Optimization for Machine Learning

Raphaël Berthier

January 14, 2026

These notes are associated to lectures given at Sorbonne Université. They are evolving and might contain errors. Feel free to use them, report mistakes (`raphael.berthier@inria.fr`) or contribute (github). Please cite this work when using or adapting it (license CC BY 4.0).

Abstract

This course provides a theoretical introduction to optimization methods tailored for machine learning. As modern learning tasks often involve complex, non-convex cost functions and massive datasets, standard optimization techniques require adaptation. The course explores stochastic and non-convex optimization strategies, focusing on their role in training machine learning models efficiently and ensuring generalization beyond training data. Emphasis is placed on foundational concepts such as overfitting, complexity control, variance reduction, and implicit regularization. While the approach is theoretical and often idealized, the goal is to build core intuitions that inform and improve practical machine learning methods.

Contents

Acknowledgements

The author thanks Loucas Pillaud-Vivien for his help in building those lecture notes, and Alejandro Saldarriaga for optimizing the proof of Lemma 7.3.

1 Introduction

Optimization in machine learning training. Many computer science problems are too complex to be solved directly by an algorithm coded by a human being. For instance, recognizing whether there is a bike in an image depends on the pixels of the image in an intricate way that no one is able to express to a computer. However, as humans can easily solve such a task, there must be a successful algorithm.

In such situations, machine learning comes to the rescue. Its strategy is to leave some free parameters in the algorithm. The optimal value of these parameters are unknown to the computer scientist; they are found by the machine itself by fitting a database of solved problem instances.

This training phase of the machine learning algorithm uses an optimization algorithm, as it seeks the optimal value of free parameters in order to minimize a cost function. This cost function describes the performance of the algorithm in solving the problem instances of the database.

Peculiarities of optimization in machine learning. Optimization is a well-established field of applied mathematics, with applications in computer science, engineering, operations research, and economics. It provides numerous numerical methods and theoretical analyses to understand them. The choice of the optimization algorithm should depend on the structure of the optimization problem: the structure of the cost function—notably whether it is convex or not—, the type of queries one can make on the function to be optimized—function, gradient, Hessian values?—, or the computation time of different operations. The optimization problems that appear in machine learning have peculiar structures that call for specific optimization techniques.

First, the cost function depends on the full training database, which can be enormous in modern machine learning applications. As a consequence, computing a function value, or a function gradient, can take a long time. This motivates a focus on stochastic methods, that compute the function values and gradients on a randomly sampled subpart of the database only, which is resampled at each iteration.

Second, the computer scientist should focus not only on fitting the training data, but also ensuring that learned algorithms are able to generalize to new problem instances. As a consequence, statistical concerns should be taken into account in the optimization process.

Finally, many machine learning algorithms lead to non-convex optimization problems, for which it is challenging to have any significant theoretical analysis, although great machine learning performance can be routinely observed in these cases.

As a consequence, the fields of stochastic and non-convex optimization have been greatly stimulated by the development of machine learning methods. Some analyses are able to control the generalization ability of the machine learning method, beyond its performance on the database.

Goal of this course. These notes aim at providing an introduction to optimization methods and analyses suited for machine learning problems. Our focus is theoretical. Provided the current limitations of theory in explaining the practice of machine learning methods, we present vanilla and sometimes idealized algorithms and analyses. However, we will see that even simplified analyses build intuitions and central concepts such as overfitting, complexity control, variance reduction, implicit regularization or benign overfitting. These intuitions can guide the improvement of state-of-the-art methods in practical situations.

Section 2 introduces the stakes of optimization in machine learning. Section 3 introduces stochastic gradient descent in an abstract formalism that enables to unify several algorithms. Section 4 presents the convergence analysis of stochastic gradient descent and discusses some practical consequences. These first three sections form the core of the course.

We then turn to more specialized topics, chosen biasedly by the author. Section 5 introduces importance sampling through exercises. Section 6 introduces variance reduction. Finally, Section 7 discusses some attempts to understand the regularization induced by neural networks and their non-convex optimization.

2 Empirical risk minimization and generalization error

2.1 Formal expression of a learning problem

In this course, we restrict our discussion on learning problems to supervised learning problems only. This still covers many practical applications and makes discussions more concrete.

In a supervised learning problem, we want to predict an output $y \in \mathcal{Y}$ from an input $x \in \mathcal{X}$. One can think of them as $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$. Often, the input and output spaces are not naturally $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$. For instance, when $\mathcal{X}$ is not a vector space, it is classical to choose an embedding of $\mathcal{X}$ in a (finite-dimensional) vector space and make the prediction of y as a function of the embedding. Further, in binary classification, it is more natural to take $\mathcal{Y} = \{-1, 1\}$. However, for convexity concerns, it is preferable to embed $\mathcal{Y} = \{-1, 1\} \subset \mathbb{R}$ and to aim at making predictions in $\mathbb{R}$ that have the same sign as the true output, see Remark 2.1 for more comments or [Bach, 2024, Section 4] for a more detailed discussion. As a consequence, we will often take $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$ as a running example.

Finally, we assume that the data comes from some probability distribution $\mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$.

In order to transform the supervised learning problem into an optimization problem, we need to choose a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. This loss function $\ell(y, \widehat{y})$ quantifies the price to pay for predicting $\widehat{y}$ when the true output is y. For instance, in regression, a common choice is the squared loss $\ell(y, \widehat{y}) = \frac{1}{2}(y - \widehat{y})^2$, while in binary classification, a common choice is the logistic loss $\ell(y, \widehat{y}) = \log(1 + \exp(-y\widehat{y}))$. Once this loss is chosen, we seek a function $\varphi : \mathcal{X} \to \mathcal{Y}$ that minimizes the so-called *expected risk* or *generalization loss / error*:

$$\mathcal{R}(\varphi) = \mathbb{E}[\ell(y, \varphi(x))] \,, \qquad (x, y) \sim \mathcal{P} \,.$$

Here, we have rephrased our learning problem as the minimization of a function $\mathcal{R}(\varphi)$; thus optimization theory starts to play a role. However, we face several difficulties. First, we would like to optimize over the infinite-dimensional space of functions $\mathcal{X} \to \mathcal{Y}$, which is intractable. We will restrict ourselves to a class of functions $\mathcal{F}$ that is more manageable, for instance a subset of functions parametrized by a finite-dimensional space. Second, we have no direct access to the distribution $\mathcal{P}$, which makes the computation of our objective function $\mathcal{R}(\varphi)$ impossible. Instead, in statistical learning, we seek to learn from n data samples $(x_1, y_1), \ldots, (x_n, y_n)$ that are i.i.d. from $\mathcal{P}$.

In order to approximate the expected risk $\mathcal{R}(\varphi)$, we can use the so-called *empirical risk* or *training loss / error*:

$$\widehat{\mathcal{R}}(\varphi) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \varphi(x_i)) \,.$$

Statistical wisdom warns that minimizing the empirical loss $\widehat{\mathcal{R}}(\varphi)$ instead of the expected loss $\mathcal{R}(\varphi)$ can lead to overfitting. There exists functions φ that predict well on the training data $(x_1, y_1), \ldots, (x_n, y_n)$ but generalize poorly on new data $(x, y) \sim \mathcal{P}$. Said differently, there exists functions φ that minimize the empirical loss $\widehat{\mathcal{R}}(\varphi)$ but are not close to optimizing the expected loss $\mathcal{R}(\varphi)$. Such overfitting can be detected by a train / test split of the data or by cross-validation.

In order to reduce a potential overfitting, classical approaches consist in restricting the class of functions over which we minimize the empirical loss, or in adding a regularization term to the empirical loss. Both approaches can be seen as ways to control the complexity of the function φ that we learn. The next section quantifies the tradeoff when restricting the class of function.

Remark 2.1. As a side note, we can mention that the loss function ℓ is a choice of the computer scientist. It should reflect the true cost function of the computer scientist if the algorithm makes poor predictions, but it is also chosen to make the resulting optimization problems tractable. For instance, the logistic loss is differentiable and convex in its second arguments, which makes it preferable to the 0-1 loss that is non-differentiable and non-convex, although the 0-1 loss is more faithful to the true cost function of the computer scientist. The quadratic loss might overestimate the cost of outliers, but leads in linear regression to quadratic convex functions that are easy to optimize.

2.2 Decomposition of the error

2.2.1 Decomposition between estimation and approximation errors

Let us denote $\mathcal{Y}^{\mathcal{X}}$ the set of functions from $\mathcal{X}$ to $\mathcal{Y}$ and $\mathcal{F}$ the class of functions from $\mathcal{X} \to \mathcal{Y}$ over which we optimize the empirical risk. For instance, we often consider parametrized classes of functions $\varphi(., \theta) : \mathcal{X} \to \mathcal{Y}$, $\theta \in \mathbb{R}^p$, in which case $\mathcal{F} = \{\varphi(., \theta) \,|\, \theta \in \mathbb{R}^p\}$.

We denote by $\varphi_*(\mathcal{Y}^{\mathcal{X}})$ a minimizer of the expected loss $\mathcal{R}(\varphi)$ over $\mathcal{Y}^{\mathcal{X}}$ and by $\widehat{\varphi}_*(\mathcal{F})$ a minimizer of the empirical loss $\widehat{\mathcal{R}}(\varphi)$ over $\mathcal{F}$. In this section, we study the suboptimality gap $\mathcal{R}(\widehat{\varphi}_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}}))$ of $\widehat{\varphi}_*(\mathcal{F})$.

Let $\varphi_*(\mathcal{F})$ be a minimizer of the expected loss $\mathcal{R}(\varphi)$ over $\mathcal{F}$. We can decompose the suboptimality gap as follows:

$$\mathcal{R}(\widehat{\varphi}_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}})) = \underbrace{\mathcal{R}(\widehat{\varphi}_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{F}))}_{\text{estimation error}} + \underbrace{\mathcal{R}(\varphi_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}}))}_{\text{approximation error}} .$$

The second term is called the *approximation error*. It quantifies the price to pay for restricting the class of functions over which we optimize.

The first term is called the *estimation error*. It measures how suboptimal is the function built from the data in minimizing the expected loss over $\mathcal{F}$. Typically, we control the estimation error as follows:

$$\begin{aligned}\mathcal{R}(\widehat{\varphi}_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{F})) = {} & \mathcal{R}(\widehat{\varphi}_*(\mathcal{F})) - \widehat{\mathcal{R}}(\widehat{\varphi}_*(\mathcal{F})) \\ & + \widehat{\mathcal{R}}(\widehat{\varphi}_*(\mathcal{F})) - \widehat{\mathcal{R}}(\varphi_*(\mathcal{F})) \\ & + \widehat{\mathcal{R}}(\varphi_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{F})) .\end{aligned}$$

As $\widehat{\varphi}_*(\mathcal{F})$ is a minimizer of the empirical loss $\widehat{\mathcal{R}}(\varphi)$ over $\mathcal{F}$, we have that $\widehat{\mathcal{R}}(\widehat{\varphi}_*(\mathcal{F})) - \widehat{\mathcal{R}}(\varphi_*(\mathcal{F})) \leqslant 0$. We thus obtain

$$\mathcal{R}(\widehat{\varphi}_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{F})) \leqslant 2 \sup_{\varphi \in \mathcal{F}} \left| \widehat{\mathcal{R}}(\varphi) - \mathcal{R}(\varphi) \right| . \tag{2.1}$$

For all $\varphi \in \mathcal{F}$, by the law of large numbers, we have that

$$\widehat{\mathcal{R}}(\varphi) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \varphi(x_i)) \xrightarrow[n \to \infty]{} \mathbb{E}[\ell(y, \varphi(x))] = \mathcal{R}(\varphi) \qquad \text{almost surely} .$$

To control the estimation error through Eq. (2.1), one needs a uniform law of large numbers over $\mathcal{F}$.

This upper-bound suggests that the estimation error improves as $n \to \infty$, but degrades as $\mathcal{F}$ increases (for the inclusion). On the contrary, the approximation error does not depend on n, but improves as $\mathcal{F}$ increases. As a consequence, choosing the class of functions $\mathcal{F}$ is a tradeoff between the approximation error and the estimation error. As more samples are available, one can afford to choose a larger class of functions $\mathcal{F}$.

2.2.2 Optimization error

In the section above, we have discussed the decomposition of the error for the function $\widehat{\varphi}_*(\mathcal{F})$ that minimizes the empirical loss $\widehat{\mathcal{R}}(\varphi)$ over $\mathcal{F}$. In this course, we study how to minimize this empirical loss. However, the optimization algorithms do not provide an exact minimizer: an optimization error remains. Here, we describe how the error decomposition is perturbed by this optimization error.

Let $\widehat{\varphi}$ be any function in $\mathcal{F}$. (We use the notation $\widehat{\varphi}$ to suggest that this function has been computed from the data through an algorithm, typically gradient descent on the empirical risk.) We can still decompose the suboptimality gap into an estimation and an approximation error:

$$\mathcal{R}(\widehat{\varphi}) - \mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}})) = \underbrace{\mathcal{R}(\widehat{\varphi}) - \mathcal{R}(\varphi_*(\mathcal{F}))}_{\text{estimation error}} + \underbrace{\mathcal{R}(\varphi_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}}))}_{\text{approximation error}} . \tag{2.2}$$

Typically, we control the estimation error as follows:

$$\begin{aligned}\mathcal{R}(\widehat{\varphi}) - \mathcal{R}(\varphi_*(\mathcal{F})) = \mathcal{R}(\widehat{\varphi}) - \widehat{\mathcal{R}}(\widehat{\varphi}) \\ + \widehat{\mathcal{R}}(\widehat{\varphi}) - \widehat{\mathcal{R}}(\varphi_*(\mathcal{F})) \\ + \widehat{\mathcal{R}}(\varphi_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{F})) .\end{aligned} \tag{2.3}$$

As $\widehat{\varphi}_*(\mathcal{F})$ is a minimizer of the empirical loss $\widehat{\mathcal{R}}(\varphi)$ over $\mathcal{F}$, we have that

$$\widehat{\mathcal{R}}(\widehat{\varphi}) - \widehat{\mathcal{R}}(\varphi_*(\mathcal{F})) \leqslant \widehat{\mathcal{R}}(\widehat{\varphi}) - \widehat{\mathcal{R}}(\widehat{\varphi}_*(\mathcal{F})) . \tag{2.4}$$

Thus we obtain

$$\mathcal{R}(\widehat{\varphi}) - \mathcal{R}(\varphi_*(\mathcal{F})) \leqslant \underbrace{\widehat{\mathcal{R}}(\widehat{\varphi}) - \widehat{\mathcal{R}}(\widehat{\varphi}_*(\mathcal{F}))}_{\text{optimization error}} + 2 \sup_{\varphi \in \mathcal{F}} \left| \widehat{\mathcal{R}}(\varphi) - \mathcal{R}(\varphi) \right| . \tag{2.5}$$

Note that this bound is a straighforward generalization of (2.1). The result is that the estimation error is controlled by two terms. The first one is an *optimization error*: the suboptimality gap of $\widehat{\varphi}$ in the minimization of $\widehat{\mathcal{R}}$. The second one is, again, the uniform law of large numbers over $\mathcal{F}$.

2.2.3 Relevance of this decomposition and goal of the course

A traditional way to show the success of a machine learning algorithm would be to follow the steps above, and to control separately the approximation error $\mathcal{R}(\varphi_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}}))$, the optimization error $\widehat{\mathcal{R}}(\widehat{\varphi}) - \widehat{\mathcal{R}}(\widehat{\varphi}_*(\mathcal{F}))$ in minizing the empirical risk, and the deviation in the uniform law of large numbers $\sup_{\varphi \in \mathcal{F}} \left| \widehat{\mathcal{R}}(\varphi) - \mathcal{R}(\varphi) \right|$. The approximation error is controlled by choosing a class of functions $\mathcal{F}$ that is rich enough, the optimization error is controlled by choosing a good optimization algorithm, while there are classical methods to quantify the uniform law of large numbers, see e.g. [Bach, 2024, Section 4.4].

However, it is important to evaluate how tight this approach is. In the decomposition of Eq. (2.2), the approximation and estimation error are both non-negative quantities: one can not compensate for the other. As a consequence, it is necessary to control both errors in order to control the expected risk. The same can not be said for the decomposition made in Eq. (2.3). The different terms can have different signs, and thus cancel each other (although the sum must be non-negative). The resulting bound (2.5), that does not take into account these cancellations, might be overly pessimistic. Moreover, the bound (2.5) also follows from the inequality (2.4), that might also be loose. As a consequence, there are settings where the sketch of proof outlined above fails to provide a meaningful bound, although the machine learning algorithm performs well in practice.

Take-home message for optimizers. When designing an optimization algorithm for machine learning, we can have two objectives in mind. The first one is simply to reduce the optimization error $\widehat{\mathcal{R}}(\widehat{\varphi}) - \widehat{\mathcal{R}}(\widehat{\varphi}_*(\mathcal{F}))$. When doing so, we leave it to statisticians to choose the class of functions $\mathcal{F}$ so that the other terms in the decomposition of the expected risk are controlled. The second one is to design algorithms that control the whole estimation error. We will see that, in some situations, an improved estimation error can be obtained when deteriorating the optimization error on the empirical risk. For instance, stopping a (stochastic) gradient descent early, or taking a single pass on the data samples rather than multiple passes, might actually improve the estimation error.

There are incompressible losses: the optimal expected risk $\mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}}))$ and the approximation error $\mathcal{R}(\varphi_*(\mathcal{F})) - \mathcal{R}(\varphi_*(\mathcal{Y}^{\mathcal{X}}))$. As a consequence, it is useless to work on reducing the optimization and estimation errors much below these losses. This justifies that in machine learning, we often seek a fast optimization algorithm that provides a satisfactory solution, rather than a slower algorithm that would provide a very accurate solution.

3 Stochastic gradient descent

In this section, we motivate multi-pass stochastic gradient descent as a computationally-efficient alternative to gradient descent for the minimization of the empirical risk (Sections 3.1, 3.2). We then abstract the stochastic gradient descent algorithm in order to unify several algorithms (Section 3.3). We show that single-pass stochastic gradient descent can be seen as a stochastic gradient descent algorithm on the expected risk (Section 3.4), and that coordinate gradient descent can be seen as a stochastic gradient descent (Section 3.5).

In all of this section, we set ourselves in the setting developed in Section 2. We assume that we are given a parametrized class of functions $\mathcal{F} = \{\varphi(., \theta) \,|\, \theta \in \mathbb{R}^p\}$.

Notations. For a vector $\theta \in \mathbb{R}^p$, we denote $\theta(1), \dots, \theta(p)$ its coordinates. Given a differentiable function $f : \mathbb{R}^p \to \mathbb{R}$, we denote $\partial_1 f, \dots, \partial_p f$ its partial derivatives, $\nabla f = (\partial_1 f, \dots, \partial_p f)$ its gradient and $\nabla^2 f = (\partial_{ij} f)_{1 \leqslant i,j \leqslant p}$ its Hessian.

3.1 Gradient descent on the empirical risk

Our first task is to optimize the empirical risk $\widehat{\mathcal{R}}(\varphi)$ over $\mathcal{F}$. More precisely, we seek to minimize $f : \mathbb{R}^p \to \mathbb{R}$, defined as

$$f(\theta) = \widehat{\mathcal{R}}(\varphi(.,\theta)) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta) , \qquad \text{where } \ell_i(\theta) = \ell(y_i, \varphi(x_i, \theta)) .$$

Typical optimization algorithms for this task take advantage of first-order information, namely that the partial derivatives $\partial_j f(\theta)$ of f and its gradient $\nabla f(\theta)$ can be computed, or of second-order information, namely that the Hessian $\nabla^2 f(\theta)$ of f can be computed. In machine learning, the number of parameters p of the function class is often very large, and the computation of the Hessian $\nabla^2 f(\theta) \in \mathbb{R}^{p \times p}$ is quadratic in p; as a consequence, second-order methods can not be used. Instead, first-order methods that require only access to the gradient $\nabla f(\theta) \in \mathbb{R}^p$ are preferred.

The emblematic first-order method is gradient descent. Choose an initialization $\theta_0 \in \mathbb{R}^p$ and a stepsize sequence $\gamma_k \in \mathbb{R}_+$, $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$, compute

$$\theta_{k+1} = \theta_k - \gamma_k \nabla f(\theta_k) . \tag{3.1}$$

Under mild assumptions, if the stepsizes γ_k are chosen appropriately, the sequence θ_k converges to a local minimizer of f.

3.2 Multi-pass stochastic gradient descent on the empirical risk

A major difficulty in running the gradient descent described above is that the computation of the gradient

$$\nabla f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell_i(\theta)$$

has a linear complexity in n, the number of data samples, as it implies reading all the samples. This can be prohibitive in modern machine learning applications, where n can be in the order of millions or billions.

In order to mitigate this complexity, the stochastic gradient strategy consists in computing the gradient on a randomly sampled subset $S_{k+1} \subset \{1, \dots, n\}$ of the samples, called a *mini-batch*, which is re-sampled at each iteration k.

More formally, choose an initialization $\theta_0 \in \mathbb{R}^p$, a stepsize sequence $\gamma_k \in \mathbb{R}_+$, $k \in \mathbb{N}$ and a mini-batch size m. Then for all $k \in \mathbb{N}$, sample S_{k+1} uniformly among subsets of $\{1, \dots, n\}$ of size m (independently of the past) and compute

$$\theta_{k+1} = \theta_k - \frac{\gamma_k}{m} \sum_{i \in S_{k+1}} \nabla \ell_i(\theta_k) . \tag{3.2}$$

This algorithm is called *multi-pass stochastic gradient descent*, as a sample i is used several times in the computation of the stochastic gradient, if the number of iterations is large enough.

The rationale behind stochastic gradient descent is that the information contained in the data samples is largely redundant, and thus that sampling only a few samples at each iteration is enough to make some progress on the learning problem. Even if there are some fluctuations due to the randomness of the sampling, we expect that these fluctuations average out over the iterations, provided that the stepsizes are small enough. This remark suggests that we can consider an extreme case of mini-batches of size $m = 1$: for all $k \in \mathbb{N}$, sample i_{k+1} uniformly in $\{1, \dots, n\}$ (independently of the past) and compute

$$\theta_{k+1} = \theta_k - \gamma_k \nabla \ell_{i_{k+1}}(\theta_k)\,. \tag{3.3}$$

Due to the fluctuations in the gradient, it is not obvious whether a stochastic gradient descent can converge to a local minimizer. In Section 4, we will see that it is indeed the case if the stepsizes decrease appropriately, and under some mild assumptions.

3.3 Abstract stochastic gradient descent

In this section, we abstract the notion of stochastic gradient descent. Consider a differentiable function $f : \mathbb{R}^p \to \mathbb{R}$ that we seek to optimize. Assume that we have no direct access to $\nabla f(\theta)$, but instead we can generate samples ξ from a distribution $\mathcal{Q}$, and compute a quantity $g(\theta, \xi)$, such that $\mathbb{E}[g(\theta, \xi)] = \nabla f(\theta)$. The random variable $g(\theta, \xi)$, $\xi \sim \mathcal{Q}$ is called an *unbiased stochastic gradient* of f at θ.

In this setting, the stochastic gradient descent algorithm is defined as follows: choose an initialization $\theta_0 \in \mathbb{R}^p$ and a stepsize sequence $\gamma_k \in \mathbb{R}_+$, $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$, sample ξ_{k+1} from $\mathcal{Q}$ (independently of the past) and compute

$$\theta_{k+1} = \theta_k - \gamma_k g(\theta_k, \xi_{k+1})\,. \tag{3.4}$$

Example: multi-pass stochastic gradient descent. The multi-pass stochastic gradient descent on the empirical risk described in the previous section fits this abstract framework. In this case, the function f is the empirical risk as a function of θ:

$$f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta)\,.$$

Consider first the case of mini-batches of size $m = 1$. In this case, we take $\mathcal{Q}$ to be the uniform distribution on $\{1, \dots, n\}$, $\xi = i$ is the uniformly sampled index of the data sample, and $g(\theta, i) = \nabla \ell_i(\theta)$. It is then indeed true that $\mathbb{E}[g(\theta, \xi)] = \nabla f(\theta)$ and the iteration of Eq. (3.4) gives the iteration of Eq. (3.3).

Further, consider the case of mini-batches of any size $m \leqslant n$. In this case, we take $\mathcal{Q}$ to be the uniform distribution over subsets of $\{1, \dots, n\}$ of size m, $\xi = S$

is the uniformly sampled subset of the data samples, and the stochastic gradient $g(\theta, S) = \frac{1}{m}\sum_{i\in S}\nabla \ell_i(\theta)$. Let us check that such a stochastic gradient is unbiased:

$$\begin{aligned}
\mathbb{E}[g(\theta,S)] &= \mathbb{E}\left[\frac{1}{m}\sum_{i\in S}\nabla \ell_i(\theta)\right] = \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{n}\mathbb{1}_{\{i\in S\}}\nabla \ell_i(\theta)\right] \\
&= \frac{1}{m}\sum_{i=1}^{n}\mathbb{P}\left(i\in S\right)\nabla \ell_i(\theta) = \frac{1}{m}\sum_{i=1}^{n}\frac{m}{n}\nabla \ell_i(\theta) = \frac{1}{n}\sum_{i=1}^{n}\nabla \ell_i(\theta) \\
&= \nabla f(\theta)\,.
\end{aligned}$$

With this unbiased stochastic gradient, the iteration of Eq. (3.4) gives the iteration of Eq. (3.2).

We now turn to other instantiations of the abstract stochastic gradient descent algorithm.

3.4 Single-pass stochastic gradient descent on the expected risk

Let us now consider a variant of the multi-pass stochastic gradient descent algorithm (3.3), where when we use a sample i to compute a stochastic gradient descent, we decide not to use it again in the future. We then sample among the remaining data samples at the next iterations. This algorithm is called *single-pass stochastic gradient descent*.

Of course, such an algorithm would be limited to $k = n$ steps. Moreover, as the data (x_i, z_i) are i.i.d., the order in which we use the samples has no influence on the distribution of the iterations θ_k. As a consequence, we can assume that we use the samples in the original order $i = 1, \dots, n$. This gives the following algorithm: choose an initialization $\theta_0 \in \mathbb{R}^p$ and a stepsize sequence $\gamma_k \in \mathbb{R}_+$, $k \in \mathbb{N}$. Then for all $k \in \{0, \dots, n-1\}$, compute

$$\theta_{k+1} = \theta_k - \gamma_k \nabla \ell_{k+1}(\theta_k)\,. \tag{3.5}$$

The interest in this variant is mainly theoretical: it stems from the fact that single-pass stochastic gradient descent can be seen as a stochastic gradient descent algorithm on the expected risk (in the sense of Section 3.3).

Indeed, write the expected risk as a function of θ:

$$f(\theta) = \mathcal{R}(\varphi(.,\theta)) = \mathbb{E}[\ell(y,\varphi(x,\theta))]\,, \qquad (x,y)\sim\mathcal{P}\,.$$

The gradient of f is then

$$\nabla f(\theta) = \mathbb{E}[\nabla_\theta \ell(y,\varphi(x,\theta))]\,, \qquad (x,y)\sim\mathcal{P}\,.$$

This suggests to consider the unbiased stochastic gradient

$$g(\theta,\xi) = \nabla_\theta \ell(y,\varphi(x,\theta))\,, \qquad \xi = (x,y)\sim\mathcal{P} = \mathcal{Q}\,.$$

The abstract stochastic gradient descent requires to generate i.i.d. samples $\xi_k = (x_k, y_k) \sim \mathcal{P} = \mathcal{Q}$, $k = 1, 2, \ldots$ and computes

$$\theta_{k+1} = \theta_k - \gamma_k \nabla_\theta \ell(y_{k+1}, \varphi(x_{k+1}, \theta_k)) \, .$$

As we have only n such i.i.d. samples, we can run this algorithm only for n steps. This matches exactly Eq. (3.5).

To sum up,

- multi-pass stochastic gradient descent optimizes the empirical risk,
- single-pass stochastic gradient descent optimizes the expected risk.

The latter point can seem counter-intuitive, as we do not have a direct access to the expected risk. However, there is no contradiction as we can run only n steps of stochastic gradient descent on the expected risk (where n is the number of data samples), thus the precision we can achieve on the expected risk is limited by the amount of data we have.

However, single-pass stochastic gradient descent is less prone to overfitting than multi-pass stochastic gradient descent.

Remark 3.1 (stochastic approximation). In Section 3, we have been interested, so far, only in the optimization of functions that can be written as expectations:

$$f(\theta) = \mathbb{E}\left[f(\theta, \xi)\right] \, , \qquad \xi \sim \mathcal{Q} \, , \tag{3.6}$$

with an obvious abuse of notation. The optimization of such functions is the subject of the field of *stochastic approximation*.

Multi-pass stochastic gradient descent with batch size $m = 1$ corresponds to $\xi = i \sim \mathrm{Unif}(\{1, \ldots, n\})$ and $f(\theta, \xi) = \ell_i(\theta)$. Multi-pass stochastic gradient descent with a general batch size m corresponds to $\xi = S$ with uniform distribution over subsets of $\{1, \ldots, n\}$ of size m and $f(\theta, \xi) = \frac{1}{m} \sum_{i \in S} \ell_i(\theta)$. Single-pass stochastic gradient descent corresponds to $\xi = (x, y) \sim \mathcal{P}$ and $f(\theta, \xi) = \ell(y, \varphi(x, \theta))$.

More generally, in the presence of the structure (3.6), and if we can sample from Q, we can optimize f through a stochastic gradient descent with $g(\theta, \xi) = \nabla_\theta f(\theta, \xi)$.

In the next section, we provide an example of a stochastic gradient descent that does not follow the structure (3.6).

3.5 Coordinate gradient descent

In this section, we consider the minimization of any differentiable function $F : \mathbb{R}^p \to \mathbb{R}$—not necessarily a risk in a learning problem. When the number of parameters p is large, running a gradient descent iteration of the form (3.1) can be computationally expensive, as it requires the computation of the full gradient $\nabla F(\theta) \in \mathbb{R}^p$ at each iteration. Instead, the *coordinate gradient descent* algorithm consists in computing only one partial derivative at each iteration, and updating the corresponding coordinate of θ. The updated coordinate is randomly sampled at each iteration, uniformly in $\{1, \dots, p\}$.

More formally, choose an initialization $\theta_0 \in \mathbb{R}^p$ and a stepsize sequence $\gamma_k \in \mathbb{R}_+$, $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$, sample j_{k+1} uniformly in $\{1, \dots, p\}$ (independently of the past) and compute θ_{k+1} such that:

$$\begin{aligned} \theta_{k+1}(j_{k+1}) &= \theta_k(j_{k+1}) - \gamma_k \partial_{j_{k+1}} F(\theta_k) \,, \\ \theta_{k+1}(j) &= \theta_k(j) \qquad \text{for all } j \neq j_{k+1} \,. \end{aligned}$$

This algorithm is a stochastic gradient descent algorithm in the sense of Section 3.3 with $\xi = j \sim \mathrm{Unif}(\{1, \dots, p\})$ and $g(\theta, j) = \partial_j F(\theta) e_j$ (where $e_1, \dots, e_p$ is the canonical basis of $\mathbb{R}^p$). As $\mathbb{E}[g(\theta, \xi)] = \frac{1}{p} \nabla F(\theta)$, the coordinate gradient descent is a stochastic gradient descent algorithm on the function $f(\theta) = \frac{1}{p} F(\theta)$.

3.6 Exercises

Exercise 3.2 (coordinate stochastic gradient descent)**.** Let $F : \mathbb{R}^p \to \mathbb{R}$ be a differentiable function. We assume that F can be written as a finite sum of function $F(\theta) = \frac{1}{n} \sum_{i=1}^n F_i(\theta)$, where each F_i is differentiable. (For instance, $F(\theta)$ could be the empirical risk of a machine learning algorithm on a dataset of size n.)

To minimize F, we propose a coordinate stochastic gradient descent algorithm, defined as follows. Choose an initialization $\theta_0 \in \mathbb{R}^p$ and a stepsize $\gamma \in \mathbb{R}_+$. Then for all $k \in \mathbb{N}$, sample i_{k+1} uniformly at random in $\{1, \dots, n\}$ and j_{k+1} uniformly at random in $\{1, \dots, p\}$, independently of each other and of the past. Compute θ_{k+1} such that

$$\begin{aligned} \theta_{k+1}(j_{k+1}) &= \theta_k(j_{k+1}) - \gamma_k \partial_{j_{k+1}} F_{i_{k+1}}(\theta_k) \,, \\ \theta_{k+1}(j) &= \theta_k(j) \qquad \text{for all } j \neq j_{k+1} \,. \end{aligned}$$

1. Show that $\mathbb{E}\theta_1 = \theta_0 - \frac{\gamma_0}{p} \nabla F(\theta_0)$.

2. Show that the coordinate stochastic gradient descent algorithm is an abstract stochastic gradient descent algorithm in the sense of Section 3.3. In particular, show that stochastic gradients are unbiased.

Exercise 3.3 (gossip problem)**.** Consider a communication network, that we model as a graph $G = (V, E)$, where V is the set of communication nodes and $E \subset \{\{i, j\} \,|\, i \neq j \in V\}$ is the set of communication links or edges. The gossip problem is an elementary problem in decentralized distributed computing where each node $v \in V$ is given a value $\theta_0(v)$ and the goal of the network is to compute the average of all these values. However, there is no central node in the network that can collect all of the information and computed the average. Instead, the nodes can only communicate along the edges of the graph when the communication link is activated.

To solve the gossip problem, we propose the following algorithm. At each iteration $k \in \mathbb{N}$, a communication link $(v_{k+1}, w_{k+1}) \in E$ is activated, uniformly at random. The nodes v_{k+1} and w_{k+1} exchange their values $\theta_k(v_{k+1})$ and $\theta_k(w_{k+1})$ and update them by averaging the two values:

$$\begin{aligned}\theta_{k+1}(v_{k+1}) &= \frac{1}{2}\theta_k(v_{k+1}) + \frac{1}{2}\theta_k(w_{k+1})\,,\\ \theta_{k+1}(w_{k+1}) &= \frac{1}{2}\theta_k(v_{k+1}) + \frac{1}{2}\theta_k(w_{k+1})\,.\end{aligned}$$

All other nodes keep their values unchanged: $\theta_{k+1}(z) = \theta_k(z)$ for $z \neq v_{k+1}, w_{k+1}$.

1. Show that the algorithm described above is a stochastic gradient descent algorithm in the sense of Section 3.3, on a function f to be determined.

2. Describe the set of minimizers of f.

4 Analyses for stochastic gradient descent

This section introduces the analyses for stochastic gradient descent. There exists many analyses of stochastic gradient descents, where the results depend on the assumptions on the function f. Our goal here is not to cover all of these analyses, but only a few representative results. The assumptions on the function f that we use are reminded in Section 4.1.

For the sake of comparison, we then present in Section 4.2 the analysis for gradient descent. We then continue with the analysis of stochastic gradient descent in Section 4.3. This gives different results for all of the stochastic gradient descents we have derived in Section 3.

4.1 Function structures

In this section, f denotes a function from $\mathbb{R}^p$ to $\mathbb{R}$.

4.1.1 Convexity

Definition 4.1. The function f is said to be *convex* if for all $\theta, \theta' \in \mathbb{R}^p$ and all $\alpha \in [0, 1]$, we have

$$f((1-\alpha)\theta + \alpha\theta') \leqslant (1-\alpha)f(\theta) + \alpha f(\theta')\,.$$

Proposition 4.2. *We assume that f is continuously differentiable. The following conditions are equivalent:*

(i) f is convex,

(ii) *for all* $\theta, \theta' \in \mathbb{R}^p$, $f(\theta') \geqslant f(\theta) + \langle \nabla f(\theta), \theta' - \theta \rangle$,

(iii) *for all* $\theta, \theta' \in \mathbb{R}^p$, $\langle \nabla f(\theta) - \nabla f(\theta'), \theta - \theta' \rangle \geqslant 0$.

Remark 4.3. The condition (iii) is generalization of the condition "f' is increasing" for functions of one variable.

Proof. $(i) \Rightarrow (ii)$**.** Let $\theta, \theta' \in \mathbb{R}^p$. For $\alpha \in [0, 1]$, we denote

$$g(\alpha) = (1 - \alpha) f(\theta) + \alpha f(\theta') - f((1 - \alpha)\theta + \alpha\theta')\,.$$

By (i), $g \geqslant 0$. Moreover, $g(0) = 0$. Thus

$$0 \leqslant g'(0) = f(\theta') - f(\theta) - \langle \nabla f(\theta), \theta' - \theta \rangle\,.$$

$(ii) \Rightarrow (i)$**.** Let $\theta, \theta' \in \mathbb{R}^p$ and $\alpha \in [0, 1]$. By (ii), we have

$$\begin{aligned} f(\theta') &\geqslant f((1 - \alpha)\theta + \alpha\theta') + \langle \nabla f((1 - \alpha)\theta + \alpha\theta'), (1 - \alpha)(\theta' - \theta) \rangle\,, \\ f(\theta) &\geqslant f((1 - \alpha)\theta + \alpha\theta') + \langle \nabla f((1 - \alpha)\theta + \alpha\theta'), \alpha(\theta - \theta') \rangle\,. \end{aligned}$$

We take the linear combination of these two inequalities with respective weights α and $1 - \alpha$:

$$(1 - \alpha) f(\theta) + \alpha f(\theta') \geqslant f((1 - \alpha)\theta + \alpha\theta')\,.$$

$(ii) \Rightarrow (iii)$**.** Let $\theta, \theta' \in \mathbb{R}^p$. From (ii), we have

$$\begin{aligned} 0 &\leqslant f(\theta') - f(\theta) - \langle \nabla f(\theta), \theta' - \theta \rangle\,, \\ 0 &\leqslant f(\theta) - f(\theta') - \langle \nabla f(\theta'), \theta - \theta' \rangle\,. \end{aligned}$$

Summing these two inequalities gives (iii).

$(iii) \Rightarrow (ii)$**.** Let $\theta, \theta' \in \mathbb{R}^p$. For $\alpha \in [0, 1]$, define

$$h(\alpha) = f((1 - \alpha)\theta + \alpha\theta') - f(\theta) - \langle \nabla f(\theta), \alpha(\theta' - \theta) \rangle\,.$$

By (iii),

$$h'(\alpha) = \langle \nabla f((1 - \alpha)\theta + \alpha\theta') - \nabla f(\theta), \theta' - \theta \rangle \geqslant 0\,.$$

Thus

$$f(\theta') - f(\theta) - \langle \nabla f(\theta), \theta' - \theta \rangle = h(1) \geqslant h(0) = 0\,.$$

□

Proposition 4.4. *We assume that* f *is twice continuously differentiable. The following conditions are equivalent:*

(i) f *is convex,*

(ii) for all $\theta \in \mathbb{R}^p$, $\nabla^2 f(\theta) \succcurlyeq 0$.

Proof. $(i) \Rightarrow (ii)$**.** Let $\theta \in \mathbb{R}^p$ and $v \in \mathbb{R}^p$. For $\delta \geqslant 0$, we define

$$g(\delta) = \langle \nabla f(\theta + \delta v) - \nabla f(\theta), v \rangle\,.$$

By (i) and Prop. 4.2, for all $\delta \geqslant 0$, $g(\delta) \geqslant 0$. Moreover, $g(0) = 0$. Thus

$$0 \leqslant g'(0) = \langle \nabla^2 f(\theta) v, v \rangle\,.$$

As this is true for all $v \in \mathbb{R}^p$, we have $\nabla^2 f(\theta) \succcurlyeq 0$.

$(ii) \Rightarrow (i)$**.** Let $\theta, \theta' \in \mathbb{R}^p$. For $\alpha \in [0,1]$, we define

$$h(\alpha) = \langle \nabla f((1-\alpha)\theta + \alpha\theta') - \nabla f(\theta), \theta' - \theta \rangle\,.$$

By (ii),

$$h'(\alpha) = \langle \nabla^2 f((1-\alpha)\theta + \alpha\theta')(\theta' - \theta), \theta' - \theta \rangle \geqslant 0\,.$$

Thus

$$\langle \nabla f(\theta') - \nabla f(\theta), \theta' - \theta \rangle = h(1) \geqslant h(0) = 0\,.$$

This allows to conclude using Prop. 4.2.

□

4.1.2 Strong convexity

Definition 4.5. Let $\mu > 0$. The function f is said to be *μ-strongly convex* if for all $\theta, \theta' \in \mathbb{R}^p$ and all $\alpha \in [0,1]$,

$$f((1-\alpha)\theta + \alpha\theta') \leqslant (1-\alpha) f(\theta) + \alpha f(\theta') - \frac{\mu}{2}\alpha(1-\alpha)\|\theta - \theta'\|^2\,.$$

Further, the function f is said to be *strongly convex* if it is μ-strongly convex for some $\mu > 0$.

Proposition 4.6. *The following conditions are equivalent:*

(i) f is μ-strongly convex,

(ii) the function $g(\theta) = f(\theta) - \frac{\mu}{2}\|\theta\|^2$ is convex.

Proof.

$$\begin{aligned}
&(1-\alpha)g(\theta) + \alpha g(\theta') - g((1-\alpha)\theta + \alpha\theta') \\
&\quad = (1-\alpha)f(\theta) - (1-\alpha)\frac{\mu}{2}\|\theta\|^2 + \alpha f(\theta') - \alpha\frac{\mu}{2}\|\theta'\|^2 \\
&\qquad - f((1-\alpha)\theta + \alpha\theta') + \frac{\mu}{2}\|(1-\alpha)\theta + \alpha\theta'\|^2 \\
&\quad = \Big[(1-\alpha)f(\theta) + \alpha f(\theta') - f((1-\alpha)\theta + \alpha\theta') - \alpha(1-\alpha)\frac{\mu}{2}\|\theta - \theta'\|^2\Big] \\
&\qquad + \frac{\mu}{2}\left[\|(1-\alpha)\theta + \alpha\theta'\|^2 + \alpha(1-\alpha)\|\theta - \theta'\|^2 - (1-\alpha)\|\theta\|^2 - \alpha\|\theta'\|^2\right]
\end{aligned}$$

As simple expansion shows that the last bracket is actually equal to 0. The proposition easily follows. □

Proposition 4.7. *We assume that f is continuously differentiable. The following conditions are equivalent:*

(i) *f is μ-strongly convex,*

(ii) *for all $\theta, \theta' \in \mathbb{R}^p$, $f(\theta') \geqslant f(\theta) + \langle \nabla f(\theta), \theta' - \theta \rangle + \frac{\mu}{2} \|\theta' - \theta\|^2$,*

(iii) *for all $\theta, \theta' \in \mathbb{R}^p$, $\langle \nabla f(\theta) - \nabla f(\theta'), \theta - \theta' \rangle \geqslant \mu \|\theta' - \theta\|^2$.*

Proof. These results are obtained by combining Prop. 4.2 and Prop. 4.6. □

Proposition 4.8 (implications of strong convexity)**.** *We assume that f is continuously differentiable and μ-strongly convex. Then:*

(i) *f has a unique minimizer θ_*,*

(ii) *we have the* Polyak-Lojasiewicz condition*: for all $\theta \in \mathbb{R}^p$,*

$$\frac{1}{2} \|\nabla f(\theta)\|^2 \geqslant \mu \left(f(\theta) - f(\theta_*) \right) ,$$

(iii) *for all $\theta, \theta' \in \mathbb{R}^p$, $\|\nabla f(\theta) - \nabla f(\theta')\| \geqslant \mu \|\theta - \theta'\|$.*

Proof. *(i)* We thus prove the existence of a minimizer for the function f. Let $\theta \in \mathbb{R}^p$. By Prop. 4.7(*ii*),

$$\begin{aligned} f(\theta) &\geqslant f(0) + \langle \nabla f(0), \theta \rangle + \frac{\mu}{2} \|\theta\|^2 \\ &\geqslant f(0) - \|\nabla f(0)\| \|\theta\| + \frac{\mu}{2} \|\theta\|^2 \xrightarrow[\|\theta\| \to \infty]{} +\infty \, . \end{aligned}$$

Thus there exists $R > 0$ such that for all $\theta \in \mathbb{R}^p$ with $\|\theta\| \geqslant R$, $f(\theta) \geqslant f(0) + 1$. Moreover, as f is continuous on the compact set $\overline{B(0, R)}$, it reaches its minimum on this set. This minimum is a global minimizer of f.

We now show the uniqueness of this global minimizer. Let θ_*, θ'_* be two global minimizers of f. By the definition of strong convexity, we have

$$f\left(\frac{\theta_* + \theta'_*}{2} \right) \leqslant \frac{f(\theta_*) + f(\theta'_*)}{2} - \frac{\mu}{2} \frac{1}{4} \|\theta_* - \theta'_*\|^2 \, .$$

Here, by optimality of θ_* and θ'_*, $f\left(\frac{\theta_* + \theta'_*}{2} \right) \geqslant f(\theta_*) = f(\theta'_*)$. Thus we must have $\|\theta_* - \theta'_*\|^2 = 0$. This proves the uniqueness.

(ii) Let $\theta, \theta' \in \mathbb{R}^p$. By Prop. 4.7(*ii*),

$$f(\theta') \geqslant f(\theta) + \langle \nabla f(\theta), \theta' - \theta \rangle + \frac{\mu}{2} \|\theta' - \theta\|^2 .$$

We now minimize in θ' over both sides of the inequality. The upper bound is minimized in $\theta' = \theta_*$. The lower bound is minimized in $\theta' = \theta - \frac{1}{\mu} \nabla f(\theta)$. This gives

$$\begin{aligned} f(\theta_*) &\geqslant f(\theta) + \left\langle \nabla f(\theta), -\frac{1}{\mu} \nabla f(\theta) \right\rangle + \frac{\mu}{2} \left\| -\frac{1}{\mu} \nabla f(\theta) \right\|^2 \\ &= f(\theta) - \frac{1}{2\mu} \|\nabla f(\theta)\|^2 . \end{aligned}$$

This gives the desired inequality.

(iii) Let $\theta, \theta' \in \mathbb{R}^p$. By Prop. 4.7(*iii*) and the Cauchy-Schwarz inequality,

$$\mu \|\theta - \theta'\|^2 \leqslant \langle \nabla f(\theta) - \nabla f(\theta'), \theta - \theta' \rangle \leqslant \|\nabla f(\theta) - \nabla f(\theta')\| \|\theta - \theta'\| .$$

This gives the desired inequality.

□

Proposition 4.9. *We assume that f is twice continuously differentiable. The following conditions are equivalent:*

(i) *f is μ-strongly convex,*

(ii) *for all $\theta \in \mathbb{R}^p$, $\nabla^2 f(\theta) \succcurlyeq \mu I_p$.*

Proof. This result is obtained by combining Prop. 4.4 and Prop. 4.6. □

4.1.3 Smoothness

In all of this section, we assume that f is *convex* and *continuously differentiable*.

Definition 4.10. Let $L > 0$. The function f is said to be *L-smooth* if for all $\theta, \theta' \in \mathbb{R}^p$, for all $\alpha \in [0, 1]$,

$$f((1-\alpha)\theta + \alpha\theta') \geqslant (1-\alpha) f(\theta) + \alpha f(\theta') - \frac{L}{2} \alpha(1-\alpha) \|\theta - \theta'\|^2 .$$

Further, the function f is said to be *smooth* if it is L-smooth for some $L > 0$.

Proposition 4.11. *The following conditions are equivalent:*

(i) *f is L-smooth,*

(ii) *the function $g(\theta) = \frac{L}{2} \|\theta\|^2 - f(\theta)$ is convex.*

Proof. The proof is similar to the proof of Prop. 4.6. □

Proposition 4.12. *The following conditions are equivalent:*

(i) f *is* L*-smooth,*

(ii) *for all* $\theta, \theta' \in \mathbb{R}^p$, $f(\theta') \leqslant f(\theta) + \langle \nabla f(\theta), \theta' - \theta \rangle + \frac{L}{2}\|\theta' - \theta\|^2$,

(iii) *for all* $\theta, \theta' \in \mathbb{R}^p$, $\langle \nabla f(\theta') - \nabla f(\theta), \theta' - \theta \rangle \leqslant L\|\theta' - \theta\|^2$

(iv) ∇f *is co-coercive: for all* $\theta, \theta' \in \mathbb{R}^p$,

$$\|\nabla f(\theta') - \nabla f(\theta)\|^2 \leqslant L\langle \theta' - \theta, \nabla f(\theta') - \nabla f(\theta) \rangle \,,$$

(v) ∇f *is* L*-Lipschitz: for all* $\theta, \theta' \in \mathbb{R}^p$, $\|\nabla f(\theta') - \nabla f(\theta)\| \leqslant L\|\theta' - \theta\|$.

Proof. $(i) \Leftrightarrow (ii) \Leftrightarrow (iii)$ These equivalences can be proven by combining Prop. 4.2 and Prop. 4.11.

$(ii) \Rightarrow (iv)$ Let $\theta, \theta', \theta'' \in \mathbb{R}^p$. By (ii),

$$f(\theta + \theta'') \leqslant f(\theta) + \langle \nabla f(\theta), \theta'' \rangle + \frac{L}{2}\|\theta''\|^2 \,.$$

Moreover, by convexity of f and Prop. 4.2(ii),

$$f(\theta + \theta'') \geqslant f(\theta') + \langle \nabla f(\theta'), \theta + \theta'' - \theta' \rangle \,.$$

Combining these two inequalities gives

$$\langle \nabla f(\theta') - \nabla f(\theta), \theta'' \rangle - \frac{L}{2}\|\theta''\|^2 \leqslant f(\theta) - f(\theta') + \langle \nabla f(\theta'), \theta' - \theta \rangle \,.$$

We now maximize over θ''. The maximum is obtained for

$$\theta'' = \frac{1}{L}\left(\nabla f(\theta') - \nabla f(\theta)\right) \,.$$

We obtain

$$\frac{1}{2L}\|\nabla f(\theta') - \nabla f(\theta)\|^2 \leqslant f(\theta) - f(\theta') + \langle \nabla f(\theta'), \theta' - \theta \rangle \,.$$

Of course, the same inequality holds with θ and θ' exchanged:

$$\frac{1}{2L}\|\nabla f(\theta') - \nabla f(\theta)\|^2 \leqslant f(\theta') - f(\theta) + \langle \nabla f(\theta), \theta - \theta' \rangle \,.$$

Adding these two inequalities gives the desired result.

$(iv) \Rightarrow (v)$ This easily follows by applying the Cauchy-Schwarz inequality to the right-hand side of (iv).

$(v) \Rightarrow (iii)$ This easily follows by combining the Cauchy-Schwarz inequality with (v).

□

Proposition 4.13. *We assume that f is twice continuously differentiable. The following conditions are equivalent:*

(i) f is L-smooth,

(ii) for all $\theta \in \mathbb{R}^p$, $\nabla^2 f(\theta) \preccurlyeq L I_p$.

Proof. This result is obtained by combining Prop. 4.4 and Prop. 4.11. □

4.2 Analysis of gradient descent

In this section, we assume that f is differentiable, μ-strongly convex and L-smooth for some $\mu, L > 0$. We denote θ_* the unique minimizer of f. These assumptions are not always satisfied for the empirical and expected risks of machine learning problems; however they are still useful as a first setting of analysis in order to grasp the qualitative behavior of stochastic gradient descents.

To start with, we analyze the gradient descent algorithm introduced in Section 3.1, that we recall here for convenience: choose an initialization $\theta_0 \in \mathbb{R}^p$ and a stepsize sequence $\gamma_k \in \mathbb{R}_+$, $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$, compute

$$\theta_{k+1} = \theta_k - \gamma_k \nabla f(\theta_k)\,.$$

Theorem 4.14. *Assume $\gamma_k = \gamma$ is constant and $\gamma \leqslant \frac{1}{L}$. Then*

$$\|\theta_k - \theta_*\|^2 \leqslant (1-\gamma\mu)^k \|\theta_0 - \theta_*\|^2\,.$$

In particular, for $\gamma = \frac{1}{L}$,

$$\|\theta_k - \theta_*\|^2 \leqslant \left(1 - \frac{\mu}{L}\right)^k \|\theta_0 - \theta_*\|^2\,.$$

The convergence is said to be linear*, as the bound is exponential in k. The condition number L/μ describes the typical time of convergence.*

Proof.

$$\begin{aligned}\|\theta_{k+1} - \theta_*\|^2 &= \|\theta_k - \theta_* - \gamma \nabla f(\theta_k)\|^2 \\ &= \|\theta_k - \theta_*\|^2 - 2\gamma\langle \nabla f(\theta_k), \theta_k - \theta_*\rangle + \gamma^2 \|\nabla f(\theta_k)\|^2\,.\end{aligned}$$

By co-coercivity of ∇f (Prop. 4.12(iv)),

$$\begin{aligned}\|\nabla f(\theta_k)\|^2 &= \|\nabla f(\theta_k) - \nabla f(\theta_*)\|^2 \\ &\leqslant L\langle \theta_k - \theta_*, \nabla f(\theta_k) - \nabla f(\theta_*)\rangle \\ &= L\langle \theta_k - \theta_*, \nabla f(\theta_k)\rangle\,.\end{aligned}$$

This gives

$$\|\theta_{k+1} - \theta_*\|^2 \leqslant \|\theta_k - \theta_*\|^2 + \gamma(-2 + \gamma L)\langle \nabla f(\theta_k), \theta_k - \theta_* \rangle \,.$$

By assumption, we have that $-2 + \gamma L \leqslant -2 + \frac{L}{L} = -1$ and by the strong convexity of f (Prop. 4.7(*iii*)),

$$\langle \nabla f(\theta_k), \theta_k - \theta_* \rangle = \langle \nabla f(\theta_k) - \nabla f(\theta_*), \theta_k - \theta_* \rangle \geqslant \mu \|\theta_k - \theta_*\|^2 \,.$$

This gives

$$\|\theta_{k+1} - \theta_*\|^2 \leqslant \|\theta_k - \theta_*\|^2 - \gamma\mu\|\theta_k - \theta_*\|^2 = (1 - \gamma\mu)\|\theta_k - \theta_*\|^2 \,,$$

and concludes the proof. □

Remark 4.15. Note that the above result on the distance $\|\theta_k - \theta_*\|$ to the optimum can be translated into a result on the suboptimality gap $f(\theta_k) - f(\theta_*)$. Indeed, as f is L-smooth,

$$f(\theta_k) - f(\theta_*) \leqslant \langle \nabla f(\theta_*), \theta_k - \theta_* \rangle + \frac{L}{2}\|\theta_k - \theta_*\|^2 \leqslant \frac{L}{2}(1 - \gamma\mu)^k \|\theta_0 - \theta_*\|^2 \,.$$

Recall that in machine learning applications—for instance when optimizing an empirical risk—we only seek a limited precision in the optimization as incompressible estimation and approximation errors remain in the generalization error. As a consequence, we sometimes present the above result differently, in order to discuss the number of iterations needed to reach a certain precision.

Corollary 4.16. *Fix $\varepsilon > 0$. Using a fixed stepsize $\gamma = 1/L$, and*

$$k \geqslant \left(\log \frac{\|\theta_0 - \theta_*\|}{\varepsilon}\right) \frac{L}{\mu}$$

iterations, we have $\|\theta_k - \theta_0\| \leqslant \varepsilon$.

Proof.

$$\|\theta_k - \theta_*\|^2 \leqslant \left(1 - \frac{\mu}{L}\right)^k \|\theta_0 - \theta_*\|^2 \leqslant e^{-k\mu/L}\|\theta_0 - \theta_*\|^2 \,,$$

and

$$e^{-k\mu/L}\|\theta_0 - \theta_*\|^2 \leqslant \varepsilon \qquad \Longleftrightarrow \qquad k \geqslant \left(\log \frac{\|\theta_0 - \theta_*\|}{\varepsilon}\right) \frac{L}{\mu} \,.$$

□

4.3 Analysis of abstract stochastic gradient descent

We now analyze the abstract stochastic gradient descent in the setting introduced in Section 3.3, that we recall here for convenience. In this setting, we can generate samples ξ from some distribution $\mathcal{Q}$ and compute unbiased stochastic gradients $g(\theta, \xi)$ of f at θ, in the sense that $\mathbb{E}g(\theta, \xi) = \nabla f(\theta)$. We then choose an initialization $\theta_0 \in \mathbb{R}^p$ and a stepsize sequence $\gamma_k \in \mathbb{R}_+$, $k \in \mathbb{N}$. Then for all $k \in \mathbb{N}$, we sample ξ_{k+1} from $\mathcal{Q}$ and compute

$$\theta_{k+1} = \theta_k - \gamma_k g(\theta_k, \xi_{k+1}) \, .$$

To prove a theorem on stochastic gradient descent, we make two assumptions.

Assumption 4.17. *The function f is μ-strongly convex.*

In particular, f has a unique minimum θ_*. Moreover, we will denote $\sigma^2 = \mathbb{E}\|g(\theta_*, \xi)\|^2$ the expected square norm of the stochastic gradient at the optimum.

Assumption 4.18. *There exists $M > 0$ such that for all $\theta, \theta' \in \mathbb{R}^p$,*

$$\mathbb{E}\|g(\theta, \xi) - g(\theta', \xi)\|^2 \leqslant M\langle \theta - \theta', \nabla f(\theta) - \nabla f(\theta')\rangle \, .$$

This assumption can be interpreted as follows. We could imagine that we would need to assume that the stochastic gradients are gradients of convex M-smooth functions, i.e., that they are co-coercive:

$$\|g(\theta, \xi) - g(\theta', \xi)\|^2 \leqslant M\langle \theta - \theta', \nabla g(\theta, \xi) - \nabla g(\theta', \xi)\rangle \quad \text{for } Q\text{-almost all } \xi \, .$$

In fact, we do not assume this but only the "expected" version of this inequality, where we take expectations on both sides.

Note that Assumption 4.18 implies that f is M-smooth. Indeed, by Jensen's inequality,

$$\begin{aligned} \|\nabla f(\theta) - \nabla f(\theta')\|^2 &= \|\mathbb{E}\left[g(\theta, \xi) - g(\theta', \xi)\right]\|^2 \leqslant \mathbb{E}\|g(\theta, \xi) - g(\theta', \xi)\|^2 \\ &\leqslant M\langle \theta - \theta', \nabla f(\theta) - \nabla f(\theta')\rangle \, . \end{aligned}$$

Before proving a result under these assumptions, we show how Assumption 4.18 can be checked in the cases introduced in Section 3.

Stochastic approximation setting. We set ourselves in the setting where $f(\theta) = \mathbb{E}f(\theta, \xi)$ for some $\xi \sim Q$ and that $g(\theta, \xi) = \nabla f(\theta, \xi)$. Assume that for all ξ, $f(., \xi)$ is convex, $L(\xi)$-smooth. Then Assumption 4.18 is satisfied with $M = \|L\|_{L^\infty(\mathcal{Q})}$.

Indeed, as $f(., \xi)$ is $L(\xi)$-smooth, we have that for all $\theta, \theta' \in \mathbb{R}^p$,

$$\|g(\theta, \xi) - g(\theta', \xi)\|^2 \leqslant L(\xi)\langle \theta - \theta', g(\theta, \xi) - g(\theta', \xi)\rangle \, .$$

Taking expectation on both sides gives

$$\begin{aligned} \mathbb{E}\|g(\theta, \xi) - g(\theta', \xi)\|^2 &\leqslant \mathbb{E}\left[L(\xi)\langle \theta - \theta', g(\theta, \xi) - g(\theta', \xi)\rangle\right] \\ &\leqslant \|L\|_{L^\infty(\mathcal{Q})}\langle \theta - \theta', \mathbb{E}g(\theta, \xi) - \mathbb{E}g(\theta', \xi)\rangle \\ &= \|L\|_{L^\infty(\mathcal{Q})}\langle \theta - \theta', \nabla f(\theta) - \nabla f(\theta')\rangle \, . \end{aligned}$$

Coordinate gradient descent. We assume that the function $F : \mathbb{R}^p \to \mathbb{R}$ on which we perform a coordinate gradient descent is μ-strongly convex and L-smooth. Recall that coordinate gradient descent corresponds to $\xi = j \sim \mathrm{Unif}(\{1, \dots, p\})$, $g(\theta, j) = \partial_j F(\theta) e_j$ and $f(\theta) = \frac{1}{p} F(\theta)$. Then Assumption 4.18 is satisfied with $M = L$.

Indeed,

$$\begin{aligned}
\mathbb{E}\|g(\theta, j) - g(\theta', j)\|^2 &= \mathbb{E}\left\|\partial_j F(\theta) e_j - \partial_j F(\theta') e_j\right\|^2 \\
&= \mathbb{E}(\partial_j F(\theta) - \partial_j F(\theta'))^2 \\
&= \frac{1}{p}\|\nabla F(\theta) - \nabla F(\theta')\|^2 \\
&\leqslant \frac{L}{p}\langle \theta - \theta', \nabla F(\theta) - \nabla F(\theta')\rangle \\
&= L\langle \theta - \theta', \nabla f(\theta) - \nabla f(\theta')\rangle \,.
\end{aligned}$$

Now that our assumptions are motivated, we now state our general theorem.

Theorem 4.19. *Make Assumptions 4.17 and 4.18. Assume that $\gamma_k = \gamma$ is constant and $\gamma \leqslant \frac{1}{2M}$. Then*

$$\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant (1 - \gamma\mu)^k \|\theta_0 - \theta_*\|^2 + \frac{2\gamma\sigma^2}{\mu} \,. \tag{4.1}$$

This theorem generalizes Theorem 4.14 to the stochastic setting. As the optimization procedure is random, we now bound the square distance to the optimum *in expectation.* The bound is the same as in the deterministic case, with an additional term $\frac{2\gamma\sigma^2}{\mu}$ due to the stochasticity of the gradients. It is a measure of the noise in the optimization process. This term is due to the fact that, even if we have reached the optimum $\theta_k = \theta_*$, if $\sigma^2 = \mathbb{E}\|g(\theta_*, \xi)\|^2 > 0$, a stochastic gradient might make the trajectory move away from the optimum. As a consequence, there is only a limited precision as $k \to \infty$:

$$\limsup_{k \to \infty} \mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \frac{2\gamma\sigma^2}{\mu} \,.$$

This precision might actually be sufficient in machine learning if approximation and estimation errors dominate. As a consequence, using a fixed-step size stochastic gradient descent might be a good strategy in practice.

More precisely, the asymptotic performance of $\frac{2\gamma\sigma^2}{\mu}$ is proportional to the (square) magnitude $\sigma^2 = \mathbb{E}\|g(\theta_*, \xi)\|^2$ of the stochastic gradients at optimum. This shows that the more "stochastic" the gradient descent, the larger the asymptotic performance. However, this term is also proportional to the stepsize γ, more precisely to the ratio γ/μ. This shows that one can limit the effect of noise by using smaller stepsizes. This effect is intuitive: taking small stepsizes enables to take advantage of the averaging of the stochastic gradients along the trajectory, and thus reduce the effect of stochasticity. However, reducing the stepsize degrades the first term of (4.1); as a consequence, there is a tradeoff between fast initial decay of the distance to the optimum and the precision of the final estimate.

Before we start the proof, let us introduce a convenient notation. Let $\mathcal{F}_k$ denote the sigma-algebra generated by the first k samples $\xi_1, \dots, \xi_k$. Then the conditional expectation $\mathbb{E}[\,\cdot\,|\mathcal{F}_k]$ can be understood as the expectation over $\xi_{k+1}, \xi_{k+2}, \dots$, where $\xi_1, \dots, \xi_k$ are considered as deterministic.

Proof.

$$\begin{aligned}
\|\theta_{k+1} - \theta_*\|^2 &= \|\theta_k - \theta_* - \gamma g(\theta_k, \xi_{k+1})\|^2 \\
&= \|\theta_k - \theta_*\|^2 - 2\gamma\langle g(\theta_k, \xi_{k+1}), \theta_k - \theta_*\rangle + \gamma^2\|g(\theta_k, \xi_{k+1})\|^2 \\
&\leqslant \|\theta_k - \theta_*\|^2 - 2\gamma\langle g(\theta_k, \xi_{k+1}), \theta_k - \theta_*\rangle \\
&\qquad + 2\gamma^2\|g(\theta_k, \xi_{k+1}) - g(\theta_*, \xi_{k+1})\|^2 + 2\gamma^2\|g(\theta_*, \xi_{k+1})\|^2 \,.
\end{aligned}$$

As θ_k is $\mathcal{F}_k$-measurable—it is a function of $\xi_1, \dots, \xi_k$ only—, it can be considered as a constant for $\mathbb{E}[\,\cdot\,|\mathcal{F}_k]$. We thus have

$$\begin{aligned}
\mathbb{E}\left[\|\theta_{k+1} - \theta_*\|^2 \,\middle|\, \mathcal{F}_k\right] &= \|\theta_k - \theta_*\|^2 - 2\gamma\left\langle \mathbb{E}[g(\theta_k, \xi_{k+1}) \,|\, \mathcal{F}_k], \theta_k - \theta_*\right\rangle \\
&\qquad + 2\gamma^2\mathbb{E}\left[\|g(\theta_k, \xi_{k+1}) - g(\theta_*, \xi_{k+1})\|^2 \,|\, \mathcal{F}_k\right] \\
&\qquad + 2\gamma^2\mathbb{E}\left[\|g(\theta_*, \xi_{k+1})\|^2 \,|\, \mathcal{F}_k\right] \,.
\end{aligned}$$

As the stochastic gradient are unbiased, $\mathbb{E}[g(\theta_k, \xi_{k+1}) \,|\, \mathcal{F}_k] = \nabla f(\theta_k)$. Moreover, by Assumption 4.18,

$$\begin{aligned}
\mathbb{E}\left[\|g(\theta_k, \xi_{k+1}) - g(\theta_*, \xi_{k+1})\|^2 \,|\, \mathcal{F}_k\right] &\leqslant M\langle\theta_k - \theta_*, \nabla f(\theta_k) - \nabla f(\theta_*)\rangle \\
&= M\langle\theta_k - \theta_*, \nabla f(\theta_k)\rangle \,.
\end{aligned}$$

Finally, we have that $\mathbb{E}\left[\|g(\theta_*, \xi_{k+1})\|^2 \,|\, \mathcal{F}_k\right] = \sigma^2$. This gives

$$\begin{aligned}
\mathbb{E}\left[\|\theta_{k+1} - \theta_*\|^2 \,\middle|\, \mathcal{F}_k\right] &\leqslant \|\theta_k - \theta_*\|^2 - 2\gamma\langle\nabla f(\theta_k), \theta_k - \theta_*\rangle \\
&\qquad + 2\gamma^2 M\langle\theta_k - \theta_*, \nabla f(\theta_k)\rangle + 2\gamma^2\sigma^2 \\
&= \|\theta_k - \theta_*\|^2 + 2\gamma(-1 + \gamma M)\langle\nabla f(\theta_k), \theta_k - \theta_*\rangle \\
&\qquad + 2\gamma^2\sigma^2 \,.
\end{aligned}$$

By assumption, we have that $-1+\gamma M \leqslant -1+\frac{M}{2M} = -\frac{1}{2}$ and by the strong convexity of f (Prop. 4.7(*iii*)),

$$\begin{aligned}
\mathbb{E}\left[\|\theta_{k+1} - \theta_*\|^2 \,\middle|\, \mathcal{F}_k\right] &\leqslant \|\theta_k - \theta_*\|^2 - \gamma\mu\|\theta_k - \theta_*\|^2 + 2\gamma^2\sigma^2 \\
&= (1 - \gamma\mu)\|\theta_k - \theta_*\|^2 + 2\gamma^2\sigma^2 \,.
\end{aligned}$$

In particular, taking expectations, we have

$$\mathbb{E}\|\theta_{k+1} - \theta_*\|^2 \leqslant (1 - \gamma\mu)\mathbb{E}\|\theta_k - \theta_*\|^2 + 2\gamma^2\sigma^2 \,.$$

To conclude, we seek a constant N such that $N = (1-\gamma\mu)N + 2\gamma^2\sigma^2$. This gives $N = \frac{2\gamma\sigma^2}{\mu}$. We then have

$$\mathbb{E}\|\theta_{k+1} - \theta_*\|^2 - N \leqslant (1-\gamma\mu)\left(\mathbb{E}\|\theta_k - \theta_*\|^2 - N\right) ,$$

and thus, by induction,

$$\begin{aligned} \mathbb{E}\|\theta_k - \theta_*\|^2 &\leqslant (1-\gamma\mu)^k\left(\|\theta_0 - \theta_*\|^2 - N\right) + N \\ &\leqslant (1-\gamma\mu)^k\|\theta_0 - \theta_*\|^2 + N \,. \end{aligned}$$

□

In the following corollary, we describe how to choose the stepsize γ and the number of iterations k in order to achieve a given error ε.

Corollary 4.20. *Make Assumptions 4.17 and 4.18. Fix $\varepsilon > 0$. Using a fixed stepsize*

$$\gamma = \frac{1}{2M + \frac{4\sigma^2}{\varepsilon\mu}}$$

and

$$k \geqslant 2\left(\log\frac{2\|\theta_0 - \theta_*\|^2}{\varepsilon}\right)\left(\frac{M}{\mu} + \frac{2\sigma^2}{\mu^2\varepsilon}\right)$$

iterations, we have that

$$\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \varepsilon \,.$$

Compare to Corollary 4.16: in the stochastic setting, when $\sigma^2 > 0$, the stepsize must be adapted to the noise level. The number of iterations needed to reach en error ε depends on condition number M/μ, which replaces the role of the condition number L/μ in the deterministic case. However, an additional term of the form $\sigma^2/(\mu^2\varepsilon)$ appears and plays a role when seeking a small error ε.

Proof. Note that $\gamma \leqslant 1/(2M)$ and thus we can apply Theorem 4.19. This gives

$$\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant (1-\gamma\mu)^k\,\|\theta_0 - \theta_*\|^2 + \frac{2\gamma\sigma^2}{\mu} \,. \tag{4.2}$$

As

$$\gamma = \frac{1}{2M + \frac{4\sigma^2}{\varepsilon\mu}} \leqslant \frac{1}{\frac{4\sigma^2}{\varepsilon\mu}} = \frac{\varepsilon\mu}{4\sigma^2} ,$$

the second term of Eq. (4.2) can be bounded by

$$\frac{2\gamma\sigma^2}{\mu} \leqslant \frac{\varepsilon}{2} \,.$$

We now bound the first term of Eq. (4.2). We have

$$\begin{aligned}(1-\gamma\mu)^k \|\theta_0-\theta_*\|^2 &\leqslant e^{-k\gamma\mu}\|\theta_0-\theta_*\|^2 \leqslant \frac{\varepsilon}{2} \\ \iff k &\geqslant \frac{1}{\gamma\mu}\log\frac{2\|\theta_0-\theta_*\|^2}{\varepsilon} = 2\left(\log\frac{2\|\theta_0-\theta_*\|^2}{\varepsilon}\right)\left(\frac{M}{\mu}+\frac{2\sigma^2}{\varepsilon\mu^2}\right).\end{aligned}$$

□

With a fixed stepsize, the asymptotic performance of stochastic gradient descent is limited by the noise level (when $\sigma^2 > 0$). This limiting performance can be improved by reducing the stepsize, but this degrades the initial decay of the distance to the optimum. This suggests a policy which is widely used in practice: start with a large stepsize to quickly improve the error, and reduce the stepsize when the error starts to plateau. Here, we propose an analysis of stochastic gradient descent with decaying stepsizes.

Theorem 4.21. *Make Assumptions 4.17 and 4.18. Take*

$$\gamma_k = \frac{\beta}{k_0+k}$$

where $\beta > \frac{1}{\mu}$ *and* k_0 *is chosen such that* $\gamma_0 = \frac{\beta}{k_0} \leqslant \frac{1}{2M}$. *Then*

$$\mathbb{E}\|\theta_k-\theta_*\|^2 \leqslant \frac{\nu}{k_0+k},$$

where $\nu = \max\left(k_0\|\theta_0-\theta_*\|^2, \frac{2\sigma^2\beta^2}{\beta\mu-1}\right)$.

With an appropriate choice of stepsizes, one can obtain that the expected square distance to optimum decays as $1/k$. As we had a linear convergence in the deterministic case, we observe that the noise severely degrades the convergence rate. Actually, this rate is optimal in the sense that it is the best one can achieve in the presence of noise.

Proof. We make a proof by induction. The initialization $k = 0$ is trivial. Assume that the result holds for some $k \in \mathbb{N}$. Then following the steps of the proof of Theorem 4.19, we have

$$\mathbb{E}\|\theta_{k+1}-\theta_*\|^2 \leqslant (1-\gamma_k\mu)\mathbb{E}\|\theta_k-\theta_*\|^2 + 2\gamma_k^2\sigma^2.$$

The only difference here is that the stepsize is variable, and thus it is less straightforward to combine these inequalities for different k. Using the induction hypothesis, we have

$$\begin{aligned}\mathbb{E}\|\theta_{k+1}-\theta_*\|^2 &\leqslant \left(1-\frac{\beta\mu}{k_0+k}\right)\frac{\nu}{k_0+k}+2\sigma^2\frac{\beta^2}{(k_0+k)^2}\\ &= \frac{(k_0+k-\beta\mu)\nu+2\sigma^2\beta^2}{(k_0+k)^2}\\ &= \frac{(k_0+k-1)\nu}{(k_0+k)^2}+\nu\frac{(1-\beta\mu)\nu+2\sigma^2\beta^2}{(k_0+k)^2}\,.\end{aligned}$$

By definition of ν, we have that $(1-\beta\mu)\nu+2\sigma^2\beta^2 \leqslant 0$. Further, $\frac{(k_0+k-1)}{(k_0+k)^2} \leqslant \frac{1}{k_0+k+1}$, and thus

$$\mathbb{E}\|\theta_{k+1}-\theta_*\|^2 \leqslant \frac{\nu}{k_0+k+1}\,.$$

This proves the induction step. □

4.4 Application for multi-pass stochastic gradient descent on the empirical risk

In this section, we analyse the multi-pass stochastic gradient descent introduced in Section 3.2. We recall that

$$f(\theta)=\frac{1}{n}\sum_{i=1}^{n}\ell_i(\theta)\,,$$

where $\ell_i(\theta)=\ell(y_i,\varphi(x_i,\theta))$ is the loss associated to the i-th sample.

Batch-size $m=1$. We recall that in this case, the stochastic gradient descent (3.3) corresponds to the abstract stochastic gradient descent with $\xi = i \sim \mathrm{Unif}(\{1,\dots,n\})$ and $g(\theta,i)=\nabla\ell_i(\theta)$. If the function $\ell_i(\theta)$ is convex, L_i-smooth, then Assumption 4.18 is satisfied with $M=\max(L_1,\dots,L_n)$. Assume further that the function f is μ-strongly convex—this needs to be proved by independent arguments—, then we obtain that for SGD with a fixed stepsize $\gamma \leqslant 1/(2\max(L_1,\dots,L_n))$, we have

$$\mathbb{E}\|\theta_k-\theta_*\|^2 \leqslant (1-\gamma\mu)^k\,\|\theta_0-\theta_*\|^2+\frac{2\gamma\sigma^2}{\mu}\,.$$

Here $\sigma^2=\frac{1}{n}\sum_{i=1}^{n}\|\nabla\ell_i(\theta_*)\|^2$.

If $\sigma^2=0$, then for all i, $\nabla\ell_i(\theta_*)=0$. This is the so-called *interpolation regime* where the global minimizer θ_* is actually a critical point for each one of the ℓ_i. This happens, for instance, when interpolating perfectly the data under the square loss. In this case, the stochastic gradient descent converges converges linearly to the optimum.

Outside of the interpolation regime, $\sigma^2>0$ and thus we only have convergence to a limited precision when using a fixed stepsize. Using variable stepsizes can allow to obtain any precision, although at the slower error rate $O(1/k)$. As the statistical learning task may be limited by the approximation and estimation errors, a limited precision in minimizing the optimization error can be sufficient for practical purposes. However, for the minimization of the empirical risk, we will show that a variation of the stochastic gradient descent can achieve linear convergence even outside of the interpolation regime, see Section 6.

General batch-size $m \geqslant 1$. We now analyze the effect of using larger batch-sizes. We recall that in this case, the stochastic gradient descent (3.2) corresponds to the abstract stochastic gradient descent with $\xi = S$ a uniformly random subset of size m of $\{1, \dots, n\}$ and $g(\theta, S) = \frac{1}{m}\sum_{i \in S} \nabla \ell_i(\theta)$.

The effect of using a larger batch-size can be measured in the variance σ^2 of the stochastic gradients at optimum. Let us thus denote

$$\sigma_m^2 = \mathbb{E}\|g(\theta_*, S)\|^2 = \mathbb{E}\left\| \frac{1}{m} \sum_{i \in S} \nabla \ell_i(\theta_*) \right\|^2 .$$

We would like to compare this quantity for a general m to the same quantity for $m = 1$: $\sigma_1^2 = \frac{1}{n}\sum_{i=1}^n \|\nabla \ell_i(\theta_*)\|^2$. We compute

$$\begin{aligned} \sigma_m^2 &= \frac{1}{m^2}\mathbb{E}\left\| \sum_{i=1}^n \mathbb{1}_{\{i \in S\}} \nabla \ell_i(\theta_*) \right\|^2 \\ &= \frac{1}{m^2} \sum_{i,j=1}^n \mathbb{P}(i \in S, j \in S) \langle \nabla \ell_i(\theta_*), \nabla \ell_j(\theta_*) \rangle \\ &= \frac{1}{m^2} \left[\sum_{i=1}^n \mathbb{P}(i \in S) \|\nabla \ell_i(\theta_*)\|^2 + \sum_{i \neq j = 1}^n \mathbb{P}(i \in S, j \in S) \langle \nabla \ell_i(\theta_*), \nabla \ell_j(\theta_*) \rangle \right] \end{aligned}$$

We have $\mathbb{P}(i \in S) = \frac{m}{n}$ and for $i \neq j$,

$$\mathbb{P}(i \in S, j \in S) = \frac{\binom{n-2}{m-2}}{\binom{n}{m}} = \frac{m(m-1)}{n(n-1)} .$$

This gives

$$\sigma_m^2 = \frac{1}{mn} \left[\sum_{i=1}^n \|\nabla \ell_i(\theta_*)\|^2 + \frac{m-1}{n-1} \sum_{i \neq j = 1}^n \langle \nabla \ell_i(\theta_*), \nabla \ell_j(\theta_*) \rangle \right] .$$

Further,

$$\begin{aligned}\sum_{i\neq j=1}^{n} \langle \nabla \ell_i(\theta_*), \nabla \ell_j(\theta_*)\rangle &= \sum_{i,j=1}^{n} \langle \nabla \ell_i(\theta_*), \nabla \ell_j(\theta_*)\rangle - \sum_{i=1}^{n} \|\nabla \ell_i(\theta_*)\|^2 \\ &= \left\| \sum_{i=1}^{n} \nabla \ell_i(\theta_*) \right\|^2 - \sum_{i=1}^{n} \|\nabla \ell_i(\theta_*)\|^2 \\ &= -\sum_{i=1}^{n} \|\nabla \ell_i(\theta_*)\|^2 .\end{aligned}$$

We thus obtain

$$\sigma_m^2 = \frac{1}{m}\left(1 - \frac{m-1}{n-1}\right)\sigma_1^2 .$$

In the small batch-size regime $m \ll n$, we have $\sigma_m^2 \approx \frac{\sigma_1^2}{m}$: by averaging m stochastic gradients, the variance of stochastic gradients is divided by m. This variance reduction helps to obtain a better asymptotic performance, without reducing the stepsize γ. In the large batch-size regime $m = \Theta(n)$, then there is an extra variance reduction due to the fact that the sampling becomes exhaustive, down to the limiting case $\sigma_n^2 = 0$. This recovers the linear convergence rate of the deterministic gradient descent.

Is using larger batch-sizes $m > 1$ worth it? The answer is not clear-cut. On the one hand, using larger batch-sizes reduces the variance of the stochastic gradients, which helps to obtain a better asymptotic performance. On the other hand, using larger batch-sizes increases the computational cost of each iteration, as it requires to compute the gradients of m samples. Recall Corollary 4.20. Stochastic gradient descent with one sample requires

$$k \geqslant 2\left(\log \frac{2\|\theta_0 - \theta_*\|^2}{\varepsilon}\right)\left(\frac{M}{\mu} + \frac{2\sigma_1^2}{\mu^2\varepsilon}\right)$$

iterations to reach an error ε. Meanwhile, stochastic gradient descent with m samples requires

$$\begin{aligned}k &\geqslant 2\left(\log \frac{2\|\theta_0 - \theta_*\|^2}{\varepsilon}\right)\left(\frac{M}{\mu} + \frac{2\sigma_m^2}{\mu^2\varepsilon}\right) \\ &\approx 2\left(\log \frac{2\|\theta_0 - \theta_*\|^2}{\varepsilon}\right)\left(\frac{M}{\mu} + \frac{2\sigma_1^2}{m\mu^2\varepsilon}\right)\end{aligned} \tag{4.3}$$

For large ε, the first terms dominate and thus the iteration complexity is the same in both cases. As a consequence, it is recommended to use batch-sizes as small as possible to reduce the per-iteration computational cost. For small ε, the second terms dominate, thus using a batch-size of m allows to divide the iteration complexity by m. As a first approximation, the per-iteration complexity of using batch-sizes

m is m times larger than using batch-sizes 1. As a consequence, the tradeoff is not obvious. However, in practice, we can take advantage of parallel computing to reduce the compute time of a stochastic gradient with batch-size m below the compute time of m stochastic gradients with batch-size 1. Batch-sizes of size $m = 32, 64, 128$ are often used in practice.

Remark 4.22. In the discussion above, we have used that Assumption 4.18 is satisfied with $M = L_{\max} := \max(L_1, \ldots, L_n)$. However, if f is L-smooth for some constant $L < L_{\max}$, then the bound on M can be improved. See Exercise 4.23 for details.

4.5 Application for single-pass stochastic gradient descent on the expected risk

In Section 3.4, we have seen that the single-pass stochastic gradient descent

$$\theta_{k+1} = \theta_k - \gamma_k \nabla \ell_{k+1}(\theta_k)\,, \qquad k = 0, \ldots, n-1\,.$$

can be interpreted as a stochastic gradient descent directly on the generalization error

$$f(\theta) = \mathbb{E}\left[\ell(y, \varphi(x,\theta))\right]\,, \qquad (x,y) \sim Q\,.$$

If we assume that the stochastic function $\theta \mapsto \ell(y, \varphi(x,\theta))$, $(x,y) \sim Q$ is almost surely convex, M-smooth, then we can control directly the generalization error. For a fixed-stepsize $\gamma_k = \gamma \leqslant 1/(2M)$,

$$\mathbb{E}f(\theta_k) - f(\theta_*) \leqslant \frac{M}{2}\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \frac{M}{2}\left((1-\gamma\mu)^k\|\theta_0 - \theta_*\|^2 + \frac{2\gamma\sigma^2}{\mu}\right)\,.$$

Further, for a variable stepsize $\gamma_k = \Theta(1/k)$ as prescribed in Theorem 4.21, we have

$$\mathbb{E}f(\theta_k) - f(\theta_*) = O\left(\frac{1}{k}\right)\,.$$

In particular, after using all of the n samples

$$\mathbb{E}f(\theta_n) - f(\theta_*) = O\left(\frac{1}{n}\right)\,.$$

This result controls the full estimation error, and not only the optimization error. While the rate $O(1/n)$ seems slow from the perspective of the optimization error—we could wish for a linear convergence of the optimization error—, in fact the rate $O(1/n)$ is optimal for the estimation error (see Exercise 4.25). As a consequence, a single pass on the data samples is sufficient to get optimal statistical rates.

4.6 Application for coordinate gradient descent

Consider a μ-strongly convex and L-smooth function $F : \mathbb{R}^p \to \mathbb{R}$ on which we perform a coordinate gradient descent: for all $k \in \mathbb{N}$, sample j_{k+1} uniformly in $\{1, \dots, p\}$ (independently of the past) and compute θ_{k+1} such that:

$$\begin{aligned} \theta_{k+1}(j_{k+1}) &= \theta_k(j_{k+1}) - \gamma_k \partial_{j_{k+1}} F(\theta_k)\,, \\ \theta_{k+1}(j) &= \theta_k(j) \qquad \text{for all } j \neq j_{k+1}\,. \end{aligned}$$

As we have seen in Section 3.5, coordinate gradient descent corresponds to $\xi = j \sim \mathrm{Unif}(\{1, \dots, p\})$, $g(\theta, j) = \partial_j F(\theta) e_j$ and $f(\theta) = \frac{1}{p} F(\theta)$. Assumption 4.17 is satified, with the subtlety that f is (μ/p)-strongly convex, and Assumption 4.18 is satisfied with $M = L$. Note further that

$$\sigma^2 = \frac{1}{p} \sum_{j=1}^{p} \|\partial_j F(\theta_*) e_j\|^2 = \frac{1}{p} \sum_{j=1}^{p} (\partial_j F(\theta_*))^2 = 0\,.$$

Thus Theorem 4.19 gives that for a fixed step-size policy $\gamma_k = \gamma \leqslant 1/(2L)$,

$$\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \left(1 - \gamma \frac{\mu}{p}\right)^k \|\theta_0 - \theta_*\|^2\,.$$

As there is no variance in the stochastic gradients at optimum, the stochastic gradient descent converges to the optimum, at a linear rate. There is no interest to reduce stepsizes in this case, and thus one can use $\gamma = 1/(2L)$, for which

$$\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \left(1 - \frac{\mu}{2Lp}\right)^k \|\theta_0 - \theta_*\|^2\,.$$

By Corollary 4.20, the iteration complexity to reach an error ε is

$$k \geqslant 2 \left(\log \frac{2\|\theta_0 - \theta_*\|^2}{\varepsilon}\right) \frac{L}{\mu} p\,.$$

Compare with the iteration complexity of plain gradient descent, given in Corollary 4.16: up to constants, the iteration complexity of coordinate gradient descent is p times larger than the iteration complexity of plain gradient descent. This is coherent as the per-iteration complexity of coordinate gradient descent is p times smaller than the per-iteration complexity of plain gradient descent.

As a consequence, the comparison of the number of computation of partial derivatives does not give a clear choice between coordinate gradient descent and plain gradient descent. As for the choice of batch-sizes, the organization of the computations needs to be taken into account in order to make the comparison. The use of factorized and parallel computations advocates for plain gradient descent, while the potential saturation of the computer memory advocates for coordinate gradient descent. As for batch computations, the optimal tradeoff can be to use block coordinate gradient descent methods, that update only a randomly sampled subset of the coordinates at each iteration.

4.7 Exercises

Exercise 4.23. We set ourselves in the setting on Section 4.4. The objective function $f(\theta) = \frac{1}{n}\sum_{i=1}^n \ell_i(\theta)$ is a finite sum of $L_{\max}$-smooth and convex functions ℓ_i. We assume that f is μ-strongly convex and L-smooth. Note that one can always assume that $L \leqslant L_{\max}$.

Set m the batch-size of the stochastic gradient descent and let $g(\theta, S) = \frac{1}{m}\sum_{i\in S}\nabla\ell_i(\theta)$ denote the associated stochastic gradient, where S is a uniformly random subset of size m of $\{1,\dots,n\}$.

1. Show that for all $\theta, \theta' \in \mathbb{R}^p$,

$$\begin{aligned}&\mathbb{E}\left[\|g(\theta,S)-g(\theta',S)\|^2\right]\\ &\quad\leqslant \frac{1}{mn}\sum_{i=1}^n \|\nabla f_i(\theta)-\nabla f_i(\theta')\|^2\\ &\qquad + \frac{m-1}{mn(n-1)}\sum_{i\neq j=1}^n \langle\nabla\ell_i(\theta)-\nabla\ell_i(\theta'), \nabla\ell_j(\theta)-\nabla\ell_j(\theta')\rangle\,.\end{aligned}$$

2. Show that for all $\theta, \theta' \in \mathbb{R}^p$,

$$\begin{aligned}&\mathbb{E}\left[\|g(\theta,S)-g(\theta',S)\|^2\right]\\ &\quad\leqslant \left[\frac{1}{m}\left(1-\frac{m-1}{n-1}\right)L_{\max}+\frac{(m-1)n}{m(n-1)}L\right]\left\langle\theta-\theta', \nabla f(\theta)-\nabla f(\theta')\right\rangle\,.\end{aligned}$$

3. Using this result, improve the complexity bound of Eq. (4.3). What is the consequence of this improved bound on the discussion of the choice of the batch-size?

Section 4.3 analyses the performance of stochastic gradient descent by discussing the behavior of upper bounds on some precision criteria, such as the expected squared distance to optimum or the suboptimality gap. However, it could be argued that these bounds are not tight, and thus that they do not reflect the actual performance of stochastic gradient descent. A more complete analysis would require to prove lower bounds on the same precision criteria; ideally, these lower bounds would match the upper bounds up to multiplicative constants. While upper bounds are often proved for all problem instances, lower bounds are often proved only for some specific problem instances.

Exercise 4.24. Consider coordinate gradient descent on a μ-strongly convex and L-smooth function $F : \mathbb{R}^p \to \mathbb{R}$, with stepsize $\gamma = 1/(2L)$. Recall from Sec. 4.6 that

$$\mathbb{E}\|\theta_k-\theta_*\|^2 \leqslant \left(1-\frac{\mu}{2Lp}\right)^k \|\theta_0-\theta_*\|^2\,.$$

We seek to prove a corresponding lower bound. Consider the function $F : \mathbb{R}^p \to \mathbb{R}$ defined as

$$F(\theta) = \frac{\mu}{2}\theta(1)^2 + \frac{L}{2}\sum_{j=2}^{p}\theta(j)^2 \,.$$

1. Show that F is L-smooth and μ-strongly convex. Compute its unique minimizer θ_*.

We initialize coordinate gradient descent at $\theta_0 = (1, 0, \dots, 0)$.

2. Show that for $k \geqslant 0$, for all $j = 2, \dots, p$, $\theta_k(j) = 0$.

3. Show that for $k \geqslant 0$, $\mathbb{E}\left[\theta_k(1)^2\right] = \left(1 - \frac{\mu}{2Lp}\right)^k$.

4. Conclude that for all $k \geqslant 0$,

$$\mathbb{E}\|\theta_k - \theta_*\|^2 = \left(1 - \frac{\mu}{2Lp}\right)^k \|\theta_0 - \theta_*\|^2 \,.$$

Beyond proving that some upper bound on the performance of an algorithm is tight, it is stronger to prove lower bounds that are valid for all algorithms (in a certain class). If such a general lower bound matches the upper bound of a specific algorithm, then this algorithm is said to be optimal.

Exercise 4.25 (tightness and statistical optimality)**.** The goal of this exercise is to explore whether the bounds proven in this section are tight and statistically optimal.

Let ξ be a random variable with some distribution $\mathcal{Q}$ with a finite second moment and define $f : \mathbb{R} \to \mathbb{R}$,

$$f(\theta) = \frac{1}{2}\mathbb{E}\left(\xi - \theta\right)^2 \,.$$

1. Compute the unique minimizer θ_* of f and $f(\theta_*)$.

2. Express a stochastic gradient descent on f that does not have a direct access to $\mathcal{Q}$ but only to i.i.d. samples $\xi_1, \xi_2, \dots \sim \mathcal{Q}$.

3. We consider the stochastic gradient descent with constant stepsize γ.

(a) Using Theorem 4.19, bound $\mathbb{E}(\theta_k - \theta_*)^2$.

(b) Compute $\mathbb{E}(\theta_k - \theta_*)^2$ exactly and compare with the bound obtained in the previous question.

4. We consider the stochastic gradient descent with variable stepsize $\gamma_k = \beta/(k_0+k)$.

(a) Using Theorem 4.21, bound $\mathbb{E}(\theta_k - \theta_*)^2$.

(b) Assume $\beta = 1$ and $k_0 = 1$. Express θ_k as a function of $\xi_1, \dots, \xi_k$. Compute $\mathbb{E}(\theta_k - \theta_*)^2$.

The empirical average is an optimal (minimax) estimator of the mean; stochastic gradient descent is said to be statistically optimal on this problem as its performance differs only by a multiplicative constant.

This exercise motivates the use of decaying stepsizes $\gamma = \Theta(1/k)$ and that it is hopeless to obtain a better rate than $\Theta(1/k)$ in our general setting.

5 Exercise/practical session: importance sampling

This section studies how biasing the sampling distribution of stochastic gradient descent, if done appropriately, improves its convergence rate. This section is largely inspired from the article of [Needell et al., 2014].

5.1 Stochastic gradient descent for finite sums

This section introduces the concept of importance sampling for the optimization of finite sums. Let $f : \mathbb{R}^p \to \mathbb{R}$ be decomposed as $f(\theta) = \frac{1}{n}\sum_{i=1}^n f_i(\theta)$. We assume that f is μ-strongly convex and that each f_i is continuously differentiable and L_i-smooth. As usual, we denote θ_* the global minimizer of f and $\sigma^2 = \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\theta_*)\|^2$.

1. We consider stochastic gradient descent: choose $\theta_0 \in \mathbb{R}^p$ and for all $k \in \mathbb{N}$, sample $i_{k+1} \sim \text{Unif}(\{1, \dots, n\})$ independently of the past and compute

$$\theta_{k+1} = \theta_k - \gamma \nabla f_{i_{k+1}}(\theta_k) \,.$$

Show that, for some appropriate choice of the stepsize γ to be determined, $k \geqslant 2\left(\log \frac{2\|\theta_0 - \theta_*\|^2}{\varepsilon}\right)\left(\max_j \frac{L_j}{\mu} + \frac{\sigma^2}{\mu^2 \varepsilon}\right)$ iterations are sufficient to obtain $\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \varepsilon$.

One can improve the dependence in $\max_j \frac{L_j}{\mu}$ by using importance sampling. Let $\pi = (\pi_1, \dots, \pi_n)$ denote a probability distribution on $\{1, \dots, n\}$. Choose $\theta_0 \in \mathbb{R}^p$ and for all $k \in \mathbb{N}$, sample $i_{k+1} \sim \pi$ independently of the past and compute

$$\theta_{k+1} = \theta_k - \gamma_{i_{k+1}} \nabla f_{i_{k+1}}(\theta_k) \,.$$

In the above algorithm, the stepsizes $\gamma_1, \dots, \gamma_n$ depend on the sampled function index. Finally, denote $\overline{L} = \frac{1}{n}\sum_{i=1}^n L_i$.

2. Show that, if $\gamma_j \propto \pi_j^{-1}$, the above iteration is a stochastic gradient descent in the sense of Sec. 3.3. In particular, show that stochastic gradients are unbiased.

3. In this question, we take $\gamma_i = \frac{\gamma}{n\pi_i}$ and $\pi_i = \frac{L_i}{n\overline{L}}$. Show that, for some appropriate choice of the stepsize γ to be determined, $k \geqslant 2\left(\log \frac{2\|\theta_0-\theta_*\|^2}{\varepsilon}\right)\left(\frac{\overline{L}}{\mu} + \frac{\overline{L}}{\min_i L_i}\frac{\sigma^2}{\mu^2\varepsilon}\right)$ iterations are sufficient to obtain $\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \varepsilon$.

In the iteration complexity, we have improved the dependence of the first term from the worst condition number $\max_i \frac{L_i}{\mu}$ to the average condition number $\frac{\overline{L}}{\mu}$. This can bring a potentially large improvement, especially when ε is large. However, the second term was worsened by a factor $\frac{\overline{L}}{\min_i L_i}$. This can be harmful when ε is small or σ^2 is large.

4. In this question, we modify π to $\pi_i = \frac{1}{2n}\left(1+\frac{L_i}{\overline{L}}\right)$ (partial biasing). Show that, for some appropriate choice of the stepsize γ to be determined, $k \geqslant 4\left(\log \frac{2\|\theta_0-\theta_*\|^2}{\varepsilon}\right)\left(\frac{\overline{L}}{\mu} + \frac{\sigma^2}{\mu^2\varepsilon}\right)$ iterations are sufficient to obtain $\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \varepsilon$.

Partially biasing the sampling allows to enjoy the best of both worlds.

5.2 Simulations: the least-squares case

Consider the minimization of a least-squares function of the form

$$f(\theta) = \frac{1}{2n}\sum_{i=1}^{n}\left(\langle x_i, \theta\rangle - y_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)\,, \qquad f_i(\theta) = \frac{1}{2}\left(\langle x_i, \theta\rangle - y_i\right)^2\,,$$

where $(x_1, y_1), \dots, (x_n, y_n)$ are given input-output pairs.

5. We denote $X \in \mathbb{R}^{n\times p}$ the design matrix whose rows are $x_1, \dots, x_n$. Under which condition on X is f strongly convex? If this condition holds, what is the associated strong convexity parameter?

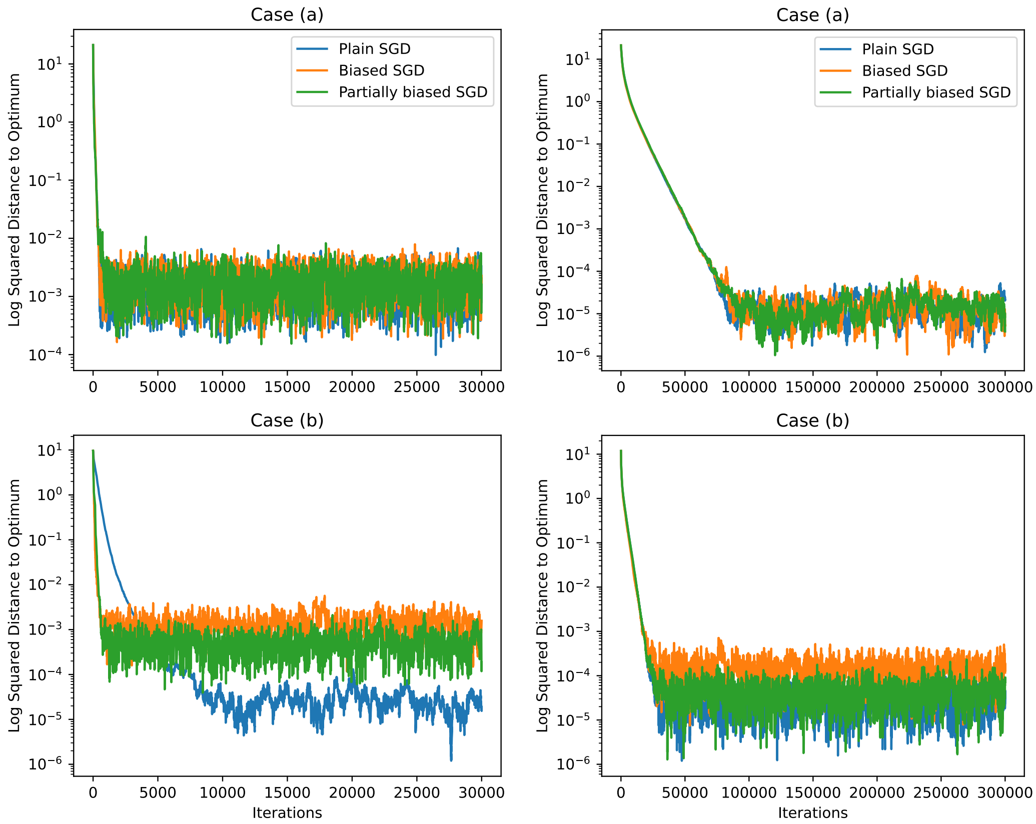
6. Give the minimal value L_i such that f_i is L_i-smooth.

7. We now run simulations with $n = 10^3$ and $p = 10$, in the two following cases:

(a) $x_1, \dots, x_n \sim_{\text{i.i.d.}} \mathcal{N}(0, I_p)$, $\theta_0 \sim \mathcal{N}(0, I_p)$, $\varepsilon_1, \dots, \varepsilon_n \sim_{i.i.d.} \mathcal{N}(0, 0.1^2)$ are all independent, and $y_i = \langle x_i, \theta_0\rangle + \varepsilon_i$,

(b) $x_1, \dots, x_{n-1} \sim_{\text{i.i.d.}} \mathcal{N}(0, I_p)$, $x_n \sim_{\text{i.i.d.}} \mathcal{N}(0, 10^2 I_p)$, $\theta_0 \sim \mathcal{N}(0, I_p)$, $\varepsilon_1, \dots, \varepsilon_n \sim_{i.i.d.} \mathcal{N}(0, 0.1^2)$ are all independent, and $y_i = \langle x_i, \theta_0\rangle + \varepsilon_i$,

For each one of these cases, generate a function f according to the specified distribution and compare the performance of plain, weighted and partially weighted stochastic gradient descent by plotting the logarithm of the distance to optimum as a function of k. For each algorithm, choose γ either (1) as large as possible, so that the algorithm remains stable or (2) so that it is the same for all algorithms. (This gives a total of $2 \times 2 = 4$ plots with three algorithms on each plot).



(1) Large stepsizes

(2) Equal stepsizes

Figure 1: Simulations of question 8. We observe that in case (a) (upper plots), as all functions have smoothness constants with the same order of magnitude, all algorithms behave similarly. In case (b), as one function has a much larger smoothness constant, differences appear. In the left plot, we observe that the two biased algorithms allow to take much larger stepsizes and thus obtain a much better initial decrease of the distance to the optimum. This is due to an improvement in the conditioning M/μ of the problem. However, they stabilize at a much larger error. This is due, of course, to the fact that the stepsize is larger. However, as the right plot shows, even with equal stepsizes, the biased algorithm stabilizes at larger errors (and in this case, the initial decrease is the same). This effect is mitigated in the case of the partially biased algorithm.

6 Variance reduction by gradient aggregation

In this section, we study the optimization of a function $f : \mathbb{R}^p \to \mathbb{R}$ which is a finite sum, i.e.,

$$f(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta)\,,$$

where $f_i : \mathbb{R}^p \to \mathbb{R}$ are continuously differentiable functions. We assume that f is μ-strongly convex and each f_i is L-smooth.

Under these assumptions, the analyses of Section 4 show that gradient descent with a constant stepsize converges at a linear rate. Further, stochastic gradient descent with a fixed stepsize converges only to a limited precision; and with a suitable variable stepsize it converges at a rate $O(1/k)$.

In this section, we show that the convergence rate of stochastic methods can be improved to a linear rate by using a *variance reduction* technique called *gradient aggregation*. Such an improvement is possible thanks to the finite sum strcutre; in general, the convergence rate of $O(1/k)$ for stochastic methods in optimal (see Exercise 4.25).

The precise gradient aggregation method considered in this section is called SAGA (Stochastic Average Gradient Aggregation) [Defazio et al., 2014]. There exists several variants such as SVRG [Johnson and Zhang, 2013] and SAG [Roux et al., 2012, Schmidt et al., 2017]. It is an iteration over $n+1$ variables $\theta, z_1, \dots, z_n \in \mathbb{R}^p$. Choose some initial values $\theta_0, z_1^{(0)}, \dots, z_n^{(0)} \in \mathbb{R}^p$. For all $k \in \mathbb{N}$, sample $i_k \sim \mathrm{Unif}(\{1, \dots, n\})$ independently of the past and compute

$$\begin{aligned}
\theta_{k+1} &= \theta_k - \gamma \left[\nabla f_{i_{k+1}}(\theta_k) + \frac{1}{n} \sum_{i=1}^{n} z_i^{(k)} - z_{i_{k+1}}^{(k)} \right]\,, \\
z_{i_{k+1}}^{(k+1)} &= \nabla f_{i_{k+1}}(\theta_k)\,, \\
z_i^{(k+1)} &= z_i^{(k)}\,, \qquad i \neq i_{k+1}\,.
\end{aligned}$$

In words, the variables $z_1, \dots, z_n$ record our last evaluations of the gradients of the functions $f_1, \dots, f_n$. We use these recordings to reduce the variance of the stochastic gradient $\nabla f_{i_{k+1}}(\theta_k)$.

As usual, we denote $\mathcal{F}_k$ the σ-algebra generated by the first k samples $i_1, \dots, i_k$. Then θ_k and $z_1^{(k)}, \dots, z_n^{(k)}$ are $\mathcal{F}_k$-measurable. The stochastic gradient $\nabla f_{i_{k+1}}(\theta_k)$ is unbiased, in the sense that $\mathbb{E}\left[\nabla f_{i_{k+1}}(\theta_k) \middle| \mathcal{F}_k\right] = \nabla f(\theta_k)$. Moreover, note that

$$\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} z_i^{(k)} - z_{i_{k+1}}^{(k)} \,\middle|\, \mathcal{F}_k \right] = \frac{1}{n} \sum_{i=1}^{n} z_i^{(k)} - \mathbb{E}\left[z_{i_{k+1}}^{(k)} \,\middle|\, \mathcal{F}_k \right] = 0\,.$$

As a consequence, the term $\frac{1}{n}\sum_{i=1}^{n} z_i^{(k)} - z_{i_{k+1}}^{(k)}$ does not add any bias to the stochastic gradient. Moreover, if the running iterate θ_k has not evolved too much, we expect

$\nabla f_{i_{k+1}}(\theta_k)$ and $z_{i_{k+1}}^{(k)}$ to be positively correlated (conditionally on $\mathcal{F}_k$). As a consequence, the (conditional) variance of the stochastic gradient should be reduced by this additional term. Indeed, even with a fixed stepsize, the SAGA algorithm converges at a linear rate, as proved by the following theorem.

Theorem 6.1. *Recall that f is μ-strongly convex and each f_i is L-smooth. Further, assume that $\gamma \leqslant \frac{1}{4L}$. Then*

$$\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \left(1 - \min\left(\frac{\gamma\mu}{2}, \frac{1}{3n}\right)\right)^k \left(\|\theta_0 - \theta_*\|^2 + 3\gamma^2 \sum_{i=1}^n \|\nabla f_i(\theta_*) - z_i^{(0)}\|^2\right) .$$

In particular, for $\gamma = \frac{1}{4L}$, we have

$$\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \left(1 - \min\left(\frac{\mu}{8L}, \frac{1}{3n}\right)\right)^k \left(\|\theta_0 - \theta_*\|^2 + \frac{3}{16L^2} \sum_{i=1}^n \|\nabla f_i(\theta_*) - z_i^{(0)}\|^2\right) .$$

As a consequence, the number of stochastic gradient iterations to have $\mathbb{E}\|\theta_k - \theta_*\|^2 \leqslant \varepsilon$ is $k = O\left(\max\left(\frac{L}{\mu}, n\right) \log \frac{1}{\varepsilon}\right)$ (omitting logarithmic terms in L and in the initialization). For comparison, in the same setting, (full batch) gradient descent requires $k = O\left(\frac{L}{\mu} \log \frac{1}{\varepsilon}\right)$ iterations, but each iteration requires the computation of the gradient of each of the n functions $f_1, \dots, f_n$. Thus the complexity, in terms of the number of gradient evaluation of the individual functions, is $O\left(n\frac{L}{\mu} \log \frac{1}{\varepsilon}\right)$. SAGA reduces the factor $n\frac{L}{\mu}$ in this complexity by the maximum of n and $\frac{L}{\mu}$. When the condition number is large, this can bring a significant improvement.

This improvement in the iteration complexity is obtained at the cost of greater memory requirements. Indeed, the SAGA algorithm requires to store the n vectors $z_1, \dots, z_n \in \mathbb{R}^d$. This can be a significant drawback when n is large. In some situations, these storage requirements can be mitigated. For instance, in the least-squares linear regression setting, $f_i(\theta) = \frac{1}{2}\left(y_i - x_i^\top \theta\right)$ and thus $\nabla f_i(\theta) = -(y_i - x_i^\top \theta)x_i$. As a consequence, it is sufficient to store the scalar quantities $y_i - x_i^\top \theta$ instead of the full vectors $\nabla f_i(\theta)$.

Proof. The initial steps of the proof ressemble the proof of Theorem 4.19.

$$\begin{aligned}\|\theta_{k+1}-\theta_*\|^2 &\leqslant \|\theta_k-\theta_*\|^2 - 2\gamma\left\langle \nabla f_{i_{k+1}}(\theta_k)+\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}, \theta_k-\theta_*\right\rangle \\ &\quad + 2\gamma^2\left(\left\|\nabla f_{i_{k+1}}(\theta_k)-\nabla f_{i_{k+1}}(\theta_*)\right\|^2 + \left\|\nabla f_{i_{k+1}}(\theta_*)+\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}\right\|^2\right) \\ &\leqslant \|\theta_k-\theta_*\|^2 - 2\gamma\left\langle \nabla f_{i_{k+1}}(\theta_k)+\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}, \theta_k-\theta_*\right\rangle \\ &\quad + 2\gamma^2\left(L\left\langle \theta_k-\theta_*, \nabla f_{i_{k+1}}(\theta_k)\right\rangle + \left\|\nabla f_{i_{k+1}}(\theta_*)+\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}\right\|^2\right).\end{aligned}$$

We now take the conditional expectation with respect to $\mathcal{F}_k$. As we have seen, the additional term $\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}$ does not add any bias to the stochastic gradient.

$$\begin{aligned}\mathbb{E}\left[\|\theta_{k+1}-\theta_*\|^2\middle|\mathcal{F}_k\right] &\leqslant \|\theta_k-\theta_*\|^2 - 2\gamma(1-L\gamma)\langle\nabla f(\theta_k),\theta_k-\theta_*\rangle \\ &\quad + 2\gamma^2\mathbb{E}\left[\left\|\nabla f_{i_{k+1}}(\theta_*)+\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}\right\|^2\middle|\mathcal{F}_k\right].\end{aligned}$$

Note that $\frac{1}{n}\sum_{i=1}^n z_i^{(k)}$ the conditional expectation of $-\nabla f_{i_{k+1}}(\theta_*) + z_{i_{k+1}}^{(k)}$ with respect to $\mathcal{F}_k$. As a consequence, $\mathbb{E}\left[\left\|\nabla f_{i_{k+1}}(\theta_*)+\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}\right\|^2\middle|\mathcal{F}_k\right]$ can be interpreted as a variance term, that we can bound by the second moment. Thus

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla f_{i_{k+1}}(\theta_*)+\frac{1}{n}\sum_{i=1}^n z_i^{(k)} - z_{i_{k+1}}^{(k)}\right\|^2\middle|\mathcal{F}_k\right] &\leqslant \mathbb{E}\left[\left\|\nabla f_{i_{k+1}}(\theta_*) - z_{i_{k+1}}^{(k)}\right\|^2\middle|\mathcal{F}_k\right] \\ &= \frac{1}{n}\sum_{i=1}^n\left\|\nabla f_i(\theta_*) - z_i^{(k)}\right\|^2 =: A_k\,.\end{aligned}$$

Then

$$\mathbb{E}\left[\|\theta_{k+1}-\theta_*\|^2\middle|\mathcal{F}_k\right] \leqslant \|\theta_k-\theta_*\|^2 - 2\gamma(1-L\gamma)\langle\nabla f(\theta_k),\theta_k-\theta_*\rangle + 2\gamma^2 A_k\,. \quad (6.1)$$

The random term A_k replaces the term σ^2 in the previous analyses. We now need to study how this quantity evolves (and decreases in expectation) in order to prove the variance reduction. Between the iterations k and $k+1$, only the term $z_{i_{k+1}}^{(k+1)}$ is updated among the $z_i^{(k)}$. Thus

$$A_{k+1}-A_k = \frac{1}{n}\left(\left\|\nabla f_{i_{k+1}}(\theta_*) - z_{i_{k+1}}^{(k+1)}\right\|^2 - \left\|\nabla f_{i_{k+1}}(\theta_*) - z_{i_{k+1}}^{(k)}\right\|^2\right)$$

where

$$\begin{aligned}\left\|\nabla f_{i_{k+1}}(\theta_*) - z_{i_{k+1}}^{(k+1)}\right\|^2 &= \left\|\nabla f_{i_{k+1}}(\theta_*) - \nabla f_{i_{k+1}}(\theta_k)\right\|^2 \\ &\leqslant L\left\langle \theta_k - \theta_*, \nabla f_{i_{k+1}}(\theta_k) - \nabla f_{i_{k+1}}(\theta_*)\right\rangle .\end{aligned}$$

Thus

$$\mathbb{E}[A_{k+1}\,|\,\mathcal{F}_k] - A_k \leqslant \frac{L}{n}\left\langle \theta_k - \theta_*, \nabla f(\theta_k)\right\rangle - \frac{A_k}{n} . \tag{6.2}$$

The inequality (6.2) needs to be combined with inequality (6.1) to obtain the decrease of both quantities. Consider the Lyapunov function $\Phi_k = \mathbb{E}\varphi_k$, where

$$\varphi_k = \|\theta_k - \theta_*\|^2 + 3n\gamma^2 A_k .$$

By combining (6.1) and (6.2),

$$\begin{aligned}\mathbb{E}\left[\varphi_{k+1}|\mathcal{F}_k\right] - \varphi_k &\leqslant \left(-2\gamma(1-L\gamma) + 3n\gamma^2\frac{L}{n}\right)\left\langle \theta_k - \theta_*, \nabla f(\theta_k)\right\rangle + (2\gamma^2 - 3\gamma^2)A_k \\ &\leqslant \gamma(-2+5L\gamma)\left\langle \theta_k - \theta_*, \nabla f(\theta_k)\right\rangle - \gamma^2 A_k .\end{aligned}$$

As $\gamma \leqslant \frac{1}{4L}$, $\gamma(-2+5L\gamma) \leqslant -\frac{\gamma}{2}$. Moreover, by strong convexity of the function f, $\langle \theta_k - \theta_*, \nabla f(\theta_k)\rangle \geqslant \mu\|\theta_k - \theta_*\|^2$. Thus

$$\mathbb{E}\left[\varphi_{k+1}|\mathcal{F}_k\right] - \varphi_k \leqslant -\frac{\gamma\mu}{2}\|\theta_k - \theta_*\|^2 - \gamma^2 A_k \leqslant -\min\left(\frac{\gamma\mu}{2}, \frac{1}{3n}\right)\varphi_k .$$

Thus

$$\begin{aligned}\mathbb{E}\|\theta_k - \theta_*\|^2 &\leqslant \Phi_k \\ &\leqslant \left(1 - \min\left(\frac{\gamma\mu}{2}, \frac{1}{3n}\right)\right)^k \Phi_0 \\ &= \left(1 - \min\left(\frac{\gamma\mu}{2}, \frac{1}{3n}\right)\right)^k \left(\|\theta_0 - \theta_*\|^2 + 3\gamma^2\sum_{i=1}^n \|\nabla f_i(\theta_*) - z_i^{(0)}\|^2\right) .\end{aligned}$$

□

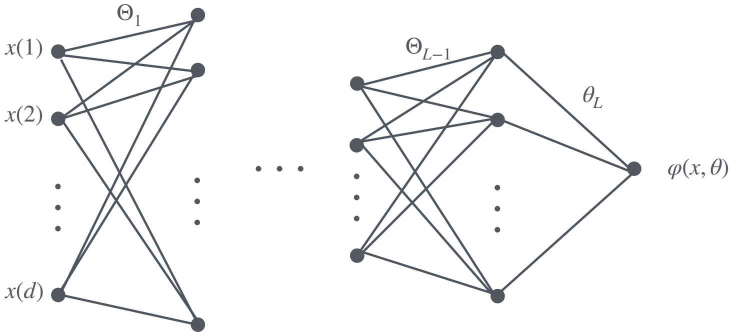
Note that we do not only prove that θ_k converges to θ_*, but also that $z_i^{(k)}$ converges to $\nabla f_i(\theta_*)$ at a linear rate.

7 Neural networks and sparse regularization

Neural networks are regression functions of the form

$$\varphi(x,\theta) = \theta_L^\top \sigma\left(\Theta_{L-1}\sigma\left(\dots\sigma\left(\Theta_1 x\right)\dots\right)\right) ,$$

where L is the number of layers in the neural network, and σ is a component-wise non-linearity.



Neural networks induce non-convex optimization problems, but in practice yield to excellent prediction functions when trained with stochastic gradient descent (or other classical optimizers like Adam), even in overparametrized regimes where the risk of overfitting should be large. In this section, we present some ongoing research attempts to explain the generalization ability of neural networks.

7.1 Neural networks with weight decay

The presentation of this section is borrowed from [Tibshirani, 2021].

7.1.1 Diagonal linear networks

Consider the lasso problem

$$\begin{aligned}\min_{\beta\in\mathbb{R}^d}.\Bigg\{&\frac{1}{2}\sum_{i=1}^n\left(y_i-\beta^\top x_i\right)^2+\lambda\|\beta\|_1\\ &=\frac{1}{2}\sum_{i=1}^n\left(y_i-\sum_{j=1}^d\beta(j)x_i(j)\right)^2+\lambda\sum_{j=1}^d|\beta(j)|\Bigg\}\end{aligned} \tag{7.1}$$

Note the following elementary remark.

Lemma 7.1. *For $\beta\in\mathbb{R}$,*

$$\min_{u,v\in\mathbb{R},\,uv=\beta}\frac{1}{2}(u^2+v^2)=|\beta|\,.$$

Proof. Consider $u,v\in\mathbb{R}$ such that $uv=\beta$. The arithmetic mean - geometric mean inequality implies that $\frac{1}{2}(u^2+v^2)\geqslant\sqrt{u^2v^2}=|uv|=|\beta|$. Moreover, the inequality is reached when $u=\sqrt{|\beta|}$ and $u=\text{sign}(\beta)\sqrt{|\beta|}$. □

As a consequence, the lasso problem (7.1) is equivalent to the minimization problem

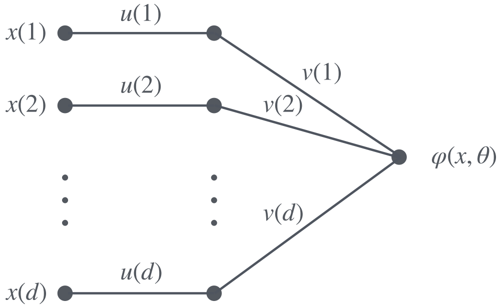
$$\begin{aligned}\min_{u,v\in\mathbb{R}^d}.\Bigg\{&\frac{1}{2}\sum_{i=1}^{n}\left(y_i-(u\circ v)^\top x_i\right)^2+\frac{\lambda}{2}\left(\|u\|_2^2+\|v\|_2^2\right)\\&=\frac{1}{2}\sum_{i=1}^{n}\left(y_i-\sum_{j=1}^{d}v(j)u(j)x_i(j)\right)^2+\frac{\lambda}{2}\sum_{j=1}^{d}\left(u(j)^2+v(j)^2\right)\Bigg\}\end{aligned}\tag{7.2}$$

where $u\circ v$ denotes the component-wise product of u and v.

Here, the equivalence between the optimization problems (7.1) and (7.2) can be interpreted in several ways:

- The two problems have the same minimal value.
- A minimizer $u, v \in \mathbb{R}^d$ of (7.2) can be transformed into a minimizer $\beta = u\circ v$ of (7.1).
- Conversely, a minimizer $\beta\in\mathbb{R}^d$ of (7.1) can be transformed into a minimizer $u, v\in\mathbb{R}^d$ of (7.2) by taking $u(j)=\sqrt{|\beta(j)|}$ and $v(j)=\mathrm{sign}(\beta(j))\sqrt{|\beta(j)|}$.

The prediction function $\varphi(x,\theta)=(v\circ u)^\top x=\sum_{j=1}^{d}v(j)u(j)x(j)$, where $\theta=(u,v)\in\mathbb{R}^d\times\mathbb{R}^d$, can be interpreted as a simple neural network, with $L=2$ layers, a linear activation function $\sigma(x)=x$ and a diagonal connection structure.



Such a neural network is called a *diagonal linear network*. In (7.2), this network is trained with weight decay, i.e. with an ℓ^2 regularization term on the network weights.

We have shown here that the optimization problem associated to the optimization of a diagonal linear network with weight decay is equivalent to a lasso optimization problem. Extrapolating largely this result, we deduce the following heuristic:

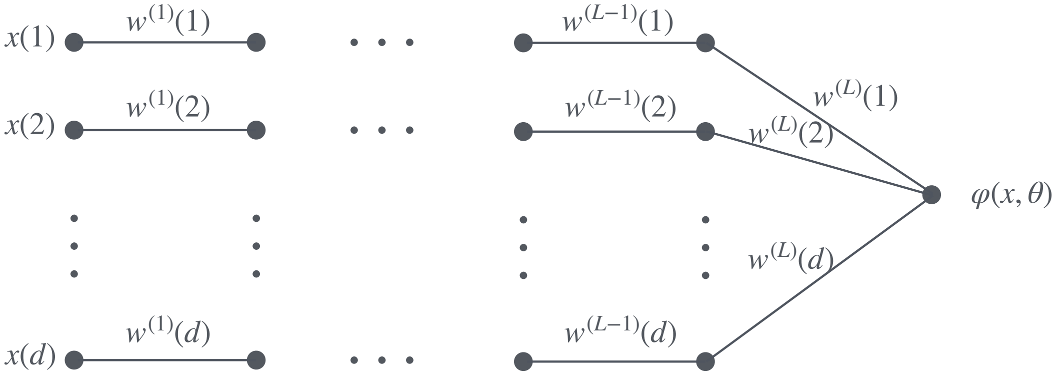
Product parametrizations induce sparsity.

7.1.2 Deeper diagonal linear networks

We now study the effect of depth of diagonal linear networks. Consider prediction functions of the form

$$\begin{aligned}\varphi(x,\theta) = (w^{(L)} \circ \cdots \circ w^{(1)})^\top x = \sum_{j=1}^{d} w^{(L)}(j) \cdots w^{(1)}(j) x(j)\,,\\ \theta = (w^{(1)}, \ldots, w^{(L)}) \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d\,.\end{aligned}$$

These prediction functions can be interpreted as diagonal linear networks of depth L.



We consider the optimization problem associated with training the neural network with weight decay

$$\begin{aligned}\min_{w^{(1)},\ldots,w^{(L)} \in \mathbb{R}^d}. \Bigg\{ \frac{1}{2} \sum_{i=1}^{n} \Big(y_i - (w^{(1)} \circ \cdots \circ w^{(L)})^\top x_i \Big)^2 \\ + \frac{\lambda}{2} \Big(\|w^{(1)}\|_2^2 + \cdots + \|w^{(L)}\|_2^2 \Big) \\ = \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} w^{(1)}(j) \cdots w^{(L)}(j) x_i(j) \right)^2 \\ + \frac{\lambda}{2} \sum_{j=1}^{d} \Big(w^{(1)}(j)^2 + \cdots + w^{(L)}(j)^2 \Big) \Bigg\}\end{aligned} \tag{7.3}$$

We make a remark analog to the one of Lemma 7.1.

Lemma 7.2. *For $\beta \in \mathbb{R}$,*

$$\min_{w^{(1)},\ldots,w^{(L)} \in \mathbb{R},\, w^{(1)} \cdots w^{(L)} = \beta} \frac{1}{2} \Big((w^{(1)})^2 + \cdots + (w^{(L)})^2 \Big) = \frac{L}{2} |\beta|^{2/L}\,.$$

Proof. Consider $w^{(1)}, \dots, w^{(L)} \in \mathbb{R}$ such that $w^{(1)} \cdots w^{(L)} = \beta$. Again, the proof stems from the arithmetic mean - geometric mean inequality:

$$\begin{aligned}\frac{1}{2}\left((w^{(1)})^2 + \cdots + (w^{(L)})^2\right) &= \frac{L}{2}\frac{(w^{(1)})^2 + \cdots + (w^{(L)})^2}{L} \\ &\geqslant \frac{L}{2}\left((w^{(1)})^2 \cdots (w^{(L)})^2\right)^{1/L} = \frac{L}{2}|\beta|^{2/L}\,.\end{aligned}$$

Moreover, the equality is obtained when $w^{(1)} = \cdots = w^{(L-1)} = |\beta|^{1/L}$ and $w^{(L)} = \mathrm{sign}(\beta)|\beta|^{1/L}$. □

This lemma shows that the optimization problem (7.3) is equivalent to the optimization problem

$$\min_{\beta \in \mathbb{R}^d}\left\{\frac{1}{2}\sum_{i=1}^{n}\left(y_i - \beta^\top x_i\right)^2 + \frac{\lambda L}{2}\sum_{j=1}^{d}|\beta(j)|^{2/L}\right\}$$

This corresponds to a sparse regularization problem with an ℓ^p regularization where $p = 2/L$. When $L > 2$, $p < 1$ and we have a non-convex sparse regularization. When $L \to \infty$, we obtain an ℓ^0 regularization. This suggests the following heuristic:

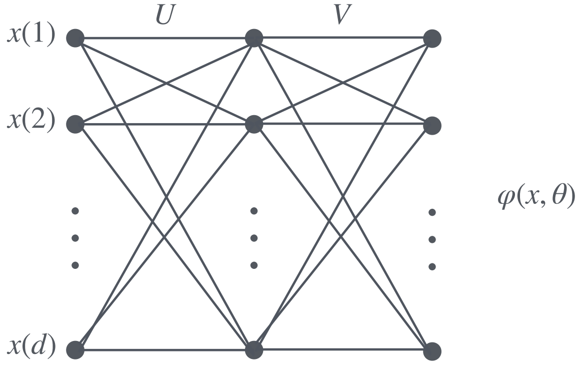
Deeper networks induce more sparsity.

7.1.3 Linear networks

We now consider the case of $L = 2$ layers, linear activation $\sigma(x) = x$ but with non-diagonal connection matrices. To make an easier analogy with the previous sections, we consider regression problems from $\mathcal{X} = \mathbb{R}^d$ to $\mathcal{Y} = \mathbb{R}^d$. We thus have n input-output pairs $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}^d$.

The linear networks that we consider are regression functions of the form

$$\varphi(x, \theta) = VUx\,, \qquad \theta = (U, V) \in \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d}\,.$$



We consider the optimization problem associated with training the neural network with weight decay

$$\min_{U,V\in\mathbb{R}^{d\times d}} \left\{ \frac{1}{2}\sum_{i=1}^{n} \|y_i - VUx_i\|^2 + \frac{\lambda}{2}\left(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2\right) \right\}, \tag{7.4}$$

where $\|.\|_{\mathrm{F}}$ denotes the Frobenius norm of a matrix.

Lemma 7.3. *For $B \in \mathbb{R}^{d\times d}$,*

$$\min_{U,V\in\mathbb{R}^{d\times d},\, VU=B} \frac{1}{2}\left(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2\right) = \|B\|_*\,,$$

where $\|.\|_*$ denotes the nuclear norm of a matrix (the sum of its singular values).

Proof. Consider the singular value decomposition $B = EDF^\top$. Here, E and F are orthogonal matrices and D is a diagonal matrix with non-negative diagonal entries that are the singular values of B. Then, by the Cauchy-Schwarz inequality,

$$\|B\|_* = \operatorname{Tr} D = \operatorname{Tr}(E^\top BF) = \operatorname{Tr}(E^\top VUF) \leqslant \|E^\top V\|_{\mathrm{F}}\|UF\|_{\mathrm{F}}\,.$$

As the Frobenius norm is orthogonally invariant, we have $\|E^\top V\|_{\mathrm{F}} = \|V\|_{\mathrm{F}}$ and $\|UF\|_{\mathrm{F}} = \|U\|_{\mathrm{F}}$. Thus

$$\|B\|_* \leqslant \|V\|_{\mathrm{F}}\|U\|_{\mathrm{F}} \leqslant \frac{1}{2}\left(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2\right)\,.$$

To prove that the inequality can be reached, consider $V = ED^{1/2}$ and $U = D^{1/2}F^\top$. In this case, $E^\top V = UF = D^{1/2}$ thus the Cauchy-Schwarz inequality is tight and both U and V have the same Frobenius norm. □

Remark 7.4. The proof of the lower bound can also be made using a matrix version of the arithmetic mean - geometric mean inequality, thus mimicking more literally the proof of Lemmas 7.1 and 7.2. From [Bhatia, 2013, Corollary IX.4.4], for any matrices $U, V \in \mathbb{R}^{d\times d}$ and any orthogonally invariant norm $\|.\|$, we have

$$\|VU\| \leqslant \frac{1}{2}\|V^\top V + UU^\top\|\,.$$

Consider $U, V \in \mathbb{R}^{d\times d}$ such that $VU = B$. As the nuclear norm is orthogonally invariant, we have

$$\|B\|_* = \|VU\|_* \leqslant \frac{1}{2}\|V^\top V + UU^\top\|_* \leqslant \frac{1}{2}\left(\|V^\top V\|_* + \|UU^\top\|_*\right)\,.$$

The nuclear norm of $V^\top V$ is the sum of its eigenvalues, which are the square of the singular values of V. As a consequence, $\|V^\top V\|_* = \|V\|_{\mathrm{F}}^2$. The same holds for $UU^\top$. Thus

$$\|B\|_* \leqslant \frac{1}{2}\left(\|V\|_{\mathrm{F}}^2 + \|U\|_{\mathrm{F}}^2\right)\,.$$

This lemma shows that the optimization problem (7.4) is equivalent to the optimization problem

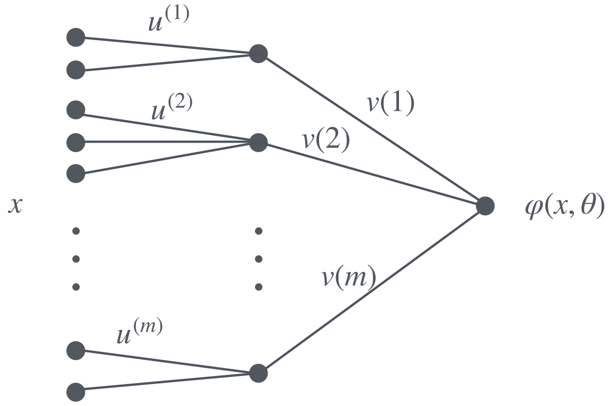
$$\min_{B\in\mathbb{R}^{d\times d}} \left\{ \frac{1}{2}\sum_{i=1}^{n} \|y_i - Bx_i\|^2 + \lambda \|B\|_* \right\} .$$

The penalization by the spectral norm is classical for optimization problems over matrices to induce low-rank solutions. This suggests the following heuristic:

In the case of product of matrices, the induced sparsity is low-rank.

7.1.4 Group diagonal linear networks

We now return to regression problems from $\mathcal{X} = \mathbb{R}^d$ to $\mathcal{Y} = \mathbb{R}$. We still consider linear activations $\sigma(x) = x$ but a more complicated connection structure.



We choose a partition of the input coordinates $\{1,\dots,d\} = G_1 \cup \dots \cup G_m$ and consider regression functions of the form

$$\varphi(x,\theta) = \sum_{k=1}^{m} v(k)(u^{(k)})^\top x_{G_k} , \quad \theta = (u^{(1)},\dots,u^{(m)},v) \in \mathbb{R}^{|G_1|} \times \dots \times \mathbb{R}^{|G_m|} \times \mathbb{R}^m .$$

We consider the optimization problem associated with training the neural network with weight decay

$$\begin{aligned} \min_{u^{(1)}\in\mathbb{R}^{|G_1|},\dots,u^{(m)}\in\mathbb{R}^{|G_m|},v\in\mathbb{R}^m} \Bigg\{ \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{k=1}^{m} v(k)(u^{(k)})^\top x_{G_k}\right)^2 \\ + \frac{\lambda}{2}\left(\sum_{k=1}^{m}\|u^{(k)}\|_2^2 + \|v\|_2^2\right)\Bigg\} . \end{aligned} \tag{7.5}$$

Lemma 7.5. *For $k \in \{1, \dots, m\}$ and $\beta \in \mathbb{R}^{|G_k|}$,*

$$\min_{u^{(k)} \in \mathbb{R}^{|G_k|},\, v \in \mathbb{R},\, vu^{(k)} = \beta} \frac{1}{2} \left(\|u^{(k)}\|_2^2 + v^2 \right) = \|\beta\|_2 \,.$$

Proof. Left as an exercise. □

This lemma shows that the optimization problem (7.5) is equivalent to the optimization problem

$$\min_{\beta \in \mathbb{R}^d}. \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \beta^\top x \right)^2 + \lambda \|\beta\|_G \right\} , \tag{7.6}$$

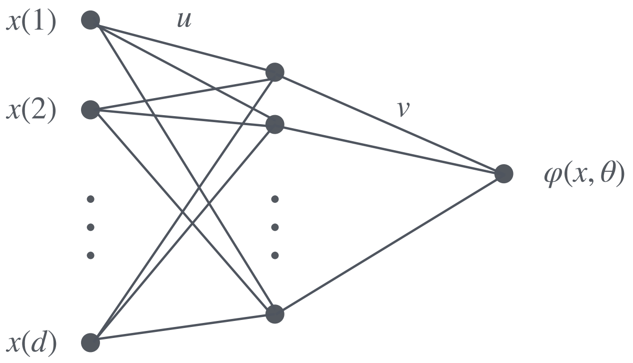
where the group ℓ_1-norm $\|\beta\|_G$ is defined as

$$\|\beta\|_G = \sum_{k=1}^{m} \|\beta_{G_k}\|_2 \,.$$

The group ℓ_1-norm is a classical regularization term to induce group-sparsity in regression problems.

7.1.5 ReLU neural networks

We now consider 2-layer neural networks with ReLU activation $\sigma(x) = \max(x, 0)$.



We consider regression functions of the form

$$\varphi(x, \theta) = \sum_{k=1}^{m} v(k) \sigma(u_k^\top x) \,, \qquad \theta = (u_1, \dots, u_m, v) \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d \times \mathbb{R}^m \,.$$

We consider the optimization problem associated with training the neural network with weight decay

$$\min_{u_1,\ldots,u_m\in\mathbb{R}^d,\, v\in\mathbb{R}^m}.\left\{\frac{1}{2}\sum_{i=1}^{n}\left(y_i-\sum_{k=1}^{m}v(k)\sigma(u_k^\top x_i)\right)^2+\frac{\lambda}{2}\left(\sum_{k=1}^{m}\|u_k\|_2^2+\|v\|_2^2\right)\right\}. \tag{7.7}$$

Here, as σ is positively homogeneous,

$$v(k)\sigma(u_k^\top x)=\operatorname{sign}(v(k))\sigma(|v(k)|u_k^\top x)\,.$$

This raises the question of computing the following.

Lemma 7.6. *For $\beta\in\mathbb{R}^d$,*

$$\min_{u\in\mathbb{R}^d,\,\mu\in\mathbb{R}_{\geqslant 0},\,\mu u=\beta}.\frac{1}{2}\left(\|u\|^2+\mu^2\right)=\|\beta\|_2\,.$$

This lemma shows that the optimization problem (7.7) is equivalent to the optimization problem

$$\min_{\beta_1,\ldots,\beta_m\in\mathbb{R}^d,\, s_1,\ldots,s_m\in\{-1,+1\}}.\left\{\frac{1}{2}\sum_{i=1}^{n}\left(y_i-\sum_{k=1}^{m}s_k\sigma(\beta_k^\top x_i)\right)^2+\lambda\sum_{k=1}^{m}\|\beta_k\|_2\right\}.$$

It is thus expected that at the optimum, many β_k will be zero, in which case the associated function $s_k\sigma(\beta_k^\top x_i)$ will be zero. This motivates the following heuristic:

ReLU neural networks induce sparsity in the number of neurons used in the representation.

Remark 7.7 (variation norm). On the space of neural networks

$$\mathcal{F}=\left\{\sum_{k=1}^{m}s_k\sigma(\beta_k^\top x)\,\middle|\, m\in\mathbb{N}, s_1,\ldots,s_m\in\{-1,+1\},\beta_1,\ldots,\beta_m\in\mathbb{R}^d\right\},$$

we can define the variation norm as follows: for $f\in\mathcal{F}$,

$$\gamma_1(f)=\min\left\{\sum_{k=1}^{m}\|\beta_k\|_2\,\middle|\, m\in\mathbb{N}, s_1,\ldots,s_m\in\{-1,+1\},\beta_1,\ldots,\beta_m\in\mathbb{R}^d \text{ with } f(x)=\sum_{k=1}^{m}s_k\sigma(\beta_k^\top x)\right\},$$

see [Bach, 2024, Section 9.3] for more details. (The proof that $\gamma_1(f)$ is indeed a norm is left as an exercise.) The equivalence shown above suggests that the variation norm describes the induced sparsity of neural networks.

Remark 7.8. The equivalence between optimization problems shown above should be taken with caution. Many of these optimization problems are non-convex, thus local descent methods are not guaranteed to converge to a global optimum. It is possible that local descent methods behave differently from one parametrization to the other, even if the two optimization problems are equivalent. For instance, local descent methods might find a global minimum in one parametrization but not in the other.

However, the heuristics derived in this section still provide some good intuition on the sparsity of neural networks when trained with local descent methods. Some more rigorous (but limited) arguments are presented in the next sections.

Finally, note that the equivalences shown in this section can be thought as alternate ways to implement sparse minimization problems. For instance, one can optimize a diagonal linear networks with weight decay (Eq. (7.2)) instead of the lasso objective (Eq. (7.1)), or the linear network of Eq. (7.5) instead of the group lasso objective in Eq. (7.6). In doing so, one trades a convex but non-smooth optimization problem for a non-convex but smooth problem; this comes with potential computational advantages [Poon and Peyré, 2023].

7.2 Implicit regularization

The previous section shows that explicit ℓ^2 regularization on the coefficients of the neural networks (weight decay) induces a sparse prior. In this section, we show that a sparse prior can actually be observed without any explicit regularization. This *implicit* regularization comes only from the training dynamics and a suitable initialization.

Consider a least-squares regression problem

$$\min_{\beta \in \mathbb{R}^d} \left\{ f(\beta) = \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \beta^\top x_i \right)^2 \right\} .$$

Using the design matrix

$$X = \begin{pmatrix} \underline{\quad x_1^\top \quad} \\ \vdots \\ \overline{\quad x_n^\top \quad} \end{pmatrix} \in \mathbb{R}^{n \times d} ,$$

the objective can be rewritten as

$$f(\beta) = \frac{1}{2} \|y - X\beta\|^2 = \frac{\|y\|^2}{2} - y^\top X \beta + \frac{1}{2} \beta^\top X^\top X \beta .$$

The objective f is quadratic with Hessian $X^\top X \in \mathbb{R}^{d \times d}$. We are interested in the case where $X^\top X$ is not full rank, which occurs in the overparametrized regime where $d > n$. In this case, there are several minimizers of f; we denote $\operatorname{Argmin} f$ the affine space of minimizers. The goal of this section is to understand which minimizer is selected, depending on the optimization algorithm. We will see that in many cases, an implicit regularization selects a regularized minimizer.

In all of this section, we analyze algorithms through their gradient flow version, i.e. their limit as the stepsize converges to 0. Going beyond the gradient flow is challenging in the case of diagonal linear networks [Even et al., 2023].

Notation. In this section, we denote Argmin the set of minimizers of an optimization problem. If the minimizer is unique, we denote it as argmin.

7.2.1 Linear parametrization

To illustrate the phenomenon of implicit regularization, we consider the simple gradient flow

$$\frac{\mathrm{d}\beta_t}{\mathrm{d}t} = -\nabla f(\beta_t)\,, \tag{7.8}$$

which is the limit of the gradient descent

$$\beta_{k+1} = \beta_k - \gamma \nabla f(\beta_k)\,, \tag{7.9}$$

as the stepsize γ converges to 0. For this gradient flow, the selected minimizer is described by the following proposition.

Proposition 7.9. *The gradient flow* (7.8)*, initialized in some $\beta_0 \in \mathbb{R}^d$, converges to β_∞ defined by*

$$\beta_\infty = \operatorname*{argmin}_{\beta_* \in \operatorname{Argmin} f} \left\{ \frac{1}{2} \|\beta_* - \beta_0\|^2 \right\} .$$

Note that $\beta_* \mapsto \frac{1}{2}\|\beta_* - \beta_0\|^2$ is a strongly convex function and $\operatorname{Argmin} f$ is an affine space thus the minimizer β_∞ is uniquely defined. In words, the gradient flow selects the minimizer of f that is closest to the initialization.

This result is mostly used with a null initialization $\beta_0 = 0$ to show that the gradient flow selects the minimizer of f with the smallest ℓ^2 norm. This can be interpreted as an implicit ℓ^2 regularization. Indeed, in this case,

$$\beta_\infty = \operatorname*{argmin}_{\beta_* \in \operatorname{Argmin} f} \left\{ \frac{1}{2} \|\beta_*\|^2 \right\} = \lim_{\lambda \to 0} \operatorname*{argmin}_{\beta \in \mathbb{R}^d} \left\{ f(\beta) + \frac{\lambda}{2} \|\beta\|^2 \right\} . \tag{7.10}$$

In words, β_∞ is the minimizer of ridge regression with an infinitesimal regularization parameter. This infinitesimal regularization is sufficient to select one of the minimizers among the infinite number of them, and thus, in some cases, to ensure generalization.

Proof. We compute

$$\frac{\mathrm{d}f(\beta_t)}{\mathrm{d}t} = \left\langle \nabla f(\beta_t), \frac{\mathrm{d}\beta_t}{\mathrm{d}t} \right\rangle = -\|\nabla f(\beta_t)\|^2 \leqslant 0\,.$$

As a consequence, $f(\beta_t)$ is non-increasing. Moreover, let β_* be any minimizer of f. Then

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\|\beta_t - \beta_*\|^2\right) = -\left\langle \beta_t - \beta_*, \nabla f(\beta_t)\right\rangle\,.$$

Denote f_* the minimum of f. We have

$$f(\beta) - f_* = \frac{1}{2}\left\langle \beta - \beta_*, X^\top X(\beta - \beta_*)\right\rangle\,,$$

and thus

$$\nabla f(\beta) = X^\top X(\beta - \beta_*)\,.$$

As a consequence,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\|\beta_t - \beta_*\|^2\right) &= -\left\langle \beta_t - \beta_*, \nabla f(\beta_t)\right\rangle \\ &= -\left\langle \beta_t - \beta_*, \nabla X^\top X(\beta_t - \beta_*)\right\rangle \\ &= -2\left(f(\beta_t) - f_*\right)\,. \end{aligned} \tag{7.11}$$

This computation has several consequences. First, as $f(\beta_t)$ is non-increasing, we have

$$\begin{aligned} t\left(f(\beta_t) - f_*\right) \leqslant \int_0^t \mathrm{d}s\left(f(\beta_s) - f_*\right) &= -\int_0^t \mathrm{d}s\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{2}\|\beta_s - \beta_*\|^2\right) \\ &= \frac{1}{4}\|\beta_0 - \beta_*\|^2 - \frac{1}{4}\|\beta_t - \beta_*\|^2 \leqslant \frac{1}{4}\|\beta_0 - \beta_*\|^2\,. \end{aligned}$$

This proves that $f(\beta_t)$ converges to f_*. Second, Eq. (7.11) shows that $\|\beta_t - \beta_*\|$ is non-increasing, and thus that the trajectory β_t is contained in some compact set. As a consequence, to show that it converges to β_∞, it is sufficient to show that β_∞ is the only possible limit for a converging subsequence. Let β_{t_k} be a subsequence converging to some limit $\beta \in \mathbb{R}^d$: $t_k \xrightarrow[k\to\infty]{} \infty$ and $\beta_{t_k} \xrightarrow[k\to\infty]{} \beta$. Then, as $f(\beta_{t_k})$ converges to f_*, we have that β is a minimizer of f.

The equality (7.11) shows that the derivative $\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\|\beta_t - \beta_*\|^2\right)$ does not depend on the minimizer β_* of f. As a consequence, it is the same for the two minimizers β_∞ and β. Integrating between 0 and t_k, we obtain

$$\frac{1}{2}\|\beta_{t_k} - \beta\|^2 - \frac{1}{2}\|\beta_0 - \beta\|^2 = \frac{1}{2}\|\beta_{t_k} - \beta_\infty\|^2 - \frac{1}{2}\|\beta_0 - \beta_\infty\|^2\,,$$

and thus

$$\frac{1}{2}\|\beta_0 - \beta\|^2 - \frac{1}{2}\|\beta_0 - \beta_\infty\|^2 = \frac{1}{2}\|\beta_{t_k} - \beta\|^2 - \frac{1}{2}\|\beta_{t_k} - \beta_\infty\|^2 \,.$$

The first term of the right-hand side converges to 0 as $k \to \infty$, and the second term is non-positive. As a consequence,

$$\frac{1}{2}\|\beta_0 - \beta\|^2 - \frac{1}{2}\|\beta_0 - \beta_\infty\|^2 \leqslant 0 \,.$$

As β_∞ is the unique minimizer of $\beta_* \mapsto \frac{1}{2}\|\beta_0 - \beta_*\|^2$ among the minimizers of f, this implies that $\beta = \beta_\infty$. This concludes the proof. □

7.2.2 Mirror parametrization

In this section, we generalize the above reasoning to a different geometry than the Euclidean one. This is done through a mirror descent / a mirror flow.

Let $D \subset \mathbb{R}^d$ be a convex open set. We say that $\Phi : D \to \mathbb{R}$ is a *mirror potential* if:

- The map Φ is differentiable and strictly convex, i.e. for all $\beta \neq \eta$, $\Phi(\beta) > \Phi(\eta) + \langle \nabla\Phi(\eta), \beta - \eta\rangle$. Equivalently, if Φ is twice differentiable, this is equivalent to the Hessian $\nabla^2\Phi(\beta)$ being positive definite for all β.
- We have $\nabla\Phi(D) = \mathbb{R}^d$.

Given a mirror potential Φ, we define the associated *Bregman divergence* $\mathcal{D}_\Phi$: for $\beta, \eta \in D$,

$$\mathcal{D}_\Phi(\beta, \eta) = \Phi(\beta) - \Phi(\eta) - \langle \nabla\Phi(\eta), \beta - \eta\rangle \,.$$

As Φ is strictly convex, we have that

$$\mathcal{D}_\Phi(\beta, \eta) \geqslant 0 \,, \qquad \text{and} \qquad \mathcal{D}_\Phi(\beta, \eta) = 0 \quad \Leftrightarrow \quad \beta = \eta \,.$$

This suggests to think of $\mathcal{D}_\Phi$ as a generalized notion of distance. However, note that we do not have the symmetry property $\mathcal{D}_\Phi(\beta, \eta) = \mathcal{D}_\Phi(\eta, \beta)$ a priori.

In order to understand how we can use this generalized notion of distance to define mirror descents, let us first reinterpret gradient descent. The iteration of gradient descent is

$$\begin{aligned}\beta_{k+1} &= \beta_k - \gamma\nabla f(\beta_k) \\ &= \operatorname*{argmin}_{\beta \in \mathbb{R}^d} \left\{ f(\beta_k) + \langle \beta - \beta_k, \nabla f(\beta_k)\rangle + \frac{1}{2\gamma}\|\beta - \beta_k\|^2 \right\} \,.\end{aligned}$$

The gradient step attempts to minimize the first-order approximation of f around β_k, $f(\beta_k) + \langle \beta - \beta_k, \nabla f(\beta_k)\rangle$, but penalizing movements that are too far away from

β_k, as the first order approximation might not be relevant there. This is the role of the quadratic term $\frac{1}{2\gamma}\|\beta - \beta_k\|^2$. For gradient descent, the distance to β_k is measured with the Euclidean distance. However, it could be that other distances are more relevant to describe the deviation of f from its first-order approximation. Mirror descent proposes to use instead a Bregman divergence.

$$\beta_{k+1} = \operatorname*{argmin}_{\beta \in \mathbb{R}^d} \left\{ f(\beta_k) + \langle \beta - \beta_k, \nabla f(\beta_k) \rangle + \frac{1}{\gamma} \mathcal{D}_\Phi(\beta, \beta_k) \right\} .$$

Note that the above optimization problem is strictly convex, thus its unique minimizer is defined by the first-order optimality condition. As $\partial_\beta \mathcal{D}(\beta, \eta) = \nabla\Phi(\beta) - \nabla\Phi(\eta)$, this gives

$$\nabla\Phi(\beta_{k+1}) = \nabla\Phi(\beta_k) - \gamma\nabla f(\beta_k) .$$

This can be interpreted as taking the gradient step in the “dual” space to which $\nabla\Phi$ maps. Note that classical gradient descent is recovered when $\Phi(\beta) = \frac{1}{2}\|\beta\|^2$, and then $\nabla\Phi(x) = x$.

In this section, we analyze the implicit regularization of mirror flow, the continuous-time limit of mirror descent. It is defined by the ordinary differential equation

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(\nabla\Phi(\beta_t)\right) &= -\nabla f(\beta_t) , \\ \nabla^2\Phi(\beta_t)\frac{\mathrm{d}\beta_t}{\mathrm{d}t} &= -\nabla f(\beta_t) . \end{aligned} \tag{7.12}$$

This can be interepreted as the gradient flow of f on the Riemannian manifold $D \subset \mathbb{R}^d$ where the metric at β is given by the Hessian $\nabla^2\Phi(\beta)$.

Proposition 7.10. *Assume that there exists a minimizer of f in D. Moreover, assume that for all $\beta \in D$, for all $C > 0$, the set $\{\eta \in D \,|\, \mathcal{D}_\Phi(\beta, \eta) \leqslant C\}$ is relatively compact*[1] *in D.*

The mirror flow (7.12), initialized in some $\beta_0 \in D$, converges to β_∞ defined by

$$\beta_\infty = \operatorname*{argmin}_{\beta_* \in D \cap \operatorname{Argmin} f} \mathcal{D}_\Phi(\beta_*, \beta_0) .$$

The function $\beta \mapsto \mathcal{D}_\Phi(\beta, \beta_0)$ is strictly convex and $D \cap \operatorname{Argmin} f$ is convex thus the minimizer defined above, if it exists, is unique. However, it is not obvious that the minimizer exists as the set $D \cap \operatorname{Argmin} f$ might not be closed. The proof shows the existence of this minimizer.

[1] The set $B = \{\eta \in D \,|\, \mathcal{D}_\Phi(\beta, \eta) \leqslant C\}$ is relatively compact in D means that for all sequence of points in B, there exists a subsequence that converges to a point in D. Note that here, B is closed in D, as the preimage of a closed set by a continuous function. Thus B being relatively compact in D is equivalent to B being compact. However, we do not need the compactness in the proof so we do not state the assumption as such.

Proof. The proof is a generalization of the proof of Proposition 7.9. We compute

$$\frac{\mathrm{d}f(\beta_t)}{\mathrm{d}t} = \left\langle \nabla f(\beta_t), \frac{\mathrm{d}\beta_t}{\mathrm{d}t} \right\rangle = -\left\langle \nabla f(\beta_t), \nabla^2\Phi(\beta_t)^{-1}\nabla f(\beta_t) \right\rangle \leqslant 0\,.$$

As a consequence, $f(\beta_t)$ is non-increasing.

Note that

$$\partial_\eta \mathcal{D}_\Phi(\beta,\eta) = -\nabla\Phi(\eta) - \nabla^2\Phi(\eta)(\beta-\eta) + \nabla\Phi(\eta) = \nabla^2\Phi(\eta)(\eta-\beta)\,.$$

By assumption, there exists a minimizer of f in D. Let β_* be any such minimizer and denote f_* the minimum of f. Then

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{D}_\Phi(\beta_*,\beta_t) &= -\left\langle \nabla_\eta \mathcal{D}_\Phi(\beta_*,\beta_t), \frac{\mathrm{d}\beta_t}{\mathrm{d}t}\right\rangle = \left\langle \nabla^2\Phi(\beta_t)(\beta_t-\beta_*), \frac{\mathrm{d}\beta_t}{\mathrm{d}t}\right\rangle \\ &= \left\langle \beta_t-\beta_*, \nabla^2\Phi(\beta_t)\frac{\mathrm{d}\beta_t}{\mathrm{d}t}\right\rangle = -\left\langle \beta_t-\beta_*, \nabla f(\beta_t)\right\rangle \\ &= -2(f(\beta_t)-f_*)\,.\end{aligned} \tag{7.13}$$

This computation has several consequences. First, as $f(\beta_t)$ is non-increasing, we have

$$\begin{aligned}t\,(f(\beta_t)-f_*) &\leqslant \int_0^t \mathrm{d}s\,(f(\beta_s)-f_*) = -\int_0^t \mathrm{d}s \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{D}_\Phi(\beta_*,\beta_s) \\ &= \frac{1}{2}\mathcal{D}_\Phi(\beta_*,\beta_0) - \frac{1}{2}\mathcal{D}_\Phi(\beta_*,\beta_t) \leqslant \frac{1}{2}\mathcal{D}_\Phi(\beta_*,\beta_0)\,.\end{aligned}$$

This proves that $f(\beta_t)$ converges to f_*. Second, Eq. (7.13) shows that $\mathcal{D}_\Phi(\beta_*,\beta_t)$ is non-increasing, and thus that the trajectory β_t is contained in some set relatively compact in D (using the related assumption). To finish the proof, it is sufficient to show that the limit of any converging subsequence of β_t minimizes $\mathcal{D}_\Phi(\beta_*,\cdot)$ over $D \cap \operatorname{Argmin} f$. We know that this minimizer, if it exists, is unique, as Φ is strictly convex.

Let β_{t_k} be a subsequence converging to some limit $\beta \in \mathbb{R}^d$: $t_k \xrightarrow[k\to\infty]{} \infty$ and $\beta_{t_k} \xrightarrow[k\to\infty]{} \beta$. Then, as $f(\beta_{t_k})$ converges to f_*, we have that β is a minimizer of f.

The equality (7.13) shows that the derivative $\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{D}_\Phi(\beta_*,\beta_t)$ does not depend on the minimizer β_* of f if D. As a consequence, it is the same for β and a generic minimizer β_* of f in D. Integrating between 0 and t_k, we obtain

$$\mathcal{D}_\Phi(\beta,\beta_{t_k}) - \mathcal{D}_\Phi(\beta,\beta_0) = \mathcal{D}_\Phi(\beta_*,\beta_{t_k}) - \mathcal{D}_\Phi(\beta_*,\beta_0)\,,$$

and thus

$$\mathcal{D}_\Phi(\beta,\beta_0) - \mathcal{D}_\Phi(\beta_*,\beta_0) = \mathcal{D}_\Phi(\beta,\beta_{t_k}) - \mathcal{D}_\Phi(\beta_*,\beta_{t_k})\,.$$

The first term of the right-hand side converges to 0 as $k \to \infty$, and the second term is non-positive. As a consequence,

$$\mathcal{D}_\Phi(\beta, \beta_0) - \mathcal{D}_\Phi(\beta_*, \beta_0) \leqslant 0 \,.$$

This proves that β minimizes $\beta_* \mapsto \mathcal{D}_\Phi(\beta_*, \beta_0)$ on $D \cap \operatorname{Argmin} f$. This concludes the proof. □

7.2.3 Diagonal linear networks

The $u \circ u$ parametrization. To start with, we consider the case of training parametrizing a space of linear functions as

$$\varphi(x, u) = (u \circ u)^\top x = \sum_{j=1}^{d} u(j)^2 x(j) \,.$$

This corresponds to a diagonal linear network with two layers as in Section 7.1.1, with the additional constraint that $u = v$. This is artificial a priori, but it is easier to analyze than $u \circ v$ parametrization; moreover, we will see that the $u \circ v$ parametrization can actually be reduced to this case.

In this section, we study how this parametrization $\beta = u \circ u$ of linear predictors influences the selected predictor. An obvious influence is that the selected predictor has nonnegative coordinates. However, we will set ourselves in the case where there are an infinite number of minimizers of f with nonnegative coordinates, and we would like to understand which one is selected by the optimization algorithm.

Our optimization algorithm is still the gradient flow of f, but this time in the variable u instead of β. This writes

$$\frac{\mathrm{d}u_t}{\mathrm{d}t} = -\nabla_u \left[f(u_t \circ u_t) \right] . \tag{7.14}$$

To analyze this gradient flow, we will interpret $\beta_t = u_t \circ u_t$ as a mirror flow. This enables to use the results of the previous section.

With $\beta_t(j) = u_t(j)^2$, we have

$$\begin{aligned}\frac{\mathrm{d}\beta_t(j)}{\mathrm{d}t} &= 2u_t(j)\frac{\mathrm{d}u_t(j)}{\mathrm{d}t} = -2u_t(j)\partial_{u(j)}\left(f(u_t \circ u_t)\right) = -4u_t(j)^2 \partial_{\beta(j)} f(\beta_t) \\ &= -4\beta(j)\partial_{\beta(j)} f(\beta_t) \,,\end{aligned}$$

or, said differently,

$$\frac{\mathrm{d}\beta_t}{\mathrm{d}t} = -4\beta_t \circ \nabla_\beta f(\beta_t) \,. \tag{7.15}$$

We would like to interpret this equation as a mirror flow, as in Eq. (7.12). It turns out that this is possible using the entropy as a mirror potential: for $\beta \in D = \mathbb{R}^d_{>0}$,

$$\begin{aligned} \Phi(\beta) &= \frac{1}{4} \sum_{j=1}^{d} \left(\beta(j) \log \beta(j) - \beta(j) \right) , \\ \nabla \Phi(\beta) &= \frac{1}{4} \begin{pmatrix} \log \beta(1) \\ \vdots \\ \log \beta(d) \end{pmatrix} , \\ \nabla^2 \Phi(\beta) &= \frac{1}{4} \operatorname{diag}(1/\beta(1), \dots, 1/\beta(d)) . \end{aligned}$$

To apply Proposition 7.10, we need to check that sublevel balls of the Bregman divergence $\mathcal{D}_\Phi$ are relatively compact in D. We compute for $\beta, \eta \in D = \mathbb{R}^d_{>0}$

$$\begin{aligned} \mathcal{D}_\Phi(\beta, \eta) &= \Phi(\beta) - \Phi(\eta) - \langle \nabla \Phi(\eta), \beta - \eta \rangle \\ &= \frac{1}{4} \sum_{j=1}^{d} \left(\beta(j) \log \frac{\beta(j)}{\eta(j)} - \beta(j) + \eta(j) \right) \\ &= \frac{1}{4} \sum_{j=1}^{d} \beta(j) \log \frac{\beta(j)}{\eta(j)} - \frac{1}{4} \|\beta\|_1 + \frac{1}{4} \|\eta\|_1 . \end{aligned}$$

Fix $\beta \in D = \mathbb{R}^d_{>0}$. Then $\mathcal{D}_\Phi(\beta, \eta) \xrightarrow[\eta \to \partial D]{} +\infty$ thus for all $C > 0$, the set $\{\eta \in D \,|\, \mathcal{D}_\Phi(\beta, \eta) \leqslant C\}$ is relatively compact in D. As a consequence, the application of Proposition 7.10 gives the following result.

Corollary 7.11. *Assume that there exists a minimizer of f in $D = \mathbb{R}^d_{>0}$. We also assume that we initialize the mirror flow (7.15) in some $\beta_0 \in D = \mathbb{R}^d_{>0}$, or equivalently, that we initialize the gradient flow (7.14) in some $u_0 \in (\mathbb{R} \backslash \{0\})^d$.*

Then, $\beta_t = u_t \circ u_t$ converges to β_∞ defined by

$$\beta_\infty = \underset{\beta_* \in D \cap \operatorname{Argmin} f}{\operatorname{argmin}} \mathcal{D}_\Phi(\beta_*, \beta_0) .$$

This result becomes interpretable in the limit of small initialization β_0. Assume that $\beta_0 = \varepsilon(\overline{\beta}(1), \dots, \overline{\beta}(d))$, where $\overline{\beta}(1), \dots, \overline{\beta}(d)$ are fixed positive constant and $\varepsilon \to 0^+$. Then

$$\begin{aligned} \mathcal{D}_\Phi(\beta_*, \beta_0) &= \frac{1}{4} \sum_{j=1}^{d} \beta_*(j) \log \frac{\beta_*(j)}{\beta_0(j)} - \frac{1}{4} \|\beta_*\|_1 + \frac{1}{4} \|\beta_0\|_1 \\ &= \frac{1}{4} \left(\log \frac{1}{\varepsilon} \right) \|\beta_*\|_1 + O(1) . \end{aligned}$$

This shows[2] that in the limit $\varepsilon \to 0^+$, β_∞ minimises $\|\beta_*\|_1$, i.e.

$$\lim_{\varepsilon\to 0^+}\lim_{t\to\infty}\beta_t \in \operatorname*{Argmin}_{\beta_*\in\overline{D}\cap\operatorname{Argmin} f} \|\beta_*\|_1 = \lim_{\lambda\to 0}\operatorname*{Argmin}_{\beta\in\overline{D}}\{f(\beta)+\lambda\|\beta\|_1\}\,. \tag{7.16}$$

Note that the objective $\|.\|_1$ is not strongly convex, thus the above minimizer might not be unique. Moreover, the minimization is now on $\overline{D}\cap\operatorname{Argmin} f = \mathbb{R}_{\geqslant 0}\cap\operatorname{Argmin} f$ as the limit $\varepsilon \to 0$ has potentially sent the minimizer to the boundary of D.

In words, the $u\circ u$ parametrization enforces a sparse implicit regularization when coupled with a small initialization.

The $u \circ v$ parametrization. We now consider diagonal linear networks of depth $L = 2$, i.e. the parametrization of linear functions as

$$\varphi(x,\theta) = (u\circ v)^\top x = \sum_{j=1}^d u(j)v(j)x(j)\,, \qquad \theta = (u,v)\in\mathbb{R}^d\times\mathbb{R}^d\,.$$

We study the gradient flow of f in the variable (u, v) instead of β. This writes

$$\begin{aligned}\frac{\mathrm{d}u_t}{\mathrm{d}t} &= -\nabla_u\left(f(u_t\circ v_t)\right)\,,\\ \frac{\mathrm{d}v_t}{\mathrm{d}t} &= -\nabla_v\left(f(u_t\circ v_t)\right)\,.\end{aligned} \tag{7.17}$$

Our first steps are a reduction to the $u \circ u$ above. Note that a gradient flow is still a gradient flow in any orthogonal transformation of the variables. We consider the variables

$$\begin{pmatrix} w\\ z\end{pmatrix} = U\begin{pmatrix} u\\ v\end{pmatrix}\,, \qquad\qquad U = \begin{pmatrix}\frac{1}{\sqrt{2}}I_d & \frac{1}{\sqrt{2}}I_d\\ \frac{1}{\sqrt{2}}I_d & -\frac{1}{\sqrt{2}}I_d\end{pmatrix}\,.$$

Note that U is orthogonal, that

$$\begin{pmatrix} u\\ v\end{pmatrix} = U^\top\begin{pmatrix} w\\ z\end{pmatrix} = \begin{pmatrix}\frac{1}{\sqrt{2}}I_d & \frac{1}{\sqrt{2}}I_d\\ \frac{1}{\sqrt{2}}I_d & -\frac{1}{\sqrt{2}}I_d\end{pmatrix}\begin{pmatrix} w\\ z\end{pmatrix}\,,$$

and thus that

$$u\circ v = \left(\frac{1}{\sqrt{2}}w + \frac{1}{\sqrt{2}}z\right)\circ\left(\frac{1}{\sqrt{2}}w - \frac{1}{\sqrt{2}}z\right) = \frac{1}{2}w\circ w - \frac{1}{2}z\circ z\,.$$

[2] Actually, this is not obvious, as the convergence of objective functions does not necessarily imply the convergence of minimizers. A complete proof would require more arguments, see [Attouch, 1996] for instance. We do not complete the argument to insist on the qualitative picture.

Thus the gradient flow (7.17) writes

$$\begin{aligned}
\frac{\mathrm{d}w_t}{\mathrm{d}t} &= -\nabla_w \left(f \left(\frac{1}{2} w_t \circ w_t - \frac{1}{2} z_t \circ z_t \right) \right) \,, \\
\frac{\mathrm{d}z_t}{\mathrm{d}t} &= -\nabla_z \left(f \left(\frac{1}{2} w_t \circ w_t - \frac{1}{2} z_t \circ z_t \right) \right) \,.
\end{aligned} \tag{7.18}$$

Define $\overline{f}(\beta^+, \beta^-) = f\left(\frac{1}{2}\beta^+ - \frac{1}{2}\beta^-\right)$. Then

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} w_t \\ z_t \end{pmatrix} = -\nabla_{(w,z)} \left[\overline{f} \left(\begin{pmatrix} w_t \\ z_t \end{pmatrix} \circ \begin{pmatrix} w_t \\ z_t \end{pmatrix} \right) \right] \,.$$

This shows (w_t, z_t) is the gradient flow of the quadratic function $\overline{f}$ parametrized as a square, as in Eq. (7.14). As a consequence, we can apply Corollary 7.11. Note that there exists a minimizer of $\overline{f}$ in $D = \mathbb{R}^{2d}_{>0}$: indeed, the minimum of f and $\overline{f}$ are the same, and for each minimizer of f we can find a minimizer of $\overline{f}$ in $D = \mathbb{R}^{2d}_{>0}$. We obtain the following result.

Corollary 7.12. *Assume that we initialize Eq.* (7.18) *in some* $(w_0, z_0) \in (\mathbb{R}\backslash\{0\})^{2d}$*, or equivalently, that we initialize Eq.* (7.17) *in some* $(u_0, v_0) \in \mathbb{R}^{2d}$ *such that for all* $j \in \{1, \dots, d\}$*,* $u_0(j) \neq \pm v_0(j)$*.*

Denote $\eta_t = (w_t, z_t) \circ (w_t, z_t)$*. Then,* η_t *converges to* η_∞ *defined by*

$$\eta_\infty = \underset{\eta_* \in D \cap \operatorname{Argmin} \overline{f}}{\operatorname{argmin}} \mathcal{D}_\Phi(\eta_*, \eta_0) \,.$$

Moreover, if $\eta_0 = \varepsilon(\overline{\eta}(1), \dots, \overline{\eta}(d))$, where $\overline{\eta}(1), \dots, \overline{\eta}(d)$ are fixed positive constant and $\varepsilon \to 0^+$. Then[3]

$$\lim_{\varepsilon \to 0^+} \lim_{t \to \infty} \eta_t \in \underset{\eta_* \in \overline{D} \cap \operatorname{Argmin} \overline{f}}{\operatorname{Argmin}} \|\eta_*\|_1 \,.$$

Note that we are not interested in understanding the limit of $\eta_t = (w_t, z_t) \circ (w_t, z_t)$ but in the limit of the linear regressor $\beta_t = u_t \circ v_t = \frac{1}{2} w_t \circ w_t - \frac{1}{2} z_t \circ z_t = \frac{1}{2}\eta_t^+ - \frac{1}{2}\eta_t^-$ where we decompose $\eta_t = (\eta_t^+, \eta_t^-)$. Consider a minimizer $\beta_* \in \mathbb{R}^d$ of f. This minimizer is the limit of the gradient flow if it corresponds to an η_* with minimal ℓ^1-norm. To the minimizer β_* corresponds an infinite number of minimizers $\eta_* = (\eta_*^+, \eta_*^-)$ of $\overline{f}$ in $\overline{D} = \mathbb{R}^{2d}_{\geqslant 0}$: $\eta_*^+ = \max(2\beta_*, 0) + \alpha$, $\eta_*^- = \min(-2\beta_*, 0) + \alpha$, for any $\alpha \in \mathbb{R}^d_{\geqslant 0}$. The one with minimum ℓ^1-norm is $\eta_* = (\eta_*^+, \eta_*^-) = (\max(2\beta_*, 0), \min(-2\beta_*, 0))$ and has ℓ^1-norm $\|\eta_*\|_1 = 2\|\beta_*\|_1$. Minimizing this quantity, we obtain that

$$\lim_{\varepsilon \to 0^+} \lim_{t \to \infty} \beta_t \in \underset{\beta_* \in \operatorname{Argmin} f}{\operatorname{Argmin}} \|\beta_*\|_1 = \lim_{\lambda \to 0} \underset{\beta \in \mathbb{R}^d}{\operatorname{Argmin}} \{f(\beta) + \lambda \|\beta\|_1\} \,.$$

[3] Again, we do not have a complete proof of this.

Compare with Eq. (7.16). The only difference is that thanks to the $u \circ v$ parametrization, there is no sign constraint. The ℓ^1-norm is minimized over the full set of minimizers of f.

Finally, compare with the result (7.10) obtained with the linear parametrization. The parametrization of diagonal linear networks changes the implicit regularization from an ℓ^2-norm to an ℓ^1-norm. This is a strong sparsity-inducing regularization, that might explain the good performance of such networks in practice.

7.3 Incremental learning

Presentation based on [Berthier, 2023, Berthier, 2025].

7.4 Exercises

Exercise 7.13. We consider the following generalization of Section 7.1.3.

We now consider regression problems from $\mathcal{X} = \mathbb{R}^d$ to $\mathcal{Y} = \mathbb{R}^p$, thus with an output dimension different from the input dimension. We thus have n input-output pairs $(x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}^p$.

Further, we consider a linear network with k neurons in the intermediate layer. This gives a regression function of the form

$$\varphi(x, \theta) = VUx\,, \qquad\qquad \theta = (U, V) \in \mathbb{R}^{k\times d} \times \mathbb{R}^{p\times k}\,.$$

Show that the optimization problem associated with training the neural network with weight decay

$$\min_{U\in\mathbb{R}^{k\times d}, V\in\mathbb{R}^{p\times k}} . \left\{ \frac{1}{2}\sum_{i=1}^n \|y_i - VUx_i\|^2 + \frac{\lambda}{2}\left(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2\right) \right\}\,,$$

is equivalent to the following rank-constrained optimization problem with nuclear norm regularization:

$$\min_{B\in\mathbb{R}^{p\times d},\,\operatorname{rank} B \leqslant k} . \left\{ \frac{1}{2}\sum_{i=1}^n \|y_i - Bx_i\|^2 + \lambda\|B\|_* \right\}\,.$$

Solution. We first show the following result: fix $B \in \mathbb{R}^{p\times d}$. Denote $r = \operatorname{rank} B$. Then

$$\min_{U\in\mathbb{R}^{k\times d}, V\in\mathbb{R}^{p\times k}\,:\, VU = B} . \left\{ \frac{1}{2}\left(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2\right) \right\} = \begin{cases} \|B\|_* & \text{if } r \leqslant k\,, \\ +\infty & \text{otherwise}\,. \end{cases} \tag{7.19}$$

Indeed, if $r > k$, it is impossible to write B as VU with $U \in \mathbb{R}^{k\times d}$ and $V \in \mathbb{R}^{p\times k}$. We now assume that $r \leqslant k$. By the singular value decomposition, we can decompose $B = EDF^\top$ where $E \in \mathbb{R}^{p\times r}$ is an isometry between $\mathbb{R}^r$ and $\operatorname{Span} B$, $F \in \mathbb{R}^{d\times r}$ is an isometry between $\mathbb{R}^r$ and $(\ker B)^\perp$ and $D \in \mathbb{R}^{r\times r}$ is a diagonal matrix whose diagonal entries are the non-zero singular values of B. Then

$$\|B\|_* = \operatorname{Tr} D = \operatorname{Tr}(E^\top B F) = \operatorname{Tr}(E^\top V U F) = \left\langle V^\top E, UF \right\rangle \leqslant \|V^\top E\|_{\mathrm{F}} \|UF\|_{\mathrm{F}}\,.$$

Moreover, $EE^\top$ is the projection onto $\operatorname{Span} B$ thus $EE^\top \preccurlyeq I_p$ and thus

$$\|V^\top E\|_{\mathrm{F}}^2 = \operatorname{Tr}(V^\top E E^\top V) \leqslant \operatorname{Tr}(V^\top V) = \|V\|_{\mathrm{F}}^2\,.$$

Similarly, $FF^\top$ is the projection onto $(\ker B)^\perp$ thus $FF^\top \preccurlyeq I_d$ and thus

$$\|UF\|_{\mathrm{F}}^2 = \operatorname{Tr}(UFF^\top U^\top) \leqslant \operatorname{Tr}(UU^\top) = \|U\|_{\mathrm{F}}^2\,.$$

As a consequence, we have

$$\|B\|_* \leqslant \|V\|_{\mathrm{F}} \|U\|_{\mathrm{F}} \leqslant \frac{1}{2}\left(\|U\|_{\mathrm{F}}^2 + \|V\|_{\mathrm{F}}^2\right)\,.$$

To show that the inequality can be reached, consider the block matrix $A = \begin{pmatrix} I_r & 0 \end{pmatrix} \in \mathbb{R}^{r\times k}$. Then we consider $V = ED^{1/2}A$ and $U = A^\top D^{1/2} F^\top$.

We thus have proved Eq. (7.19). The exercise follows immediately.

Remark 7.14. Note that the proof of the lower bound provided in Remark 7.4 generalizes directly to this exercise.

□

References

[Attouch, 1996] Attouch, H. (1996). Viscosity solutions of minimization problems. *SIAM Journal on Optimization*, 6(3):769–806.

[Bach, 2024] Bach, F. (2024). Learning theory from first principles. Lecture notes, available online at `https://www.di.ens.fr/~fbach/ltfp_book.pdf`. Accessed: Oct. 4, 2024.

[Berthier, 2023] Berthier, R. (2023). Incremental learning in diagonal linear networks. *Journal of Machine Learning Research*, 24(171):1–26.

[Berthier, 2025] Berthier, R. (2025). Diagonal linear networks and the lasso regularization path. *arXiv preprint arXiv:2509.18766*.

[Bhatia, 2013] Bhatia, R. (2013). *Matrix analysis*, volume 169. Springer Science & Business Media.

[Defazio et al., 2014] Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27.

[Even et al., 2023] Even, M., Pesme, S., Gunasekar, S., and Flammarion, N. (2023). (s) gd over diagonal linear networks: Implicit bias, large stepsizes and edge of stability. *Advances in Neural Information Processing Systems*, 36:29406–29448.

[Johnson and Zhang, 2013] Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26.

[Needell et al., 2014] Needell, D., Ward, R., and Srebro, N. (2014). Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Advances in neural information processing systems*, 27.

[Poon and Peyré, 2023] Poon, C. and Peyré, G. (2023). Smooth over-parameterized solvers for non-smooth structured optimization. *Mathematical programming*, 201(1):897–952.

[Roux et al., 2012] Roux, N., Schmidt, M., and Bach, F. (2012). A stochastic gradient method with an exponential convergence _rate for finite training sets. *Advances in neural information processing systems*, 25.

[Schmidt et al., 2017] Schmidt, M., Le Roux, N., and Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162:83–112.

[Tibshirani, 2021] Tibshirani, R. J. (2021). Equivalences between sparse models and neural networks. *Working Notes. URL https://www. stat. cmu. edu/ryantibs/papers/sparsitynn. pdf.*