Proceedings of Machine Learning Research vol 178:1–29, 2022 35th Annual Conference on Learning Theory

On the Benefits of Large Learning Rates for Kernel Methods

Gaspard Beugnot GASPARD.BEUGNOT@INRIA.FR
Inria, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France

Julien Mairal JULIEN.MAIRAL@INRIA.FR
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

Alessandro Rudi ALESSANDRO.RUDI@INRIA.FR
Inria, École normale supérieure, CNRS, PSL Research University, 75005 Paris, France

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

This paper studies an intriguing phenomenon related to the good generalization performance of estimators obtained by using large learning rates within gradient descent algorithms. First observed in the deep learning literature, we show that such a phenomenon can be precisely characterized in the context of kernel methods, even though the resulting optimization problem is convex. Specifically, we consider the minimization of a quadratic objective in a separable Hilbert space, and show that with early stopping, the choice of learning rate influences the spectral decomposition of the obtained solution on the Hessian's eigenvectors. This extends an intuition described by Nakkiran (2020) on a two-dimensional toy problem to realistic learning scenarios such as kernel ridge regression. While large learning rates may be proven beneficial as soon as there is a mismatch between the train and test objectives, we further explain why it already occurs in classification tasks without assuming any particular mismatch between train and test data distributions.

Keywords: Optimization; Statistical Learning; Kernel methods

1. Introduction

Gradient descent methods are omnipresent in machine learning, and a lot of effort has been devoted to better understand their theoretical properties. Optimal rates of convergence have been well characterized for minimizing convex functions in various contexts, including, for instance, stochastic optimization (Nemirovski et al., 2009). For supervised learning, one is however more interested in the statistical optimality of the resulting estimator rather than in the ability to quickly optimize a training objective (Bottou and Bousquet, 2007). When considering both optimization and statistical questions, gradient descent methods were proven to be optimal under many assumptions (Yao et al., 2007; Pillaud-Vivien et al., 2018a).

An important observation for this paper is that gradient descent algorithms typically require to tune some learning rate, or step size, to achieve the best performance. This has been thoroughly investigated in the optimization literature. For convex smooth problems in particular, the influence of step size on convergence rates is well understood (Nesterov, 2018). However, recent empirical studies have highlighted a surprising aspect of this parameter: when using gradient descent methods on neural networks, *large* learning rates were found to be useful for obtaining *good generalization properties*, or in other words, good statistical performance (Jastrzebski et al., 2021), even though they may be sub-optimal from an optimization point of view.

This paper aims at understanding this phenomenon from a broad but simple perspective, where both the function F we optimize and the function R used to evaluate the statistical performance are quadratic forms of some separable Hilbert space $\mathcal{H}$. Specifically, we assume that

$$\forall \theta \in \mathcal{H}, \quad F(\theta) = \frac{1}{2}\left\|\theta - \theta^\star\right\|_{\mathsf{T}}^2 + \mathsf{cst}, \quad \text{and} \quad R(\theta) = \frac{1}{2}\left\|\theta - \nu^\star\right\|_{\mathsf{U}}^2, \tag{1}$$

where $\|\cdot\|_A$ denotes the norm on $\mathcal{H}$ induced by a positive definite operator A, *i.e.*, $\|\theta\|_A^2 = \langle \theta, A\theta \rangle$ for any θ in $\mathcal{H}$. With Eq. (1), F and R are characterized by positive definite operators T and U, along with their minimizers denoted by $\theta^\star$ and $\nu^\star$, respectively. The constant value cst does not affect the optimization problem and can be safely ignored in the rest of this presentation. The model from Eq. (1) captures a large class of problems such as learning with kernels, detailed in Section 4, but we give here a simple example with ridge regression.

Example 1 (T and U with Ridge Regression.) *Let $x_1, \dots, x_n$ be data points in $\mathbb{R}^d$, and $y_1, \dots, y_n$ prediction variables, with $n \geq d$. Define $X \in \mathbb{R}^{n \times d}$ the data matrix. We consider the ridge regression estimator with regularization $\lambda > 0$, which is defined as the minimum of*

$$\forall \theta \in \mathbb{R}^d, \quad F(\theta) = \frac{1}{n}\sum_{i=1}^n \frac{1}{2}(\theta^\top x_i - y_i)^2 + \frac{\lambda}{2}\left\|\theta\right\|^2 = \frac{1}{2n}\left\|X\theta - y\right\|^2 + \frac{\lambda}{2}\left\|\theta\right\|^2.$$

F is a quadratic function of θ, which can be rewritten as in Eq. (1) with

$$\forall \theta \in \mathbb{R}^d, \quad F(\theta) = \frac{1}{2}\left\|\theta - \nu^\star\right\|_{\mathsf{T}}^2 + \mathsf{cst}, \quad \textit{with} \quad \begin{cases} \nu^\star &= \frac{1}{n}\left(\frac{1}{n}X^\top X + \lambda \mathbf{I}_d\right)^{-1} X^\top y \\ \mathsf{T} &= \frac{1}{n}(X^\top X + \lambda \mathbf{I}_d). \end{cases}$$

Assuming that the output can be written $y_i = x_i^\top \nu^\star + \epsilon_i$ with $\nu^\star \in \mathbb{R}^d$ and ϵ_i some independent, zero-mean noise, then the population loss *is $\mathcal{P}(\theta) = \mathbb{E}\, 1/2\left(\theta^\top x_i - y_i\right)^2$, and the* excess risk *defined by $R(\theta) = \mathcal{P}(\theta) - \inf_\nu \mathcal{P}(\nu)$ is given with*

$$R(\theta) = \mathbb{E}\left[\frac{1}{2}\left((\theta - \nu^\star)^\top x\right)^2\right] = \frac{1}{2}\left\|\theta - \nu^\star\right\|_{\mathsf{U}}^2, \quad \textit{with} \quad \mathsf{U} = \mathbb{E}\left[xx^\top\right].$$

In this example, a discrepancy between train and test losses (between T and U) may occur in particular situations (*e.g.*, presence of data augmentation during training, or simply mismatch between train and test distributions). The next example shows that such a mismatch may be in fact frequent for classification problems, even when train and test distributions do match.

Example 2 (Discrepancy between train and test losses in classification with separable classes.) *The scenario described in Eq. (1) is particularly evident in the context of binary classification, when the classes are separable by a non-zero margin. This is considered a typical situation in many learning scenarios of interest, as classification over natural images—motivating the wide use of large-margin based classifiers in the field (Rawat and Wang, 2017). We highlight here, that in this context, the loss we are using for training is not the best loss to consider for the test error, as discussed next. More precisely, consider a classification problem with two classes with non-zero margin. Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, 1\}$ be the input and the output space. Denote by $\rho(x, y) = \rho_{\mathcal{X}}(x)\rho(y|x)$ the probability distribution describing the classification problem, where $\rho_{\mathcal{X}}$ is the marginal probability*

over $\mathcal{X}$, while $\rho(y|x)$ is the conditional probability of y given x. The error that we would like to minimize is the binary error on the population, i.e. $B(\theta) = \mathbb{P}[\mathrm{Sign}[\theta(x)] \neq y]$ for a model θ. Let $\nu^\star$ be a function minimizing the binary error and $\mathcal{H}$ be the class of models under consideration. Assume, for simplicity, that $\mathcal{H}$ is a RKHS with norm $\|\cdot\|$ (Aronszajn, 1950), i.e.*, there exists a map $\phi : \mathcal{X} \to \mathcal{H}$ such that the function in $\mathcal{H}$ are characterized as $\theta(x) = \langle \phi(x), \theta \rangle$, for any θ in $\mathcal{H}$. It has been shown by Pillaud-Vivien et al. (2018a) (in particular, Lemma 1 and Appendix A, Theorem 13), that in the context of two classes separated by a non-zero margin and whose conditional probability is regular enough, then $\nu^\star$ is in $\mathcal{H}$ and moreover $B(\theta) - B(\nu^\star) \leq e^{-c/\|\theta - \nu^\star\|}$ for some constant c. Therefore, the binary error decreases exponentially in terms of the Hilbert norm $\|\cdot\|$. On the other hand, the norm minimized at training time is some smooth convex surrogate of the binary loss, as, for example, the quadratic loss.*

In this case, the trained vector may be obtained by minimizing the population loss (in fact, a regularized empirical version, but we omit this fact here for simplicity) such that $F(\theta) = \mathbb{E}(\theta(x) - y)^2$. Noting that $\int (\theta(x) - \theta^\star(x))^2 \mathrm{d}\rho(x) = \|\mathsf{T}^{1/2}(\theta - \theta^\star)\|^2 = \|\theta - \theta^\star\|_{\mathsf{T}}$ for $\mathsf{T} = \int \phi(x)\phi(x)^\top \mathrm{d}\rho_{\mathcal{X}}(x)$, and $\theta^\star = \nu^\star$ under the considered conditions (see the same paper), we have

$$F(\theta) - F(\theta^\star) = \int (\theta(x) - \theta^\star(x))^2 \mathrm{d}\rho_{\mathcal{X}}(x) = \|\theta - \theta^\star\|_{\mathsf{T}}^2.$$

This is a typical case, where there is a discrepancy between the error of interest

$$R(\theta) - R(\nu^\star) = \|\theta - \nu^\star\|^2,$$

which, if optimized, would lead to an exponential decrease of the classification error, and the loss that is instead optimized by the algorithm at training time, i.e. F, for which we have only the slower rate $B(\theta) - B(\nu^\star) \leq (F(\theta) - F(\nu^\star))^\alpha$, with $\alpha \in (1/2, 1)$ (see, e.g. Audibert (2004) or Audibert and Tsybakov (2007) for what concerns the CAR assumption).

In this paper, we are interested in understanding in which regime large learning rates with early stopping could be useful for kernel methods, even if they are suboptimal from an optimization point of view. We consider indeed the optimization of F in Eq. (1) with plain gradient descent, starting from a vector θ_0 in $\mathcal{H}$ with step-size η, and we distinguish between two cases: having a *small* learning rate η_s or a *large* learning rate η_b, the range of both is to be detailed later. A simple intuition was suggested by Nakkiran (2020) on a two-dimensional toy problem, showing that large learning rates may be beneficial as soon as there is a mismatch between F and R (meaning, what we train on does not correspond to what we test on). We show that such an insight can be extended beyond toy problems to realistic scenarios with traditional kernel methods, and that, perhaps surprisingly, this phenomenon occurs already in simple classification tasks.

Theorem 1 (Informal version of our main result) *Under a few assumptions described later in this paper, consider the target accuracy α and large and small step sizes θ_b and θ_s (these quantities being defined in the aforementioned assumptions). Consider the gradient descent iterations $\theta_{t+1} = \theta_t - \eta\mathsf{T}(\theta_t - \theta^\star)$ either with step size $\eta = \eta_b$ or $\eta = \eta_s$, and stop the procedure as soon as $F(\theta_{t+1}) \leq \alpha$, resulting in two estimators θ_b or θ_s. Then,*

$$R(\theta_b) - R(\nu^\star) \;\leq\; 34 \frac{\kappa_{\mathsf{U}}}{\kappa_{\mathsf{T}}} (R(\theta_s) - R(\nu^\star)), \tag{2}$$

where κ_{U} and κ_{T} are the condition numbers of the operators U and T, respectively, restricted to $\mathcal{H}_n$.

Note that $R(\nu^\star) = 0$ by definition of R; we made this quantity explicit in the bound for clarity purposes. The main conclusion from the theorem is that with early stopping (and with a target accuracy that is reasonable according to statistical learning theory, as discussed later), large learning rates can provide better estimators than small ones, even though the quantity η_s may yield much faster convergence for minimizing the objective function F than η_b (a fact also discussed later). This phenomenon occurs when the condition number κ_{T} is much larger than κ_{U}, which may already arise in classification tasks, as mentioned earlier in Example 2 where $\kappa_{\mathsf{U}} = 1$.

Note that Eq. (2) raises several questions and could be easily misinterpreted, since such a relation may suggest that an arbitrarily small risk $R(\theta_b)$ could be obtained by considering minimization problems that are arbitrarily badly conditioned. Unfortunately, but not surprisingly, there is however no free lunch here, as discussed in the next remark.

Remark 2 (The issue of ill-conditioning.) *A naive observation is that $R(\theta_b)$ could be arbitrarily small by making the problem more ill-conditioned. However, the bound on $R(\theta_b)$ in Eq. (2) is relative to $R(\theta_s)$. Notably, a careful reading of the proof shows that $R(\theta_s)$ is an increasing function of the conditioning number, for the chosen level sets α.*

Summary of contributions. Our first contribution is the relation described in Eq. (2), highlighting potential benefits of large learning rate strategies when the training objective has a worse condition number than the one used to evaluate the quality of the estimator. This is illustrated in Fig. 1, a figure inspired by Nakkiran (2020). Our second contribution is to show that such a mismatch systematically occurs in simple classification scenarios with low noise, where the quantity of interest to minimize may not be the population risk, as discussed earlier. Overall, this allows us to show that the previous phenomenon occurs in realistic learning scenarios with kernels, which we also check in practice through numerical experiments.

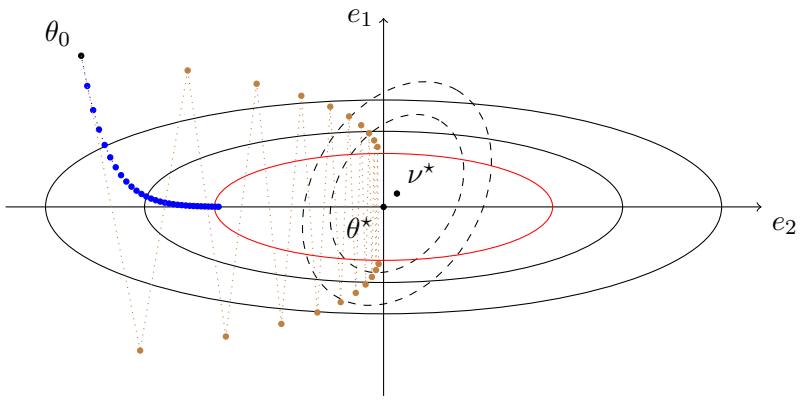


Figure 1: We optimize the quadratic F *(level sets are filled lines, centered in $\theta^\star$)* with gradient descent, starting from θ_0 until we reach the level sets α *(filled line, red)*. However, we evaluate the quality of the estimator through R *(level sets are dashed lines, centered in $\nu^\star$)*. Doing small step size *(blue dots)* optimizes the direction e_1 first, and yields an estimate which is far from $\nu^\star$ in U norm; doing big step size *(brown dots)* oscillates in the direction e_1, but ultimately yields an estimator which is close to $\nu^\star$ in U norm.

2. Related Work

Our main motivation is to better understand the role of learning rate in obtaining good generalization for supervised learning. Even though empirical benefits of large learning rates were first described for neural networks, a few recent works have studied this phenomenon for convex problems. We review here some relevant work.

Setting the learning rate in neural networks. Stochastic gradient descent has become the standard tool for optimizing neural networks. When the learning rate is very small, the network evolves in a so-called "lazy-regime" where its dynamics are well understood (Chizat et al., 2019; Jacot et al., 2018) but which fails to capture the good generalization performance observed with large learning rate. Specifically, this phenomenon has been empirically observed numerous times (see, for instance Smith et al., 2021; Jastrzebski et al., 2021, 2020); common strategies consist of using first a large learning rate, before annealing it to a smaller value. As a first step towards proving theoretically the effect of choosing large learning rates for training neural networks, Li et al. (2019) devise a two-layer neural network model with different set of features where the order in which they are learnt matters, where the previous annealing strategy could be shown to be useful in theory.

A convex perspective. Recently, different papers tried to reproduce this phenomenon in convex settings. This is probably thanks to the observation made by Nakkiran (2020), where a toy dataset is exhibited, which was the main motivation for this work. However, it fails to capture realistic scenarios where the data distribution is not isotropic, or with non linear data embeddings. Wu et al. (2021) aimed at filling these gaps but again, relies on the data distribution to be linear, isotropic, with the number of dimension going to infinity in order to have all data points approximately orthogonal; we do not make any of those assumptions. The bound with the condition number we obtain in Theorem 5 is totally new. Finally, we highlight that we use *plain* gradient descent, and do not need stochasticity to exhibit the big learning rate phenomenon. This is consistent with recent work (Geiping et al., 2021) which shows that SGD is not necessary to obtain state of the art performances, and that GD simply needs a better fine tuning of hyper parameters.

3. Main Result

In this section, we show that by performing standard gradient descent on the empirical loss F, choosing a big learning rate will first optimize the smallest eigencomponent of T. That is, the resulting estimator is mostly located on the biggest eigenvector of T. On the other hand, the smaller the learning rate, the more will the solution be located on the small eigencomponents, with biggest eigenvectors of T being learnt first.

3.1. Settings and notations

Gradient descent updates. We perform standard gradient descent on the empirical loss F, starting from some $\theta_0 \in \mathcal{H}$, with step size η. We obtain

$$\forall t \geq 0, \;\; \theta_{t+1} = \theta_t - \eta \mathsf{T}(\theta_t - \theta^\star), \;\; \text{thus} \;\; \theta_t - \theta^\star = (\mathbf{I} - \eta\mathsf{T})^t(\theta_0 - \theta^\star). \tag{3}$$

This enables a very simple analysis of the training in the eigenbasis of T. We now give a more precise definition of the model in Eq. (1).

Assumption 1 (Representer theorem assumption) *There is a n-dimensional subspace $\mathcal{H}_n \subseteq \mathcal{H}$ that is invariant by T—that is, $T\theta$ is in $\mathcal{H}_n$ for all θ in $\mathcal{H}_n$ and such that $\nu^\star$ is in $\mathcal{H}_n$.*

We denote by (σ_i, e_i) the eigenbasis of the p.d. operator T restricted to $\mathcal{H}_n$, with $\sigma_1 > \cdots > \sigma_n > 0$, assuming eigenvalues are distinct from each other, and we call $\kappa_{\mathsf{T}} = \sigma_1/\sigma_n$ the condition number. Similarly, the restriction of U to $\mathcal{H}_n$ is a positive definite operator whose spectrum is $\varsigma_1 > \cdots > \varsigma_n$, with condition number $\kappa_{\mathsf{U}} = \varsigma_1/\varsigma_n$. Since, rescaling the objectives F and R by constant factors does not change their minimizers, we also safely assume that $\sigma_1 = \varsigma_1 = 1$.

The model described by Assumption 1 is quite natural, and ensures that a representer theorem holds when learning on a finite training set of n points. It is notably satisfied in classical learning formulations with kernels.

With the notations of Assumption 1, we can now rewrite the update of Eq. (3) along a specific direction e_i:

$$\forall t \geq 0, \quad \langle \theta_t - \theta^\star, e_i \rangle_{\mathcal{H}} = (1 - \eta\sigma_i)^t \langle \theta_0 - \theta^\star, e_i \rangle_{\mathcal{H}} \,. \tag{4}$$

Consider the quantity $|1 - \eta\sigma_i|$. The closer to 0, the smaller will the i-th component of $\theta_t - \theta^\star$ on the eigenbasis be when the number of steps t increases. We plot $|1 - \eta\sigma_i|$ in Fig. 2.

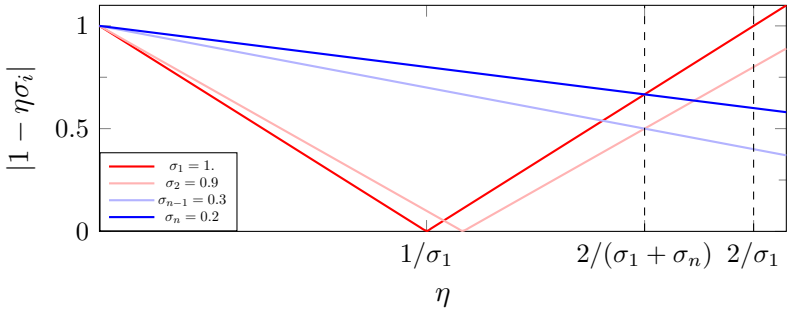


Figure 2: Attenuation coefficient $|1 - \eta\sigma_i|$ function of the step size η, for 4 eigenvalues. For $1 \leq i \leq n$, the attenuation is the quantity by which decays the projection of $\theta - \theta^\star$ on e_i at each step. The closer to 0, the faster will the direction e_i of T be learnt. In our analysis, the learning rate must satisfy $\eta_s < 2/(\sigma_1 + \sigma_n) < \eta_b < 2/\sigma_1$.

Specifically, two ranges of learning rate naturally appear. With a *small* learning rate satisfying $\eta_s < 2/(\sigma_1 + \sigma_n)$, we see on Fig. 2 that the attenuation $|1 - \eta\sigma_i|$ is biggest for the smallest eigenvalue. On the other hand, with a *big* learning rate satisfying $\eta_b > 2/(\sigma_1 + \sigma_n)$, we see that the attenuation is biggest for the biggest eigenvalue. This motivates the next assumption.

Assumption 2 (Learning rate) *The learning rates satisfy*

$$0 < \eta_s < \frac{2}{\sigma_1 + \sigma_n} < \eta_b < \frac{2}{\sigma_1}. \tag{5}$$

Note that the quantity $\eta = \frac{2}{\sigma_1 + \sigma_n}$ which naturally appears in Fig. 2 is a classical upper bound for proving the convergence of σ_1-smooth and σ_n-strongly convex function, of which F belongs to, see *e.g.* Thm 2.1.15 in Nesterov (2018). The rate $1/\sigma_1$ is the classical one when we do not have a strong convexity assumption. This means that in our model, the concept of "small" learning rate simply

means being of the order of the best possible learning rates available from an optimization point of view, while the concept of "large" means being close to values leading to diverging algorithms.

Remark 3 (Biggest learning rate before divergence.) *In a recent work, Cohen et al. (2021) observed that neural networks trained with gradient descent and "good" constant step size η were often in a regime where σ_1 – the maximum value of the Hessian of the loss – hovered just around $2/\eta$. It is surprisingly analogous to our model: the range of learning rate we consider for big step sizes in Assumption 2 enforces $2/\eta_b$ to be close to σ_1.*

Remark 4 (Oscillating weights.) *Geometrically, having $\eta > 1/\sigma_1$ means that the estimator will oscillate along the direction e_1, i.e. $\langle \theta_t - \theta^\star, e_1 \rangle$ will change sign at each iteration. Such behaviour was observed for neural networks trained with classical learning rate strategies, where the weights' sign change in the early phase of training (Xing et al., 2018).*

Then, the following technical assumption is needed to ensure that there is a signal on the lowest and biggest eigendirection. It is satisfied *e.g.* as soon as the initialization is chosen at random.

Assumption 3 (Initialization) *We assume*

$$\langle \theta_0 - \theta^\star, e_1 \rangle_{\mathcal{H}} \neq 0, \quad \langle \theta_0 - \theta^\star, e_n \rangle_{\mathcal{H}} \neq 0.$$

Finally, we assume that the target accuracy in terms of optimization is not too small compared to the model error $R(\nu^\star)$:

Assumption 4 (Target accuracy and model error) *Consider some learning rates η_b and η_s chosen in the range of Assumption 2. We assume that the target accuracy α satisfies $\alpha \leq \alpha_1$, where α_1 is given in Definition 10 and only depends on the spectrum of T and the learning rates. Furthermore, we assume that*

$$\frac{R(\theta^\star)}{\alpha} \leq \min\left\{\frac{1}{4}, \frac{\kappa_{\mathsf{T}}}{72\kappa_{\mathsf{U}}}\right\}. \tag{6}$$

This assumption is twofold, providing on the target accuracy α both an *upper* bound (with $\alpha \leq \alpha_1$) and a *lower* bound (with Eq. (6)). The *lower bound* not being satisfied amounts to F being a poor approximation of R: $R(\theta^\star)$ is too big, or the two ellipsoids' centers are far apart in Fig. 1. Usual machine learning settings often involve large amounts of data, where the limiting factor for good generalization is poor optimization rather than lacking information, so we can expect this assumption to often hold in practice. The *upper bound* stems from the proof technique of our main result in Theorem 5, a brief sketch of which is available in Section 3.4. The proof relies on the fact that sufficiently many steps t_s (resp. t_b) are made before the gradient descent is stopped, so that the biggest (resp. the smallest) eigen component is attenuated enough. To ensure this, we can *(i)* either make the learning rate smaller[1], or *(ii)* take the target error α sufficiently small. We choose the latter, hence the assumption $\alpha \leq \alpha_1$.

1. Informally, by making $\eta_b \to 2/\sigma_1$ (resp. $\eta_s \to 0$) we need bigger t_b (resp. t_s) to achieve optimization error α. Thus, $\alpha < \alpha_1$ can be replaced with $\eta_b > 2/\sigma_1 - \epsilon$ (resp. $\eta_s < \epsilon$).

3.2. The Main Theorem

We now give our main theorem, whose proof is given in Appendix A.

Theorem 5 (Benefits of large learning rates) *Consider the different quantities defined in Assumptions 1, 2, 3 and 4. Then, perform the gradient descent updates of Eq. (3), with either small step size η_s or big step size η_b, and stop as soon as $F(\theta_t) \leq \alpha$, assuming that the resulting estimators satisfy $F(\theta_s) \geq \alpha/2$ and $F(\theta_b) \geq \alpha/2$[2]. Then,*

$$R(\theta_b) - R(\nu^\star) \;\leq\; 34\frac{\kappa_{\mathsf{U}}}{\kappa_{\mathsf{T}}}(R(\theta_s) - R(\nu^\star)). \tag{7}$$

Recall that R has minimum 0, so Eq. (7) essentially guarantees better performance of θ_b. The estimator obtained with big step size is better that the one obtained with small step size as soon as the operator T is ill-conditioned. This is notably the case when doing classification with kernel methods with the ridge estimator, as we discuss in Section 4.

3.3. Discussion

Implications in classification with separable classes. In the context of the example discussed in the introduction, we see clearly that, under the simplifying hypothesis that the population error behaves similarly to the empirical error, the choice of the learning step has the unexpected impact of reducing the Hilbert norm by a multiplicative constant that can be significantly smaller than 1, leading to an exponential improvement in the classification error.

Comparison with analysis techniques based on learning rate annealing. Most recent approaches to explain the role of the learning rate in the generalization (Li et al., 2019; Wu et al., 2021) rely on *annealing* the learning rate: the first phase of the training is carried out with a large step size, before it is discounted to a lower value. We do not need such mechanism in our theoretical analysis, which turns to be simpler with a unique value for the step size. With annealing, our analysis could sum up the following way: do t steps with learning rate greater or equal to $2/\sigma_1$, so that all attenuation coefficient $|1 - \eta_b\sigma_i|$ in Eq. (4) are smaller than 1 except for the first (few) eigencomponent. Doing so, all eigendirections would be optimized except for the first (few) ones. Then, anneal the learning rate until the α level set of the loss are reached.

Discussion on the complexity. The fact that the result of Theorem 5 relies on the condition number can be somewhat surprising. Indeed, we may wonder why the other eigenvalues of the spectrum do not play a role in the result. This is in fact due to the proof technique, which relies on comparing the estimator mostly located on e_1 (for big learning rates) and the one mostly on e_n (for the small learning rates). Thus, the distance $\sigma_{n-1} - \sigma_n$ and $\sigma_1 - \sigma_2$ play a role in the *complexity* of the gradient descent, which is highlighted by the next lemma, which is a consequence of Lemmas 11 and 12 in Appendix A.

Lemma 6 (Computational complexity) *Under the settings of Theorem 5, denote t_s (resp. t_b) the number of steps necessary to obtain θ_s (resp. θ_b). Then,*

$$t_s \geq O\left(\log \frac{1 - \eta_s\sigma_n}{1 - \eta_s\sigma_{n-1}}\right), \quad t_b \geq O\left(\log \frac{|\eta_b\sigma_1 - 1|}{\max\{\eta_b\sigma_2 - 1, 1 - \eta_b\sigma_n\}}\right). \tag{8}$$

2. This is a mild assumption that could be removed at the price of cumbersome technical details.

In particular, if $s = \sigma_{n-1} - \sigma_n$, then $t_s = O(1/s)$, and the same holds for t_b with $s = \sigma_1 - \sigma_2$.

We emphasize that the complexity incurred by using learning rates satisfying Assumption 2 may be very large. However, the motivation of our work is to study an existing practice in deep learning (using large step sizes that are not optimal in terms of optimization of the training loss, but which are better in terms of test loss). We study this phenomenon from a kernel perspective to be able to leverage theoretical tools. Analysing the computational complexity to obtain a given statistical accuracy in convex problems is a separate and well-documented problem, and we do not necessarily advocate the use of very large step sizes for kernel regression.

Nyström projections. In practice projections techniques (known as Nyström projections) are used to reduce the dimension and avoid a cost quadratic in n for gradient descent. Given some $m \ll n$, this amounts to choosing m anchor points among the data and approximating the kernel matrix K with a rank-m matrix $\tilde{K} = K_{nm} K_m^{-1} K_{nm}^\top$. Fortunately, this is still encompassed in the framework of the model in Eq. (1). Indeed, the resulting problem is still a quadratic problem, for which a representer theorem holds – the solution lies on the span of the anchor points.

Beyond the square loss. The result of Theorem 5 relies on using the square loss for obtaining a closed-form expression of the gradient descent update. This is fairly common in learning theory and already provides interesting hindsight. A natural extension to other loss functions would be to consider local quadratic approximations using the Hessian. This is for instance what has been done for kernel ridge regression and self-concordant loss functions by Bach (2010).

3.4. Sketch of proof for the main result

The detailed proof is delayed to Appendix A. The idea is the following: by tuning the number of steps, we can have the estimator trained with small (resp. big) step size mostly aligned with the smallest (resp. biggest) eigenvector.

The directional bias induced by the step size. As shown in Fig. 2, having the learning rates satisfying Assumption 2 ensures that the quantities

$$\epsilon_b^2 = \frac{\sum_{2 \leq i \leq n} \langle \theta_b - \theta^\star, e_i \rangle^2}{\langle \theta_b - \theta^\star, e_1 \rangle^2}, \quad \epsilon_s^2 = \frac{\sum_{1 \leq i \leq n-1} \langle \theta_s - \theta^\star, e_i \rangle^2}{\langle \theta_s - \theta^\star, e_n \rangle^2}, \tag{9}$$

can be made arbitrarily small, while Assumption 3 ensures they are well defined. ϵ_b (resp. ϵ_s) quantifies to what extent is $\theta_b - \theta^\star$ (resp. $\theta_s - \theta^\star$) mostly on e_1 (resp. e_n). For instance, in the extreme case where $\epsilon_b = 0$ (resp. $\epsilon_s = 0$), then $\theta_b - \theta^\star = x e_1, x \in \mathbb{R}$ (resp. $\theta_s - \theta^\star = y e_n, y \in \mathbb{R}$). To see why it can be made small, refer to the closed-form expression of $\theta_{(\eta,t)}$ in Eqs. (3) and (4), and assume for simplification that $\langle \theta_0 - \theta^\star, e_i \rangle = c_0$ for all i, with c_0 some constant factor. This gives

$$\epsilon_b^2 = \frac{\sum_{2 \leq i \leq n} (1 - \eta_b \sigma_i)^{2t}}{(1 - \eta_b \sigma_1)^{2t}}, \quad \epsilon_s^2 = \frac{\sum_{1 \leq i \leq n-1} (1 - \eta_s \sigma_i)^{2t}}{(1 - \eta_s \sigma_n)^{2t}}. \tag{10}$$

Following the discussion of Assumption 2, we have that $|1 - \eta_b \sigma_1| > |1 - \eta_b \sigma_i|$ for all $i > 1$, and $|1 - \eta_s \sigma_n| > |1 - \eta_s \sigma_n|$ for all $i < n$. Thus, we have that for any $\delta > 0$,

$$t_b \geq \frac{1}{2} \frac{\log 1/\delta^2}{\log \frac{\eta_b \sigma_1 - 1}{\max_{2 \leq i \leq n} |1 - \eta_b \sigma_i|}} \implies \epsilon_b^2 \leq \delta^2, \quad t_s \geq \frac{1}{2} \frac{\log 1/\delta^2}{\log \frac{1 - \eta_s \sigma_n}{\max_{1 \leq i \leq n-1} |1 - \eta_s \sigma_i|}} \implies \epsilon_s^2 \leq \delta^2. \tag{11}$$

Different risk on the α-level sets. In this paragraph, assume *(A)* $\theta_b = xe_1$, and $\theta_s = ye_n$, that is $\epsilon_b = \epsilon_s = 0$, *(B)* that $R(\theta^\star) = 0$ and *(C)* that

$$\alpha/2 \leq F(\theta_b) \leq \alpha, \;\; \alpha/2 \leq F(\theta_s) \leq \alpha. \tag{12}$$

Then, we have for θ_b that

$$R(\theta_b) = \frac{1}{2}\left\|xe_1\right\|_{\mathsf{U}}^2 \leq \frac{1}{2}\varsigma_1 x^2 \leq \frac{\alpha}{2}\frac{\varsigma_1}{\sigma_1}, \tag{13}$$

where we used Eq. (12) to bound $\alpha \geq F(\theta_b) = 1/2 \cdot \sigma_1 x^2$. We do the same with θ_s, this time using $1/2 \cdot \sigma_n y^2 = F(\theta_s) \leq \alpha$ to obtain

$$R(\theta_s) = \frac{1}{2}\left\|ye_n\right\|_{\mathsf{U}}^2 \geq \frac{1}{2}\varsigma_n y^2 \geq \alpha\frac{\sigma_n}{\varsigma_n}, \tag{14}$$

Finally, combining Eq. (13) with Eq. (14), we obtain

$$R(\theta_b) \leq 2\frac{\kappa_{\mathsf{U}}}{\kappa_{\mathsf{T}}}R(\theta_s).$$

Ensuring both conditions can be met together. We now point out the main differences between this simplified sketch of proof and the rigorous proof in Appendix A. First of all, we do not have *(A)* but rather an approximation of it, with $\epsilon_s \leq \delta$ and $\epsilon_b \leq \delta$. Second, we do not have *(B)* and rather take into account the error $R(\theta^\star)$ to derive Theorem 5. Finally, and most importantly, we check that we can have *low ϵ and $F(\theta) \geq \alpha/2$ at the same time*. Indeed, we need a big number of iterations to achieve low ϵ_b or ϵ_s. This implies better optimization of the objective function F. To prevent this, we can either tune the learning rate (having η_s close to 0 and η_b close to $2/\sigma_1$) or provide an upper bound on α. We choose the later, hence the hypothesis $\alpha \leq \alpha_1$ in Assumption 4.

4. Comparison with Results in Kernel Regression

To provide some intuition over the result of Theorem 5, we consider its implications in a supervised learning setting, specifically classification on a low-noise dataset.

4.1. Background on the kernel ridge regression estimator

We consider standard settings. We denote by $\mathcal{X} \subseteq \mathbb{R}^d$ the input space, and $\mathcal{Y} = \{-1, 1\}$ the output space. We draw n i.i.d samples $(x_i, y_i)_{1 \leq i \leq n}$ from an unknown distribution ρ on $\mathcal{X} \times \mathcal{Y}$, and we search a prediction function $\theta \in \mathcal{H}$, where $\mathcal{H}$ is a RKHS with feature map $\phi : \mathcal{X} \to \mathcal{H}$. We assume the kernel to be bounded by a constant C_K. See Aronszajn (1950) for a precise account on RKHS. We use the square loss as loss function. In order to find a function θ which maps elements of $\mathcal{X}$ to $\mathcal{Y}$, we optimize the (regularized) *empirical risk* F, defined for all $\theta \in \mathcal{H}$ and $\lambda \geq 0$ a regularization parameter with

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\theta(x_i) - y_i\right)^2 + \frac{\lambda}{2}\left\|\theta\right\|^2. \tag{15}$$

The minimizer $\theta^\star$ of F is always well defined as soon the training samples x_i are distinct (in the case $\lambda = 0$), which we assume now. We will be minimizing F with gradient descent when we are in fact interested in minimizing the *test error*

$$B(\theta) = \mathbb{P}[\mathrm{Sign}[\theta(x) \neq y]]. \tag{16}$$

We will relate the test error and the empirical risk to quadratics forms in $\mathcal{H}$ by means of other quantities. To do that, we first define the population loss along with its regularized version with

$$\forall \theta, \;\; \mathcal{P}(\theta) = \int_{\mathcal{X}\times\mathcal{Y}} \frac{1}{2}\left(\theta(x) - y\right)^2 \mathrm{d}\rho(x,y), \;\; \mathcal{P}_\lambda(\theta) = \mathcal{P}(\theta) + \frac{\lambda}{2}\left\|\theta\right\|^2. \tag{17}$$

The minimizer of $\mathcal{P}$ on $\mathcal{L}_2(\rho_x)$ is the regression function $g^\star(x) = \mathbb{E}[y|x]$. It is an element of $\mathcal{L}_2(\rho_x)$ but not necessary of $\mathcal{H}$. We denote by $\nu^\star$ the minimizer of $\mathcal{P}_\lambda$ on $\mathcal{H}$. If $\lambda > 0$, it is always well defined; otherwise, with $\mathcal{I} : \mathcal{H} \to \mathcal{L}_2(\rho_x)$ the inclusion operator, $\nu^\star$ exists as soon as the projection of the regression function on the closure of the range of $\mathcal{I}$ belongs to the range of $\mathcal{I}$. See Vito et al. (2005) for a precise account.

In the following, we assume $\lambda \geq 0$ and that $\nu^\star$ is well defined, and we consider specific assumptions on ρ, *via* assumptions on $\nu^\star$ and $g^\star$.

4.2. Relating supervised learning with quadratic forms in $\mathcal{H}$

To relate the problems of Eqs. (15) and (16) to quadratic forms in the RKHS, we simply need to introduce the *empirical covariance operator*, with

$$\mathsf{T} = \frac{1}{n}\sum_{i=1}^n \phi(x_i) \otimes \phi(x_i). \tag{18}$$

Then, as optimizing the Hilbert norm is a good proxy for optimizing the test error (following Example 2 and Lemma 14 in Appendix B), we define

$$F\left(\theta\right) = \frac{1}{2}\left\|\theta - \theta^\star\right\|_{\mathsf{T}}^2 + \mathsf{cst}, \;\; R(\theta) = \frac{1}{2}\left\|\theta - \nu^\star\right\|_{\mathcal{H}}^2, \;\; \text{with } \mathsf{cst} = \frac{1}{2n} y^\top \left[\mathbf{I}_n - \frac{K}{n}\left(\frac{K}{n} + \lambda\right)^{-1}\right] y. \tag{19}$$

K is the kernel matrix (K/n shares the same spectrum than T), and the minimum $F(\theta^\star) = \mathsf{cst}$ is 0 when $\lambda = 0$. We are in the settings of the model of Eq. (1), and we can readily apply Theorem 5 with U being the identity operator $\mathbf{I}_{\mathcal{H}}$.

Corollary 7 (Benefit of big step size for classification task.) *Under the settings of Theorem 5 with the additional assumption that $\mathsf{U} = \mathbf{I}_{\mathcal{H}_n}$, we have*

$$R(\theta_b) - R(\nu^\star) \;\leq\; \frac{34}{\kappa_{\mathsf{T}}}(R(\theta_s) - R(\nu^\star)).$$

5. Experiments

We evaluate the claims of Section 4 on CKN-MNIST, a dataset consisting of the MNIST dataset embedded by a convolutional kernel network (Mairal, 2016). It allows for a realistic use-case, with classification accuracy close to 99%, by necessitating a reasonable number of samples $n = 1000$. On CKN-MNIST, we achieve 98.5% test accuracy with the Gaussian kernel with scale parameter 30 and no regularization. Adding regularization only improves the test accuracy by 0.04%.

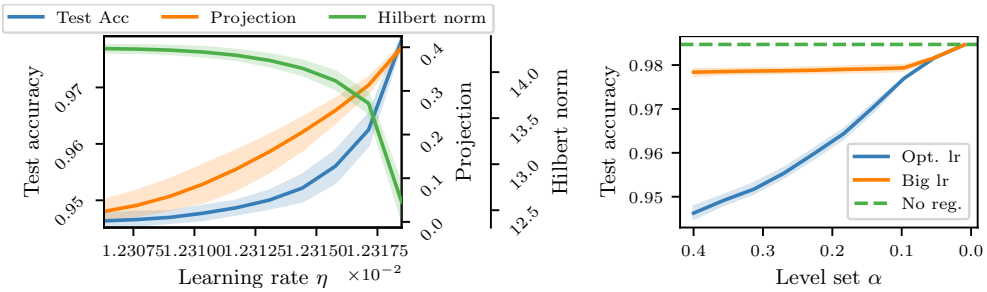
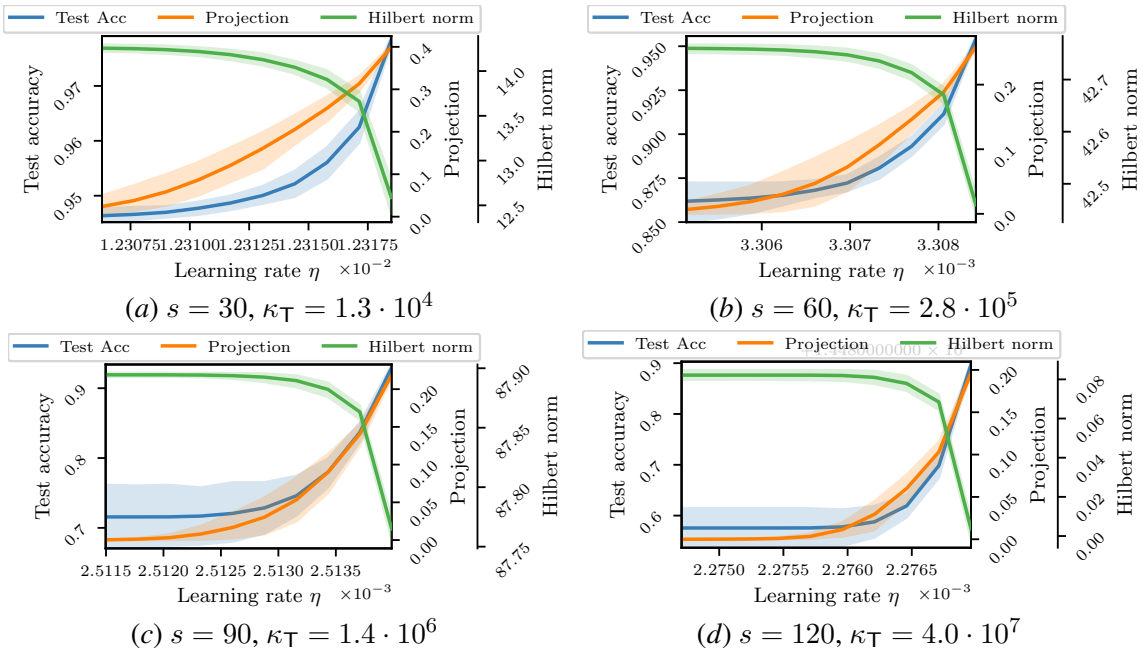


Figure 3: *(Left)* Test accuracy *(blue)* function of step size η for CKN-MNIST. As the learning rate increases, the projection on the first component *(orange)* increases, which makes the Hilbert norm *(green)* decreases. This results in predictions closest in $\mathcal{L}_\infty$ norm to the optimum.
(Right) Test accuracy function of level set α for CKN-MNIST. As we optimize more, the better performances of big step size *(orange)* compare to optimal step size *(blue)* vanish to reach the prediction of the optimum of F *(green, dashed)*.
Shaded areas show standard deviation (train set and initialization) over 10 runs.



(*a*) $s = 30$, $\kappa_{\mathsf{T}} = 1.3 \cdot 10^4$ (*b*) $s = 60$, $\kappa_{\mathsf{T}} = 2.8 \cdot 10^5$

(*c*) $s = 90$, $\kappa_{\mathsf{T}} = 1.4 \cdot 10^6$ (*d*) $s = 120$, $\kappa_{\mathsf{T}} = 4.0 \cdot 10^7$

Figure 4: Theorem 5 predicts that the improvement in taking big rather than small step size increases with the condition number of T. We test this claim on our dataset: larger kernel scale s makes the condition number of the kernel matrix κ_{T} increases, which results in a larger margin between the test accuracy of θ_b and the one of θ_s.

Test accuracy function of the step size. We define some final level set α. We plot three quantities function of the learning rate η. The *projection on the first component* $\langle \theta - \theta^\star, e_1 \rangle_{\mathcal{H}}$ must be small for moderate learning rate and big for learning rate close to $2/\sigma_1$, as the attenuation satisfies $|1 - \eta\sigma_1| \to 1$ when $\eta \to 2/\sigma_1$, see Fig. 2. This makes the *Hilbert norm* $R(\theta(\eta))$ decreases as η increases, as predicted by Corollary 7. This results in better test accuracy $1 - B$ for big learning rate, following Lemma 14. This is summed up in Fig. 3, left. Note that the range for big step size is narrow. This is consistent with Assumption 2, which requires a range equal to $2\sigma_n/(\sigma_1 + \sigma_n) \approx 2\kappa_T^{-1}$. This is also in line with practices in deep learning where the best performance in generalization is often obtained by choosing the largest possible learning rate such that the model does not diverge.

Test accuracy function of optimization. Our main bound in Theorem 5 relies on Assumption 4: in learning settings, it means that the optimization error α must be greater than a constant times the statistical error $R(\theta^\star)$ in order to observe improvements with big step sizes. This is shown in Fig. 3, right. Additional details on the plot is available in Appendix E.

Scale of the kernel. In Fig. 3, the scale of the Gaussian kernel is set to $s = 30$, with which we obtained the best results on the test set. It is worth noting though that the scale of the kernel directly impacts the conditioning of the matrix. Notably, when $s \to 0$, the kernel matrix K tends to the identity (hence $\kappa_{\mathsf{T}} \to 1$) while when $s \to \infty$, K tends to a rank-1 operator (hence $\kappa_{\mathsf{T}} \to \infty$). A core result of Theorem 5 is that we have *bigger improvement for bigger* κ_{T}. We reproduce the experiment on the test accuracy with different scale in Fig. 4, and notice indeed bigger improvements for larger scale s, hence worse conditioning.

6. Conclusion

A large class of learning problems can be formulated as optimizing a function F with gradient descent while we are interested in optimizing another function R. Using simple quadratic forms to model this mismatch is a natural thing to do, while already providing lots of insight. We indeed show that the choice of large step sizes that may be suboptimal from an optimization point of view may provide better estimators than small/medium step sizes. In particular, we show that this phenomenon occurs in realistic classification tasks with low noise when learning with kernel methods. In future work, we are planning to study other variants of gradient-based algorithms, which may be stochastic, or accelerated, and perhaps exploit the insight developed in our work to design new algorithms, which would focus on statistical efficiency, exploiting prior knowledge on the test loss, rather than on optimization of the training objective.

Acknowledgments

A.R. acknowleges support of the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). A.R. acknowledges support of the European Research Council (grant REAL 947908). J. Mairal was supported by the ERC grant number 714381 (SOLARIS project) and by ANR 3IA MIAI@Grenoble Alpes, (ANR19-P3IA-0003).

References

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

J-Y Audibert. Classification under polynomial entropy and margin assumptions and randomized estimators. 2004.

Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 35(2):608–633, 2007.

F. Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384 – 414, 2010. doi: 10.1214/09-EJS521. URL https://doi.org/10.1214/09-EJS521.

Frank Bauer, Sergei Pereverzev, and Lorenzo Rosasco. On regularization algorithms in learning theory. *Journal of Complexity*, 23(1):52–72, 2007.

Gilles Blanchard and Nicole Mücke. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 18:971–1013, 2016.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. *Advances in Neural Information Processing Systems (NIPS)*, 2007.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations (ICLR)*, 2021.

Jonas Geiping, Micah Goldblum, Phillip E. Pope, Michael Moeller, and Tom Goldstein. Stochastic training is not necessary for generalization. *arXiv preprint arXiv:2109.14119*, 2021.

L. Lo Gerfo, L. Rosasco, F. Odone, E. De Vito, and A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic Fisher explosion: Early phase Fisher matrix impacts generalization. In *International Conference on Machine Learning (ICML)*, 2021.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Julien Mairal. End-to-End Kernel Learning with Supervised Convolutional Kernel Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Preetum Nakkiran. Learning rate annealing can provably help generalization, even for convex problems. *arXiv preprint arXiv:2005.07360*, 2020.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. *Lectures on Convex Optimization*. Springer, 2018.

Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical Optimality of Stochastic Gradient Descent on Hard Learning Problems through Multiple Passes. In *Neural Information Processing Systems (NeurIPS)*, 2018a.

Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Exponential convergence of testing error for stochastic gradient methods. In *Conference On Learning Theory (COLT)*, 2018b.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations (ICLR)*, 2021.

Ernesto De Vito, Lorenzo Rosasco, Andrea Caponnetto, Umberto De Giovannini, and Francesca Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6(30):883–904, 2005.

Jingfeng Wu, Difan Zou, Vladimir Braverman, and Quanquan Gu. Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate. In *International Conference on Learning Representations (ICLR)*, 2021.

Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with SGD. *arXiv preprint arXiv:1802.08770*, 2018.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Overview of the appendix

- In Appendix A, we prove Theorem 5.
- In Appendix B, we give additional technical information on the low-noise classification task.
- We compare the bound in Theorem 5 to existing results in the case of regression task with kernels, in Appendix C. We highlight the difference between gradient descent with big step size and the estimators which write as a spectral filter.
- Appendix D makes a simple remark, highlighting the difference between gradient descent on the train loss and gradient descent on the Hilbert norm in kernel regression.
- Finally, Appendix E gives additional information on the experiments.

Appendix A. Proof of main result

A.1. Definition and assumptions

Recall from the main text Assumptions 1, 2, 3, 4.

We can rewrite precisely the gradient descent update in Eq. (3) with the following definition.

Definition 8 (Notations for the estimator) *For some step size η, a number of steps t and some $\theta_0 \in \mathcal{H}_n$, we denote $\theta^{(\eta,t)}$ the estimator obtained through gradient descent from θ_0. We denote $\left(\mu_i^{(\eta,t)}\right)_{1\leq i\leq n}$ the decomposition of $\theta^{(\eta,t)} - \theta^\star$ on e_i,* i.e.

$$\theta^{(\eta,t)} - \theta^\star = \sum_{i=1}^{n} \mu_i^{(\eta,t)} e_i. \tag{20}$$

Denoting the initialization with $(\iota_i)_{1\leq i\leq n}$,

$$\theta_0 - \theta^\star = \sum_{i=1}^{n} \iota_i e_i, \tag{21}$$

we have that

$$\forall i \in \{1,\dots,n\}\,,\ \ \mu_i^{(\eta,t)} = \iota_i A_i^{(\eta,t)} = \iota_i (1-\eta\sigma_i)^t, \tag{22}$$

where A_i is the attenuation of the i-th eigencomponent at each step.

To lighten the notations, we denote $\theta_s = \theta^{(\eta_s,t_s)}$ the estimator obtained with t_s small *step size η_s, and $\theta_b = \theta^{(\eta_b,t_b)}$ the estimator obtained with t_b* big *step size η_b. Likewise, we use μ_i when it is clear from the context which of θ_s or θ_b we study.*

With these notations and Assumption 2 and 3, note that

$$\forall(\eta,t),\ \ \mu_1^{(\eta_b,t)} \neq 0,\ \ \mu_n^{(\eta_s,t)} \neq 0.$$

Also, we can now introduce the *second biggest attenuation coefficients* in the following definition.

Definition 9 (Second biggest attenuation coefficient.) *We introduce the second biggest attenuation coefficients $\bar{A}_b, \bar{A}_s$ with*

$$\bar{A}_b \overset{\text{def.}}{=} \max\left\{A_2^{(\eta_b)}, A_n^{\eta_b}\right\}, \quad \bar{A}_s \overset{\text{def.}}{=} A_{n-1}^{(\eta_s)}. \tag{23}$$

Referring to Fig. 2, this implies

- For η_b, we have that

$$A_1 > \max\{A_2, A_n\} \overset{\text{def.}}{=} \bar{A}_b \geq A_i, \quad \forall i > 1. \tag{24}$$

  Thus, by tuning the number of steps t, we can make the ratio $(A_i/A_1)^t$ arbitrarily small for any $i > 1$.

- For η_s we have that

$$A_n > A_{n-1} \overset{\text{def.}}{=} \bar{A}_s \geq A_i, \quad \forall i < n. \tag{25}$$

  Again, for a sufficiently large number of steps t, we can make $(A_i/A_n)^t$ arbitrarily small for any $i < n$.

Definition 10 (Upper bound on α.) *Given some small and big learning rate η_s, η_b, we introduce α_1, a technical quantity depending on the spectrum of T and U and the initialization:*

$$\alpha_1 = \frac{1}{2}\sigma_n \iota_n^2 \exp\left(-\frac{\log\left[\|\theta_0 - \theta^\star\|_{\mathcal{H}}^2 \max\{16n\kappa_{\mathsf{U}}, 4\kappa_{\mathsf{T}}\} \cdot \max\left\{\frac{1}{\iota_1^2}, \frac{1}{\iota_n^2}\right\} + \frac{1}{1-\eta_s\sigma_n} + \frac{1}{\eta_b\sigma_1 - 1}\right]}{\min\left\{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}, \log\frac{A_1}{A_b}\right\}}\right) \tag{26}$$

In particular, note that we have

$$\alpha_1 \leq \frac{1}{2}\sigma_1 \iota_1^2 \exp\left(-\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_1^2} 4n\kappa_{\mathsf{U}} + \frac{1}{\eta_b\sigma_1 - 1}\right]}{\log\frac{A_1}{A_b}}\right),$$

$$\alpha_1 \leq \frac{1}{2}\sigma_n \iota_n^2 \exp\left(-\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2} \max\{16n\kappa_{\mathsf{U}}, 4\kappa_{\mathsf{T}}\} + \frac{1}{1-\eta_s\sigma_n}\right]}{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}\right),$$

which will prove useful for the derivation of Lemmas 11 and 12.

A.2. An upper bound for the estimator with big learning rate

In this subsection, we denote $\mu_i^{\eta_b, t_b}$ with μ_i.

Lemma 11 (Estimator with big step size.) *Set $\alpha > 0$ s.t.*

$$\alpha < \alpha_1, \tag{27}$$

where α_1 is defined in Def. 10. Define the quantity ϵ_b with

$$\epsilon_b^2 = \frac{\sum_{i>1} \mu_i^2}{\mu_1^2}. \tag{28}$$

Then, running gradient descent on F with step size η_b until the $(\alpha/2, \alpha)$ level sets are reached, i.e.

$$\frac{1}{2}\alpha \leq F(\theta_b) \leq \alpha, \tag{29}$$

ensures that

$$\epsilon_b^2 \leq \frac{1}{4n\kappa_{\mathsf{U}}}, \quad \textit{and} \quad \frac{2}{5}\alpha \leq \frac{1}{2}\sigma_1\mu_1^2 \leq \alpha. \tag{30}$$

The resulting estimator is obtained with t_b steps, with

$$t_b \geq \frac{1}{2}\frac{\log \frac{1}{2}\frac{\sigma_1\iota_1^2}{\alpha}}{\log \frac{1}{\eta_b\sigma_1 - 1}} \geq \frac{1}{2}\frac{\log \frac{\|\theta_0 - \theta^\star\|_{\mathcal{H}}^2}{\iota_1^2} 4n\kappa_{\mathsf{U}}}{\log \frac{A_1}{\bar{A}_b}}. \tag{31}$$

Proof

Bound on ϵ_b. By using the definition of ϵ_b and the expression of the $(\mu_i)_{1\leq i\leq n}$ given in Def. 8, we have

$$\epsilon_b^2 = \frac{\sum_{i>1}\mu_i^2}{\mu_1^2} = \frac{\sum_{i>1}\iota_i^2 A_i^{2t}}{\iota_1^2 A_1^{2t}} \leq \frac{\|\theta_0 - \theta^\star\|_{\mathcal{H}}^2}{\iota_1^2}\left(\frac{\bar{A}_b}{A_1}\right)^{2t}. \tag{32}$$

Thanks to the proper choice of η_b given in Asmpt. 2, we have that $\bar{A}_b/A_1 < 1$, so that

$$\forall \delta > 0, \;\; t \geq t_1 \overset{\text{def.}}{=} \frac{1}{2}\frac{\log \frac{\|\theta_0 - \theta^\star\|_{\mathcal{H}}^2}{\delta^2\iota_1^2}}{\log \frac{A_1}{\bar{A}_b}} \implies \epsilon_b^2 \leq \delta^2. \tag{33}$$

Bound on μ_1. Now, recall that the optimization error reads

$$F(\theta_b) = \frac{1}{2}\sum_{i=1}^n \sigma_i\mu_i^2 = \frac{1}{2}\sigma_1\mu_1^2\left(1 + \frac{\sum_{i>1}\sigma_i\mu_i^2}{\sigma_1\mu_1^2}\right)$$

as we assumed that $\iota_1 \neq 0$ in Asmpt. 3. Thus, we can bound the loss in two ways. First, by definition

$$\frac{1}{2}\sigma_1\mu_1^2 \leq F(\theta_b) \leq \frac{1}{2}\sigma_1\mu_1^2(1 + \epsilon_b^2) \tag{34}$$

and second, we assumed the estimator to belong to the $(\alpha/2, \alpha)$ level set of F, *i.e.*

$$\frac{\alpha}{2} \leq F(\theta_b) \leq \alpha. \tag{35}$$

Combining Eq. (34) with Eq. (35), we have that

$$\frac{\alpha}{2(1+\epsilon_b^2)} \leq \frac{1}{2}\sigma_1\mu_1^2 \leq \alpha. \tag{36}$$

Feasability. Finally, we consider if $\epsilon_b^2 \leq \frac{1}{4n\kappa_{\mathsf{U}}}$ and θ_b being in the $(\alpha/2, \alpha)$ level sets can occur at the same time.

First of all, we set the value of δ^2 to $\frac{1}{4n\kappa_{\mathsf{U}}}$ in Eq. (34). We get

$$t_1 \overset{\text{def.}}{=} \frac{1}{2} \frac{\log \frac{\|\theta_0 - \theta^\star\|_{\mathcal{H}}^2}{\iota_1^2} 4n\kappa_{\mathsf{U}}}{\log \frac{A_1}{A_b}}. \tag{37}$$

Then, we derive necessary conditions for having Eq. (35). Those conditions are derived through the bounds established in Eq. (34). For the lower bound, assuming $t \geq t_1$, so that, in particular, we have $\epsilon_b^2 \leq 1/4$,

$$\begin{aligned} \frac{\alpha}{2} \leq F(\theta_b) &\implies \frac{\alpha}{2} \leq \frac{1}{2}\sigma_1 \mu_1^2 (1 + \epsilon_b^2) \\ &\implies \frac{4\alpha}{5\sigma_1} \leq \mu_1^2 = \iota_1^2 (\eta_b \sigma_1 - 1)^{2t} \\ &\implies t \leq t_3 \overset{\text{def.}}{=} \frac{1}{2} \frac{\log \frac{5}{4} \frac{\sigma_1 \iota_1^2}{\alpha}}{\log \frac{1}{\eta_b \sigma_1 - 1}}. \end{aligned}$$

Likewise, for the upper bound we have,

$$\begin{aligned} F(\theta_b) \leq \alpha &\implies \frac{1}{2}\sigma_1 \mu_1^2 \leq \alpha \\ &\implies t \geq t_2 \overset{\text{def.}}{=} \frac{1}{2} \frac{\log \frac{1}{2} \frac{\sigma_1 \iota_1^2}{\alpha}}{\log \frac{1}{\eta_b \sigma_1 - 1}}. \end{aligned}$$

Summing up, we have

- $t \geq t_1$ implies that the bound on ϵ_b in Eq. (33) holds.
- Ensuring that the level set condition in Eq. (35) holds are met implies that $t \in (t_2, t_3)$.

Thus, we need to ensure that $t_2 > t_1$ (and that $t_3 - t_2 > 1$; we assume this, as we can look at smaller level sets if necessary). To do this, we use that t_2 is an decreasing function of α. Thus, $t_2 > t_1$ as soon as

$$\alpha \leq \frac{1}{2}\sigma_1 \iota_1^2 \exp\left(- \frac{\log \left[\frac{\|\theta_0 - \theta^\star\|_{\mathcal{H}}^2}{\iota_1^2} 4n\kappa_{\mathsf{U}} + \frac{1}{\eta_b \sigma_1 - 1} \right]}{\log \frac{A_1}{A_b}} \right),$$

which is exactly the purpose of the technical assumption $\alpha \leq \alpha_1$, with α_1 defined in Def. 10. ■

A.3. A lower bound for the estimator with small learning rate

We now derive a similar result for θ_s. To lighten the notations, we use $\mu_i^{(\eta_s, t_s)} = \mu_i$.

Lemma 12 (Estimator with small step size.) *Set $\alpha > 0$ s.t.*

$$\alpha < \alpha_1, \tag{38}$$

where α_1 is defined in Def. 10. Define the quantity ϵ_s with

$$\epsilon_s^2 = \frac{\sum_{i<n} \mu_i^2}{\mu_n^2}. \tag{39}$$

Then, running gradient descent on F with step size η_s until the $(\alpha/2, \alpha)$ level sets are reached, i.e.

$$\frac{1}{2}\alpha \leq F(\theta_b) \leq \alpha, \tag{40}$$

ensures that

$$\epsilon_s^2 \leq 1/(16n\kappa_{\mathsf{U}}), \quad \textit{and} \quad \frac{2}{5}\alpha \leq \frac{1}{2}\sigma_n\mu_n^2 \leq \alpha. \tag{41}$$

The resulting estimator is obtained with t_s steps, with

$$t_s \geq \frac{1}{2}\frac{\log \frac{1}{2}\frac{\sigma_n \iota_n^2}{\alpha}}{\log \frac{1}{1-\eta_s\sigma_n}} \geq \frac{1}{2}\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\max\{16n\kappa_{\mathsf{U}}, 4\kappa_{\mathsf{T}}\}\right]}{\log \frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}. \tag{42}$$

Proof The proof is very close to the one of Lemma 11. We only give the main results.

Bound on ϵ_s. This quantity can be written

$$\epsilon_s^2 = \frac{\sum_{i<n}\mu_i^2}{\mu_n^2} = \frac{\sum_{i<n}\iota_i^2 A_i^{2t}}{\iota_n^2 A_n^{2t}} \leq \frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\left(\frac{\bar{A}_s}{A_n}\right)^{2t},$$

with $1-\eta_s\sigma_{n-1} = \bar{A}_s > A_n = 1-\eta_s\sigma_n$ following Asmpt.2. Thus,

$$\forall \delta > 0, \;\; t \geq t_1 \overset{\text{def.}}{=} \frac{1}{2}\frac{\log \frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\delta^2\iota_n^2}}{\log \frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}} \implies \epsilon_s^2 \leq \delta^2.$$

Bound on μ_n Again, following the same paragraph in Lemma 11, we have

$$F(\theta_s) = \frac{1}{2}\sum_{i=1}^n \sigma_i\mu_i^2 = \frac{1}{2}\sigma_n\mu_n^2\left(1 + \frac{\sum_{i>1}\sigma_i\mu_i^2}{\sigma_n\mu_n^2}\right)$$

as we assumed that $\iota_n \neq 0$ in Asmpt. 3. We bound the loss in two ways. First, by definition

$$\frac{1}{2}\sigma_n\mu_n^2 \leq F(\theta_s) \leq \frac{1}{2}\sigma_n\mu_n^2(1+\kappa_{\mathsf{T}}\epsilon_s^2) \tag{43}$$

and second, we assumed the estimator to belong to the $(\alpha/2, \alpha)$ level set of F, *i.e.*

$$\frac{\alpha}{2} \leq F(\theta_s) \leq \alpha. \tag{44}$$

Combining Eq. (43) with Eq. (44), we have that

$$\frac{\alpha}{2(1+\kappa_{\mathsf{T}}\epsilon_s^2)} \leq \frac{1}{2}\sigma_n\mu_n^2 \leq \alpha. \tag{45}$$

Feasability. The discussion is the same, but with the values

$$t_1 = \frac{1}{2}\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\max\{16n\kappa_{\mathsf{U}}, 4\kappa_{\mathsf{T}}\}\right]}{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}$$
$$t_2 = \frac{1}{2}\frac{\log\frac{1}{2}\frac{\sigma_n\iota_n^2}{\alpha}}{\log\frac{1}{1-\eta_s\sigma_n}}$$
$$t_3 = \frac{1}{2}\frac{\log\frac{5}{4}\frac{\sigma_n\iota_n^2}{\alpha}}{\log\frac{1}{1-\eta_s\sigma_n}}.$$

Note that the addition of κ_{T} in the definition of t_1 is simply to ensure that

$$\forall t \geq t_1,\ \ \epsilon_s^2 \leq \frac{1}{4\kappa_{\mathsf{T}}},\ \ \text{so that}\ \ \frac{2}{5}\alpha \leq \frac{\alpha}{2(1+\kappa_{\mathsf{T}}\epsilon_s^2)} \leq \frac{1}{2}\sigma_n\mu_n^2.$$

To ensure the feasibility of both bounds at the same time, we need to ensure $t_2 > t_1$. A sufficient condition for this is having

$$\alpha \leq \frac{1}{2}\sigma_n\iota_n^2\exp\left(-\frac{\log\left[\frac{\|\theta_0-\theta^\star\|_{\mathcal{H}}^2}{\iota_n^2}\max\{16n\kappa_{\mathsf{U}}, 4\kappa_{\mathsf{T}}\} + \frac{1}{1-\eta_s\sigma_n}\right]}{\log\frac{1-\eta_s\sigma_n}{1-\eta_s\sigma_{n-1}}}\right)$$

which is, again, covered with the assumption $\alpha \leq \alpha_1$ defined in Def. 10. ■

A.4. Plugging the two together

Theorem 13 *Assume that the optimization error satisfy*

$$\alpha \leq \alpha_1, \tag{46}$$

where α_1 is defined in Definition 10. Assume that Assumption 1 on the operators T and U hold, that the condition on the learning rates η_b, η_s of Assumption 2 hold, as the condition on the initialization in Assumption 3.

Assume that gradient descent is performed until the $(\alpha/2, \alpha)$ level sets are reached. Then we have that

$$R(\theta_b) \leq c_\alpha R(\theta_s),\ \ \textit{with}\ \ c_\alpha = \left[\frac{1+2\frac{\sigma_1}{\varsigma_1}\frac{R(\theta^\star)}{\alpha}}{\left(1-\sqrt{18\frac{\sigma_n}{\varsigma_n}\frac{R(\theta^\star)}{\alpha}}\right)_+}\right]. \tag{47}$$

Further assume that Assumption 4 holds. Then $c_\alpha \leq 2$ and the bound becomes

$$R(\theta_b) \leq 34\frac{\kappa_{\mathsf{U}}}{\kappa_{\mathsf{T}}}R(\theta_s). \tag{48}$$

Proof

Upper bound for big learning rate. We first proceed to bounding $R(\theta_b)$. In this paragraph, we use $\mu_i = \mu_i^{\eta_b, t_b}$. We have

$$
\begin{aligned}
R(\theta_b) &= \frac{1}{2}\left\|\theta_b - \nu^\star\right\|_{\mathsf{U}}^2 && \text{(Definition)} \\
&\leq \left\|\theta_b - \theta^\star\right\|_{\mathsf{U}}^2 + \left\|\theta^\star - \nu^\star\right\|_{\mathsf{U}}^2 && \text{(Triangular inequality)} \\
&= \left\|\theta_b - \theta^\star\right\|_{\mathsf{U}}^2 + 2R(\theta^\star) \\
&\leq \left\|\mu_1 e_1\right\|_{\mathsf{U}}^2 \left(1 + \frac{\left\|\sum_{i>1}\mu_i e_i\right\|_{\mathsf{U}}}{\left\|\mu_1 e_1\right\|_{\mathsf{U}}}\right)^2 + 2R(\theta^\star) && \text{(Triangular inequality)} \\
&\leq 2\left\|\mu_1 e_1\right\|_{\mathsf{U}}^2 \left(1 + \frac{\left\|\sum_{i>1}\mu_i e_i\right\|_{\mathsf{U}}^2}{\left\|\mu_1 e_1\right\|_{\mathsf{U}}^2}\right) + 2R(\theta^\star) && ((a+b)^2 \leq 2(a^2+b^2)) \\
&\leq 2\left\|\mu_1 e_1\right\|_{\mathsf{U}}^2 \left(1 + \kappa_{\mathsf{U}} \frac{\left|\sum_{i>1}\mu_i\right|^2}{\left|\mu_1\right|^2}\right) + 2R(\theta^\star) && \text{(Def. of } \kappa_{\mathsf{U}}, \|e_i\| = 1) \\
&\leq 2\left\|\mu_1 e_1\right\|_{\mathsf{U}}^2 \left(1 + \kappa_{\mathsf{U}} n \epsilon_s^2\right) + 2R(\theta^\star). && \text{(Cauchy-Schwartz)}
\end{aligned}
$$

Now, we use the results of Lemma 11. Firstly, we have $\kappa_{\mathsf{U}} n \epsilon_b^2 \leq 1/4$. Secondly, we can use $\|e_1\|_{\mathsf{U}}^2 \leq \varsigma_1$. The previous inequality then turns to

$$
R(\theta_b) \leq \frac{5\varsigma_1}{2}\mu_1^2 + 2R(\theta^\star).
$$

Finally, we use the fact that $\frac{1}{2}\sigma_1\mu_1^2 \leq \alpha$ to conclude with

$$
R(\theta_b) \leq 5\alpha\frac{\varsigma_1}{\sigma_1} + 2R(\theta^\star). \tag{49}
$$

Lower bound for small learning rate. We now turn to bounding $R(\theta_s)$. Here, we use $\mu_i = \mu_i^{\eta_s, t_s}$. We have

$$
\begin{aligned}
R(\theta_s) &= \frac{1}{2}\|\theta_s - \nu^\star\|_{\mathsf{U}}^2 && \text{(Definition)} \\
&\geq \frac{1}{2}\left(\|\theta_s - \theta^\star\|_{\mathsf{U}} - \|\theta^\star - \nu^\star\|_{\mathsf{U}}\right)^2 && \text{(Triangular inequality)} \\
&\geq \frac{1}{2}\left(\|\mu_n e_n\|_{\mathsf{U}} - \left\|\sum_{i<n}\mu_i e_i\right\|_{\mathsf{U}} - \|\theta^\star - \nu^\star\|_{\mathsf{U}}\right)^2 && \text{(Idem)} \\
&= \frac{1}{2}\|\mu_n e_n\|_{\mathsf{U}}^2\left(1 - \frac{\left\|\sum_{i<n}\mu_i e_i\right\|_{\mathsf{U}}}{\|\mu_n e_n\|_{\mathsf{U}}} - \sqrt{\frac{2R(\theta^\star)}{\|\mu_n e_n\|_{\mathsf{U}}^2}}\right)^2 \\
&\geq \frac{1}{2}\|\mu_n e_n\|_{\mathsf{U}}^2\left(1 - 2\frac{\left\|\sum_{i<n}\mu_i e_i\right\|_{\mathsf{U}}}{\|\mu_n e_n\|_{\mathsf{U}}} - \sqrt{\frac{8R(\theta^\star)}{\|\mu_n e_n\|_{\mathsf{U}}^2}}\right) && \text{(As } (1-x)^2 \geq 1-2x\text{)} \\
&\geq \frac{1}{2}\varsigma_n\mu_n^2\left(1 - 2\sqrt{\kappa_{\mathsf{U}}}\frac{\left|\sum_{i<n}\mu_i\right|}{|\mu_n|} - \sqrt{\frac{8R(\theta^\star)}{\varsigma_n\mu_n^2}}\right) && \text{(Def. of } \kappa_{\mathsf{U}}\text{, with } \|e_i\| = 1\text{)} \\
&\geq \frac{1}{2}\varsigma_n\mu_n^2\left(1 - 2\sqrt{\kappa_{\mathsf{U}}}\left(\frac{n\sum_{i<n}\mu_i^2}{\mu_n^2}\right)^{1/2} - \sqrt{\frac{8R(\theta^\star)}{\varsigma_n\mu_n^2}}\right) && \text{(Cauchy-Schwartz)} \\
&\geq \frac{1}{2}\varsigma_n\mu_n^2\left(1 - 2\sqrt{\kappa_{\mathsf{U}} n}\epsilon_s - \sqrt{\frac{8R(\theta^\star)}{\varsigma_n\mu_n^2}}\right) && \text{(Def. of } \epsilon_s\text{)}
\end{aligned}
$$

We then use Lemma 12. Firstly, we can use $\epsilon_s^2 \leq 1/(16n\kappa_{\mathsf{U}})$ so that $2\sqrt{\kappa_{\mathsf{U}} n}\epsilon_s \leq 1/4$. This gives

$$
R(\theta_s) \geq \frac{3}{8}\varsigma_n\mu_n^2\left(1 - \frac{4}{3}\sqrt{\frac{8R(\theta^\star)}{\varsigma_n\mu_n^2}}\right).
$$

Secondly, we have $\sigma_n\mu_n^2/2 \geq 2\alpha/5$. This give ultimately

$$
R(\theta_s) \geq \frac{3}{10}\alpha\frac{\varsigma_n}{\sigma_n}\left(1 - \frac{4}{3}\sqrt{\frac{8R(\theta^\star)}{\frac{4}{5}\alpha\frac{\varsigma_n}{\sigma_n}}}\right) = \frac{3}{10}\alpha\frac{\varsigma_n}{\sigma_n}\left(1 - \sqrt{\frac{160}{9}\frac{R(\theta^\star)}{\alpha\frac{\varsigma_n}{\sigma_n}}}\right).
$$

Simplifying this expression gives

$$
R(\theta_s) \geq \frac{3}{10}\alpha\frac{\varsigma_n}{\sigma_n}\left(1 - \sqrt{18\frac{R(\theta^\star)}{\alpha\frac{\varsigma_n}{\sigma_n}}}\right). \tag{50}
$$

Combining the two bounds. We now simply combine the upper bound of Eq. (49) and the lower bound of Eq. (49). We get

$$
\frac{R(\theta_s)}{R(\theta_b)} \geq \frac{3}{50}\frac{\kappa_{\mathsf{T}}}{\kappa_{\mathsf{U}}}\left[\frac{1 - \sqrt{18\frac{\sigma_n}{\varsigma_n}\frac{R(\theta^\star)}{\alpha}}}{1 + 2\frac{\sigma_1}{\varsigma_1}\frac{R(\theta^\star)}{\alpha}}\right]. \tag{51}
$$

We may prefer the other form, introducing the positive part $(x)_+ = \max(0, x)$ and using $50/3 < 17$:

$$R(\theta_b) \leq 17\frac{\kappa_{\mathsf{U}}}{\kappa_{\mathsf{T}}}\left[\frac{1+2\frac{\sigma_1}{\varsigma_1}\frac{R(\theta^\star)}{\alpha}}{\left(1-\sqrt{18\frac{\sigma_n}{\varsigma_n}\frac{R(\theta^\star)}{\alpha}}\right)_+}\right] R(\theta_s) \overset{\text{def.}}{=} 17\frac{\kappa_{\mathsf{U}}}{\kappa_{\mathsf{T}}} c_\alpha R(\theta_s). \tag{52}$$

Finally, with Assumption 4 we have

$$1+2\frac{\sigma_1}{\varsigma_1}\frac{R(\theta^\star)}{\alpha} \leq \frac{3}{2},$$
$$1-\sqrt{18\frac{\sigma_n}{\varsigma_n}\frac{R(\theta^\star)}{\alpha}} \geq \frac{1}{2},$$

so that $c_\alpha \leq 2$. ■

Appendix B. Low-noise classification tasks

Taking big step size is particularly critical in classification tasks. In this section, we build on the result of Pillaud-Vivien et al. (2018b) to relate classification performances with Hilbert norm. Recall notably the notations of Section 4.

Assumptions. The following assumption comes from (A1) in Pillaud-Vivien et al. (2018b). It is well characterized in usual image classification settings.

Assumption 5 (Strong margin condition.) *We have $g^\star(x) \geq \delta$ for some $\delta \in (0, 1)$.*

The second assumption characterizes the statistical optimality of $\nu^\star$. It does not assume the regression function to belong to $\mathcal{H}$, but ensures some proximity in $\mathcal{L}_\infty$ norm. It is close to (A4) in Pillaud-Vivien et al. (2018b).

Assumption 6 (Statistical optimality of the population loss' optimum.) *We have that*

$$\operatorname{Sign}\left(g^\star(x)\right)\nu^\star(x) \geq \delta/2. \tag{53}$$

This assumption is satisfied as soon as the regression function can be approximated by a function of the RKHS with precision $\delta/2$ in $\mathcal{L}_\infty$ norm. For instance, a sufficient condition is the regression function $g^\star(x)$ to belong to $\mathcal{H}$. Then $g^\star = \nu^\star$ and the assumption is satisfied. Note that this always imply that $\theta^\star$ reaches 0 test error for sufficiently many samples, which is the key hindsight of Pillaud-Vivien et al. (2018b). Indeed, for a proper choice of regularization λ one has that

$$\|\theta^\star - \nu^\star\|_{\mathcal{H}} \lesssim n^{-\frac{br}{1+b(2r+1)}},$$

where (r, b) are the parameters of the source and capacity condition, both of which characterizes the difficulty of the learning task, see Blanchard and Mücke (2016). This implies that for sufficiently many training samples n, $\theta^\star$ will be close to $\nu^\star$ in Hilbert norm, which implies proximity in $\mathcal{L}_\infty$ (pointwise) norm.

Hilbert norm proximity implies statistical optimality The following lemma is very close to Lemma 1 in Pillaud-Vivien et al. (2018b), and is a direct consquence of our assumption. We first introduce

$$\Omega_+ = \{x; g^\star(x) \geq \delta\}\,, \quad \Omega_- = \{x; g^\star(x) \leq -\delta\}\,. \tag{54}$$

Next lemma basically relies on the decomposition

$$\|\theta - g^\star\|_{\mathcal{L}_\infty} \leq \|\theta - \nu^\star\|_{\mathcal{L}_\infty} + \|\nu^\star - g^\star\|_{\mathcal{L}_\infty}\,.$$

Lemma 14 (Small Hilbert norm implies statistical optimality) *Consider an estimator θ which satisfies*

$$\|\theta - \nu^\star\|_{\mathcal{H}} \leq \frac{\delta}{2C_K}.$$

Then, this estimator is statistically optimal, in the sense that it has 0 excess error:

$$B(\theta) - \inf_{\theta \in \mathcal{H}} B(\theta) = 0.$$

Proof First of all, we leverage the fact that the Hilbert norm upper bounds the L_∞ norm, with

$$\|\theta - \nu^\star\|_{L_\infty} \leq C_K \|\theta - \nu^\star\|_{\mathcal{H}} \leq \frac{\delta}{2}.$$

Then, on Ω_+, whose definition is given in Eq. (54), we have that

$$\forall x \in \Omega_+, \theta_x > g^\star(x) - \|\theta - \nu^\star\|_{L_\infty} - \|\nu^\star - g^\star\|_{L_\infty} \geq \delta - \frac{\delta}{2} - \frac{\delta}{2} = 0,$$

so θ will have accurate prediction for all positive labels. The same goes for negative labels. Thus, θ has 0 test error. ■

Thus, we see that the *Hilbert norm is a good proxy for minimizing the test error B*.

Appendix C. Regression tasks and comparison with spectral filters

If the downstream task is regression, then we can still apply our result by introducing the *population covariance operator*,

$$\mathsf{U} = \int \phi(x) \otimes \phi(x) \mathrm{d}\rho_x(x). \tag{55}$$

Then, Theorem 5 holds by considering (recall the definition of the population loss $\mathcal{P}$ in Eq. (17))

$$R(\theta) = \frac{1}{2}\|\theta - \nu^\star\|_{\mathsf{U}}^2 = \mathcal{P}(\theta) - \inf_{\nu \in \mathcal{H}} \mathcal{P}(\nu),$$

which is nothing but the *excess risk* of the estimator. Then, under the assumptions of Theorem 5 we have that

$$R(\theta_b) \;\leq\; 34 \frac{\kappa_{\tilde{\mathsf{U}}}}{\kappa_{\mathsf{T}}} R(\theta_s). \tag{56}$$

Gradient descent for kernel ridge regression has been widely studied in the past, to say the least. Equation (56) appears to be in contradiction with most of them. In this section, we emphasize the limit of our assumptions to point out that there is no conflict with existing theory.

Early stage of training. The bound in Eq. (56) ensures better generalization when taking big step size, if the r.h.s is bigger than 1. However, the pioneering work of Yao et al. (2007) established that the learning rate had no influence in the generalization capabilities of the estimator. A key difference though is that the results of Theorem 5 only holds in the early stage of training, when the optimization error α is big w.r.t to the statistical error $R(\theta^\star)$: otherwise, Assumption 4 is not satisfied. In contrast, results of the like of Yao et al. (2007) holds for sufficiently many samples n, and require a number of steps t bounded by below by a power of n – they require an upper-bound on α, while we require a lower-bound in Assumption 4.

Is Eq. (56) informative? As mentioned above, Eq. (56) ensures better generalization of big step size only if the r.h.s is bigger than 1. However, in the particular scenario of kernel regression, the empirical covariance T is the *discretization* of the population covariance U. Thus, numerous results bound the discrepancy between the two, notably when the capacity condition holds, see *e.g.* Proposition 5.3 to 5.5 in Blanchard and Mücke (2016). In these settings, we can expect the ratio $\kappa_{\mathsf{T}}/\kappa_{\mathsf{U}}$ to go to 1 for large number of samples n. Thus, we cannot conclude in better excess risk of θ_b compared to θ_s.

Comparison with spectral filters. Spectral filters are an elegant way to describe a wide family of regularization for kernel regression Gerfo et al. (2008); Bauer et al. (2007). In a nutshell, it relies on studying the class of estimator characterized by a filter function g_λ, where λ is a regularization parameter, equal to $1/t$ in the case of early stopping in GD. GD with moderate step sizes is a spectral filter; but GD with large step size is not. We now explain this difference, which helps to build an intuition on our result.

Consider the estimator $\theta = g_\lambda(\mathsf{T})S^*y$, where S is the so-called sampling operator defined in Appendix D.1. The unregularized solution is obtained with $\lambda = 0$, for which we must have $g_{\lambda=0}(\sigma) = \sigma^{-1}$. We denote it with $\theta^\star = \mathsf{T}^{-1}S^*y$, and we can see how does θ approaches the unregularized optimum. We have

$$\begin{aligned}
\langle \theta - \theta^\star, e_i \rangle &= \left\langle g_\lambda(\mathsf{T})S^*y - \mathsf{T}^{-1}S^*y, e_i \right\rangle \\
&= \left\langle \left(g_\lambda(\mathsf{T})\mathsf{T}^{-1} - \mathbf{I}\right)\mathsf{T}^{-1}S^*y, e_i \right\rangle \\
&= \left\langle \left(g_\lambda(\mathsf{T})\mathsf{T} - \mathbf{I}\right)\theta^\star, e_i \right\rangle \\
&= \left(g_\lambda(\sigma_i)\sigma_i - 1\right)\langle \theta^\star, e_i \rangle \, .
\end{aligned}$$

If we start from $\theta_0 \neq 0$, this relation turns to $|\langle \theta - \theta^\star, e_i \rangle| = |1 - g_\lambda(\sigma_i)\sigma_i| \, |\langle \theta_0 - \theta^\star, e_i \rangle|$ in the case of GD. We denote the residual with $r_\lambda(\sigma) = |1 - \sigma g_\lambda(\sigma)|$. We then compare r_λ for various spectral filters in Fig. 5. Note that for gradient descent, we have $r_{1/t}(\sigma) = |1 - \eta\sigma|^t$ and we recover the expression we obtained from Eq. (3). The key hindsight is that spectral filters will learn, *i.e optimize*, the *biggest eigendirection* first. For instance, truncated regression uses as estimator the first eigencomponents of the unregularized estimator, leaving the smaller eigencomponents untouched. This is at odds with what we aim at with big learning rate. There is no contradictions though, as we want in the end to minimize the excess risk R – a quadratic with operator U – and we assumed the empirical covariance T to be a discretization of U. Thus, in this settings F is a good proxy for R and minimizing the biggest eigendirection first will make the excess risk R decrease faster. This corresponds to having level sets well aligned in Fig. 1.

Theorem 5 in practice. The limits of this subsection – low optimization regime, low value for $\kappa_{\mathsf{T}}/\kappa_{\mathsf{U}}$ and difference with spectral filters – can be mitigated for multiple reasons. First of all,

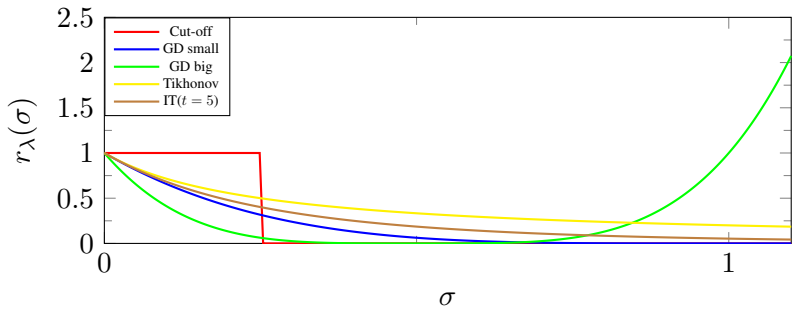


Figure 5: Residual of various spectral filters, with regularization $\lambda = 1/4$ or $t = 4$ for GD. The best filter is spectral cut-off *(red)*. The resulting estimator is purely directed along the smallest eigenvectors of T. Gradient descent with small step sizes *(blue)* and (iterated) Tikhonov *(yellow, brown)* mimick this filter. On the other hand, gradient descent with big step sizes *(green)* is not an admissible filter in the sense of Gerfo et al. (2008), as it attenuates less the biggest component.

as we discussed earlier kernel regression can simply be a mean in order to solve a *classification task*, in which case the statistical results of spectral filters are no longer relevant. Secondly, there can be big discrepancies between the train and test set in practice. Indeed, the risk R with which estimators are compared is often a separate test set, with fixed condition number κ_{U}. Additionally, data augmentation can be used on the train set, which then introduces spurrious directions in the empirical covariance matrix T. Thus, even though spectral filters are optimal in theoretical settings, taking big step size can prove useful in practical scenari, which are covered by our settings with quadratic forms of $\mathcal{H}$.

Appendix D. Gradient descent updates in practice

D.1. Useful operators

We assume there are n training samples. If considered, the test loss consists of m samples.

We denote $\hat{\mathsf{S}}, \hat{\mathsf{S}}^\star$ the *sampling* operator and its dual, which are defined as

$$\begin{aligned} \hat{\mathsf{S}} : \mathcal{H} \to \mathbb{R}^n, \quad \forall f \in \mathcal{H}, \;\; \hat{\mathsf{S}}(f) &= \frac{1}{\sqrt{n}} \begin{pmatrix} \langle f, \phi(x_1) \rangle_{\mathcal{H}} \\ \vdots \\ \langle f, \phi(x_n) \rangle_{\mathcal{H}} \end{pmatrix} \\ \hat{\mathsf{S}}^\star : \mathbb{R}^n \to \mathcal{H}, \quad \forall \alpha \in \mathbb{R}^n, \;\; \hat{\mathsf{S}}^\star(\alpha) &= \frac{1}{\sqrt{n}} \sum_{i=1}^n \alpha_i \phi(x_i), \end{aligned} \tag{57}$$

so that the covariance operator $\mathsf{T} = \hat{\mathsf{S}}^\star \hat{\mathsf{S}}$ and the kernel matrix K write

$$\begin{aligned} \mathsf{T} : \mathcal{H} \to \mathcal{H}, \quad \mathsf{T} &= \hat{\mathsf{S}}^\star \hat{\mathsf{S}} \\ K/n : \mathbb{R}^n \to \mathbb{R}^n, \quad \frac{K}{n} &= \hat{\mathsf{S}} \hat{\mathsf{S}}^\star. \end{aligned} \tag{58}$$

The population version are $\mathsf{S}, \mathsf{S}^\star, \mathsf{U}$. There are written with an expectation, or with the test dataset as a proxy.

Note that we have $\mathsf{T}^{-1}\hat{\mathsf{S}}^\star = \hat{\mathsf{S}}^\star(K/n)^{-1}$. We denote σ_i, e_i the spectrum of T and u_i the eigenvectors of K/n (not K), with the same spectrum. The eigenvectors in $\mathcal{H}$ and $\mathbb{R}^n$ are related with

$$\forall i \in \{1, \dots, n\}, \;\; u_i = \frac{1}{\sqrt{\sigma_i}} \hat{\mathsf{S}} e_i, \;\; e_i = \frac{1}{\sqrt{\sigma_i}} \hat{\mathsf{S}}^\star u_i.$$

Finally, an estimator $\theta \in \mathcal{H}$ can be represented with a vector $\alpha \in \mathbb{R}^n$. Specifically, we have the relation

$$\theta = \sqrt{n}\hat{\mathsf{S}}^\star \alpha \iff \sqrt{n}\hat{\mathsf{S}}\theta = K\alpha \iff \alpha = \sqrt{n}K^{-1}\hat{\mathsf{S}}\theta. \tag{59}$$

D.2. Gradient descent on the Hilbert norm is possible

Different spectrum between $\mathcal{H}$ and $\mathbb{R}^n$. We denote the training loss with F and the Hilbert norm with $L_{\mathcal{H}}$. Given the relation of Eq. (59), we have that

$$\begin{aligned} F(\theta) &= \frac{1}{2}\|\theta - \theta^\star\|_{\mathsf{T}}^2 = \frac{1}{2n}\|K(\alpha - \alpha^\star)\|_{\mathbb{R}^n}^2, \\ L_{\mathcal{H}}(\theta) &= \frac{1}{2}\|\theta - \theta^\star\|_{\mathcal{H}}^2 = \frac{1}{2}\|\alpha - \alpha^\star\|_K^2, \end{aligned} \tag{60}$$

where we overloaded F to be a function of $\mathcal{H}_n$ *and* $\mathbb{R}^n$. Specifically, we used $F(\alpha) = F \circ \sqrt{n}\hat{\mathsf{S}}^\star(\alpha)$. Recall that K/n and T share the same spectrum. Thus, F is a quadratic whose spectrum is $(\sigma_1, \dots, \sigma_n)$ w.r.t the variable θ, but spectrum $\left(n\sigma_1^2, \dots, n\sigma_n^2\right)$ w.r.t the variable α. Likewise, $L_{\mathcal{H}}$ is a quadratic with spectrum $(1, \dots, 1)$ w.r.t the variable θ, but a spectrum $(n\sigma_1, \dots, n\sigma_n)$ w.r.t the variable α.

Do we care about this difference? The global picture behind what follows is that α is isomorphic to $\mathsf{T}^{-1/2}\mathcal{H}$. If we expressed the estimator θ as a combination of eigenbasis vector, that is $\theta = \sum_i \beta_i e_i$, then β is isormorphic to $\mathcal{H}$ and the distinction does not hold. The fact that the estimator writes as a combination of $\phi(x_i)$ with α adds another level of geometric distortion.

Gradient descent in $\mathcal{H}$. In the Hilbert space $\mathcal{H}$, the updates are easy:

$$\begin{aligned} \theta_{t+1} = \theta_t - \eta\mathsf{T}(\theta_t - \theta^\star) &\iff \theta_t - \theta^\star = (\mathbf{I} - \eta\mathsf{T})^t(\theta_0 - \theta^\star), \quad \text{(GD on } F\text{)} \\ \theta_{t+1} = \theta_t - \eta(\theta_t - \theta^\star) &\iff \theta_t - \theta^\star = (1-\eta)^t(\theta_0 - \theta^\star), \quad \text{(GD on } L_{\mathcal{H}}\text{)}. \end{aligned}$$

The big learning rate range is then $\eta_s < 2/(\sigma_1 + \sigma_n) < \eta_b < (2/\sigma_1)$.

Gradient descent in $\mathbb{R}^n$. In practice, we do not have access to $\alpha^\star$, or only though it's evaluation with K. Yet, we are still able to minimize these quadratic form through the gradient. *E.g.* when $\alpha^\star$ is defined through $K\alpha^\star = y$ in the unregularized settings, or $(K + n\lambda)\alpha^\star = y$ in the Tikhonov-regularized case. The gradient descent updates on the train loss read:

$$\begin{aligned} \alpha_{t+1} &= \alpha_t - \eta\frac{K^2}{n}(\alpha_t - \alpha^\star) = \alpha_t - \eta\frac{K}{n}(K\alpha_t - y) \\ &\iff \alpha_t - \alpha^\star = \left(\mathbf{I} - \eta\frac{K^2}{n}\right)^t(\alpha_0 - \alpha^\star) \qquad \text{(GD on } F\text{)} \end{aligned} \tag{61}$$

Here, the range of learning rate is $\eta_s < 2/\left[n(\sigma_1^2 + \sigma_n^2)\right] < \eta_b < 2/\left[n\sigma_1^2\right]$. Interestingly, we can still do gradient descent on the Hilbert norm in closed form!

$$\begin{aligned} \alpha_{t+1} &= \alpha_t - \eta K(\alpha_t - \alpha^\star) = \alpha_t - \eta(K\alpha_t - y) \\ &\iff \alpha_t - \alpha^\star = (\mathbf{I} - \eta K)^t (\alpha_0 - \alpha^\star) \qquad \text{(GD on } L_{\mathcal{H}}\text{)} \end{aligned} \tag{62}$$

Here, the optimal learning rate is $1/[n\sigma_1]$. Interestingly, choosing a big learning rate in the range $1/\left[n(\sigma_1 + \sigma_n)\right] < \eta_b < 2/\left[n\sigma_1\right]$ results in an estimator which is closed in *euclidean* norm ($\mathbb{R}^n$) to $\alpha^\star$. Note that even though we can evaluate the gradient of $L_{\mathcal{H}}$, we *cannot* evaluate its value. Indeed, the objective function would read

$$\frac{1}{2}\left\|\alpha - \alpha^\star\right\|^2 = \frac{1}{2}\left\|\alpha - K^{-1}y\right\|^2$$

which is not accessible without inverting the (regularized) kernel matrix.

Appendix E. Additional details on the experiment

Setting the learning rate. We give additional details on the plot "test accuracy function of train loss α" in Fig. 3. The plot is averaged over 10 initialization for θ_0. We used $\eta_s = 1/\sigma_1$ and $\eta_b = \tau \cdot 2/\sigma_1$, with $\tau = 1 - 10^{-5}$. We elaborate on these choices:

- The optimal learning rate for upper bounding for σ_1-smooth, σ_n-strongly convex function is $\eta_{\text{opt}} = 2/(\sigma_1 + \sigma_n)$, as explained in the discussion of Assumption 2. However, this requires a massive amount of steps to converge. This is due to the terms depending on the initialization in the lower bound for t_b, t_s in Eqs. (31) and (42). Thus, we set $\eta_s = 1/\sigma_1$, which is the optimal rate for σ_1-smooth function, and we do observe fast convergence with this choice.
- Instead of choosing $\eta_b \in [2/(\sigma_1 + \sigma_n), 2/\sigma_1]$, we use $\eta_b = \tau \cdot 2/\sigma_1$, with τ chosen with the experiment on the test accuracy (Fig. 3, left). Indeed, setting $\tau = 1$ can result in situation where there can't be convergence; and choosing $\eta_b > \eta_{\text{opt}}$, as we describe in the theory, results in very slow convergence.

This discrepancy between theory and practice is due to our proof which is very conservative in the error bound. A more refined analysis would rely on $\theta_b - \theta^\star$ (resp. $\theta_s - \theta^\star$) belonging to the span of the k-th first (resp. last) eigenvectors. Besides, in practical settings the learning rate is an hyperparameter to tune, which is exactly the approach we used to produce Fig. 3.