of $WW^\top$ is always less than $m \leqslant d$, and in fact, the same as the rank of the initialization. The global minimizer of R on PSD matrices is the positive part of C, whose rank can be strictly smaller than m, and thus, it cannot converge to the global optimum of R. The minimum number of particles is the number of positive eigenvalues of C.

Vanishing initialization. If $V = \sqrt{\alpha}I \in \mathbb{R}^{d \times d}$, we get:

$$M = \alpha \exp(2Ct)\big(I + \alpha C^{-1}(\exp(2Ct) - 1)\big)^{-1}.$$

Then, M is a spectral variant of C, thus with the same eigenvectors, and eigenvalues $m = \frac{\alpha e^{2ct}}{1+\alpha c^{-1}(e^{2ct}-1)} \approx \frac{c}{1+e^{-2ct}c/\alpha}$ for small α, where c is the corresponding eigenvalue of c.

Thus, when α tends to zero (and therefore closer to the stationary point), the eigenvalues m stay at zero until they increase to the final positive values c, and this increase happens around $t = \frac{1}{2c}\log\frac{1}{\alpha}$. We thus observe incremental learning of each eigenvector, with each eigenvector corresponding to a positive eigenvalue c, which is a very different optimization dynamic from the one obtained from projected gradient descent, which corresponds to $m = c(1 - e^{-t})$ where all eigenvectors come in together. This incremental learning at different time scales is common in non-convex optimization, see Saxe et al. (2019); Gidel et al. (2019) for linear networks, and Berthier (2023); Pesme and Flammarion (2023) for diagonal linear networks, where precise statements can be made.

12.4 Lazy regime and neural tangent kernels (⧫)

For over-parameterized one-hidden layer neural networks, with prediction functions of the form (note the rescaling by $1/m$):

$$f(x) = \frac{1}{m}\sum_{j=1}^{m} \eta_j \sigma(w_j^\top x + b_j) = \frac{1}{m}\sum_{j=1}^{m} \Psi(v_j),$$

with Ψ defined in Section 12.3.1, we have seen two types of learning procedures in Chapter 9 when the number of neurons grows unbounded:

- Optimizing over all layers, leading to the mean-field limit presented in Section 12.3.1 and with the non-Hilbertian norm γ_1. This corresponds to all parameters being initialized with a scaling that does not depend on m.
- Optimizing over the last layer only, leading to the kernel regime, with the Hilbertian norm γ_2 defined in Section 9.5. The input weights (w_j, b_j) are all sampled without scaling. For the output weights, η_j can be $O(1)$ or $O(\sqrt{m})$ if initialized with zero-mean (so that the overall norm of the prediction function remains bounded in high probability). Since this is a convex optimization problem, scaling does not matter as much.

Lazy training. We now consider a third regime, which we refer to as the "lazy" regime, following Chizat et al. (2019). It corresponds to initializing each η_j with a scaling proportional to $\sqrt{m}$. This is made possible by having zero mean initializations so that a mean of m terms is of order $O(1/\sqrt{m})$ and not $O(1)$ (leading to an overall predictor that remains $O(1)$). We will formalize this training regime by seeing this model as a diverging constant α (here $\sqrt{m}$) times a classical model with a mean-field limit.

We thus consider the minimization of the objective function $G(V) = \mathcal{R}(h(V))$ with respect to $V = (v_1, \dots, v_m)$. In the lazy regime, we end up minimizing $\mathcal{R}(\alpha h(V))$ with a scaling factor α that tends to infinity, using a gradient flow on V, started at $V(0)$ such that $\alpha h(V(0))$ remains bounded. In our neural network example, $\alpha = \sqrt{m}$, and h is the regular neural network.

We consider the gradient flow to minimize $G(V)$, with a step-size $1/\alpha^2$ (scaling adapted to have a non-trivial dynamic), that is,

$$\frac{d}{dt} V(t) = -\frac{1}{\alpha} Dh(V)^\top \mathcal{R}'(\alpha h(V(t))), \tag{12.20}$$

where $Dh(V)$ is the differential of h at V (that is, a linear function from $\mathbb{R}^{(d+2)\times m}$ to $\mathcal{H}$). For the predictor $\alpha h(V)$, we get:

$$\frac{d}{dt}[\alpha h(V(t))] = -Dh(V(t))Dh(V(t))^\top \mathcal{R}'(\alpha h(V(t))). \tag{12.21}$$

At initialization $t = 0$, $\alpha h(V(t))$ remains bounded by construction, and since the optimization will tend to make the predictor better and better, we can expect it to remain bounded. Thus, we can expect $\mathcal{R}(\alpha h(V(t)))$ and $\mathcal{R}'(\alpha h(V(t)))$ to be $O(1)$. Thus, from Eq. (12.20), we obtain that the parameters V change at rate $O(1/\alpha)$, while from Eq. (12.21) the predictor changes at a rate which is independent of α. Thus, in the limit of large α, the parameters only move infinitesimally, while the predictor still makes significant progress, hence the name lazy training (see more formal arguments by Chizat et al., 2019).

Equivalent linear model. Since parameters move infinitesimally, the model $\alpha h(V)$ behaves like an affine model, $\alpha h(V(0)) + \alpha Dh(\alpha V(0))(V - V(0))$, and thus the corresponding cost function $\mathcal{R}(\alpha h(V))$ behaves like a convex function, hence leading to attractive global convergence results for neural network training (see, e.g. Du et al., 2018).

However, since we have an explicit linear model, the lazy regime also leads to a positive definite kernel, which we now define (with the same properties as traditional kernel methods in Chapter 7).

Neural tangent kernel (♦). If we assume that $h(V(0)) = 0$ (e.g., for neural networks, assuming that all initial neurons come by pairs with the same input weights and opposite output weights), then the affine model has a linear part proportional to $Dh(\alpha V(0))V$. We can thus associate to it a kernel, referred to as the neural tangent kernel (Jacot et al., 2018).

To make things concrete, for neural networks with one hidden layer, $h(x, v_1, \dots, v_m) = \frac{1}{\sqrt{m}} \sum_{j=1}^m \eta_j \sigma(w_j^\top x + b_j)$, the corresponding features, for each $j \in \{1, \dots, m\}$, are

$$\begin{aligned}
\text{derivative with respect to } \eta_j &: \frac{1}{\sqrt{m}} \sigma(w_j(0)^\top x + b_j(0)) \\
\text{derivative with respect to } w_j &: \frac{1}{\sqrt{m}} \eta_j(0) \sigma'(w_j(0)^\top x + b_j(0)) x \\
\text{derivative with respect to } b_j &: \frac{1}{\sqrt{m}} \eta_j(0) \sigma'(w_j(0)^\top x + b_j(0)).
\end{aligned}$$

When the initialization of neuron weights is random, we get the equivalent kernel by the law of large numbers:

$$k(x, x') = \mathbb{E}\big[\sigma(w^\top x + b)\sigma(w^\top x' + b) + \mathbb{E}\big[\sigma'(w^\top x + b)\sigma'(w^\top x' + b)(x^\top x' + 1)\big],$$

whose first part is the traditional random feature kernel, but which has an additional part, which makes for a richer model but cannot correct the limitations of kernel methods (see, e.g., Bietti and Bach, 2021, and references therein).

12.5 Conclusion

In this chapter, we have presented a series of results related to over-parameterized models, confirming that some form of regularization is needed: as opposed to previous chapters, where (except for boosting procedures in Section 10.3) an explicit penalty was put on the model parameters, overfitting is avoided by "computational regularization", that is, through the implicit bias of gradient descent techniques. This was formally shown for linear models, but this extends more generally (see, e.g., Lyu and Li, 2019; Chizat and Bach, 2020)

We also described how over-parameterization, while not detrimental to generalization performance, can be a blessing in terms of optimization, with qualitative results showing global convergence for infinitely overparameterized problems. Obtaining non-asymptotic results (in terms of convergence times of number of neurons) remains an active area of research.