Optimization for Data Science

Lecture Notes, FS 21

Bernd Gärtner, ETH
Martin Jaggi, EPFL

February 28, 2021

Contents

Chapter 2

Gradient Descent

Contents

2.1 Overview

The gradient descent algorithm (including variants such as projected or stochastic gradient descent) is the most useful workhorse for minimizing loss functions in practice. The algorithm is extremely simple and surprisingly robust in the sense that it also works well for many loss functions that are not convex. While it is easy to construct (artificial) non-convex functions on which gradient descent goes completely astray, such functions do not seem to be typical in practice; however, understanding this on a theoretical level is an open problem, and only few results exist in this direction.

The vast majority of theoretical results concerning the performance of gradient descent hold for convex functions only. In this and the following chapters, we will present some of these results, but maybe more importantly, the main ideas behind them. As it turns out, the number of ideas that we need is rather small, and typically, they are shared between different results. Our approach is therefore to fully develop each idea once, in the context of a concrete result. If the idea reappears, we will typically only discuss the changes that are necessary in order to establish a new result from this idea. In order to avoid boredom from ideas that reappear too often, we omit other results and variants that one could also get along the lines of what we discuss.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and differentiable function. We also assume that f has a global minimum $\mathbf{x}^\star$, and the goal is to find (an approximation of) $\mathbf{x}^\star$. This usually means that for a given $\varepsilon > 0$, we want to find $\mathbf{x} \in \mathbb{R}^d$ such that

$$f(\mathbf{x}) - f(\mathbf{x}^\star) < \varepsilon.$$

Notice that we are not making an attempt to get near to $\mathbf{x}^\star$ itself — there can be several minima $\mathbf{x}_1^\star \neq \mathbf{x}^\star \neq \mathbf{x}_2^\star$ with $f(\mathbf{x}_1^\star) = f(\mathbf{x}_2^\star) = f(\mathbf{x}^\star)$.

Table 2.1 gives an overview of the results that we will prove. They concern several variants of gradient descent as well as several classes of functions. The significance of each algorithm and function class will briefly be discussed when it first appears.

In Chapter 6, we will also look at gradient descent on functions that are not convex. In this case, provably small approximation error can still be obtained for some particularly well-behaved functions (we will give an example). For smooth (but not necessarily convex) functions, we gener-

	Lipschitz convex functions	smooth convex functions	strongly convex functions	smooth & strongly convex functions
gradient descent	Thm. 2.1 $\mathcal{O}(1/\varepsilon^2)$	Thm. 2.7 $\mathcal{O}(1/\varepsilon)$		Thm. 2.12 $\mathcal{O}(\log(1/\varepsilon))$
accelerated gradient descent		Thm. 2.8 $\mathcal{O}(1/\sqrt{\varepsilon})$		
projected gradient descent	Thm. 3.2 $\mathcal{O}(1/\varepsilon^2)$	Thm. 3.4 $\mathcal{O}(1/\varepsilon)$		Thm. 3.5 $\mathcal{O}(\log(1/\varepsilon))$
subgradient descent	Thm. 4.7 $\mathcal{O}(1/\varepsilon^2)$		Thm. 4.11 $\mathcal{O}(1/\varepsilon)$	
stochastic gradient descent	Thm. 5.1 $\mathcal{O}(1/\varepsilon^2)$		Thm. 5.2 $\mathcal{O}(1/\varepsilon)$	

Table 2.1: Results on gradient descent. Below each theorem, the number of steps is given which the respective variant needs on the respective function class to achieve additive approximation error at most ε.

ally cannot show convergence in error, but a (much) weaker convergence property still holds.

2.2 The algorithm

Gradient descent is a very simple iterative algorithm for finding the desired approximation $\mathbf{x}$, under suitable conditions that we will get to. It computes a sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots$ of vectors such that $\mathbf{x}_0$ is arbitrary, and for each $t \geq 0$, $\mathbf{x}_{t+1}$ is obtained from $\mathbf{x}_t$ by making a step of $\mathbf{v}_t \in \mathbb{R}^d$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_t.$$

How do we choose $\mathbf{v}_t$ in order to get closer to optimality, meaning that $f(\mathbf{x}_{t+1}) < f(\mathbf{x}_t)$?

From differentiability of f at $\mathbf{x}_t$ (Definition 1.5), we know that for $\|\mathbf{v}_t\|$

tending to 0,

$$f(\mathbf{x}_t + \mathbf{v}_t) = f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top \mathbf{v}_t + \underbrace{r(\mathbf{v}_t)}_{o(\|\mathbf{v}_t\|)} \approx f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top \mathbf{v}_t.$$

To get any decrease in function value at all, we have to choose $\mathbf{v}_t$ such that $\nabla f(\mathbf{x}_t)^\top \mathbf{v}_t < 0$. But among all steps $\mathbf{v}_t$ of the same length, we should in fact choose the one with the most negative value of $\nabla f(\mathbf{x}_t)^\top \mathbf{v}_t$, so that we maximize our decrease in function value. This is achieved when $\mathbf{v}_t$ points into the direction of the negative gradient $-\nabla f(\mathbf{x}_t)$. But as differentiabilty guarantees decrease only for small steps, we also want to control how far we go along the direction of the negative gradient.

Therefore, the step of gradient descent is defined by

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t). \tag{2.1}$$

Here, γ is a fixed *stepsize*, but it may also make sense to have γ depend on t. For now, γ is fixed. We hope that for some reasonably small integer t, in the t-th iteration we get that $f(\mathbf{x}_t) - f(\mathbf{x}^\star) < \varepsilon$; see Figure 2.1 for an example.

Now it becomes clear why we are assuming that $\mathbf{dom}(f) = \mathbb{R}^d$: The update step (2.1) may in principle take us "anywhere", so in order to get a well-defined algorithm, we want to make sure that f is defined and differentiable everywhere.

The choice of γ is critical for the performance. If γ is too small, the process might take too long, and if γ is too large, we are in danger of overshooting. It is not clear at this point whether there is a "right" stepsize.

2.3 Vanilla analysis

Let $\mathbf{x}_t$ be some iterate in the sequence (2.1). We abbreviate $\mathbf{g}_t := \nabla f(\mathbf{x}_t)$, and will relate this vector to our current direction from an optimum $\mathbf{x}_t - \mathbf{x}^\star$. By definition of gradient descent (2.1), $\mathbf{g}_t = (\mathbf{x}_t - \mathbf{x}_{t+1})/\gamma$, hence

$$\mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}^\star) = \frac{1}{\gamma} (\mathbf{x}_t - \mathbf{x}_{t+1})^\top (\mathbf{x}_t - \mathbf{x}^\star). \tag{2.2}$$

Now we apply (somewhat out of the blue, but this will clear up in the next step) the basic vector equation $2\mathbf{v}^\top \mathbf{w} = \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 - \|\mathbf{v} - \mathbf{w}\|^2$ (a.k.a. the

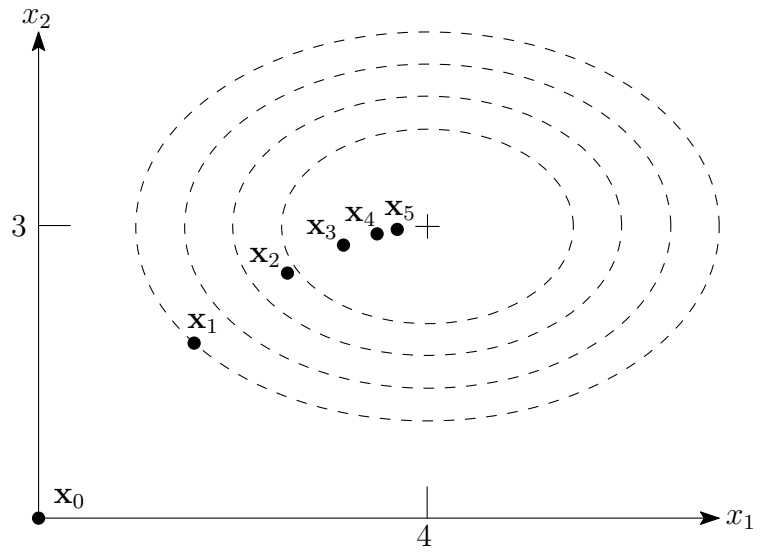

Figure 2.1: Example run of gradient descent on the quadratic function $f(x_1, x_2) = 2(x_1 - 4)^2 + 3(x_2 - 3)^2$ with global minimum $(4, 3)$; we have chosen $\mathbf{x}_0 = (0, 0), \gamma = 0.1$; dashed lines represent level sets of f (points of constant f-value)

cosine theorem) to rewrite the same expression as

$$\begin{aligned}
\mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{x}^\star) &= \frac{1}{2\gamma}\left(\|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2 + \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\right) \\
&= \frac{1}{2\gamma}\left(\gamma^2\|\mathbf{g}_t\|^2 + \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\right) \\
&= \frac{\gamma}{2}\|\mathbf{g}_t\|^2 + \frac{1}{2\gamma}\left(\|\mathbf{x}_t - \mathbf{x}^\star\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\right) \qquad (2.3)
\end{aligned}$$

Next we sum this up over the iterations t, so that the latter two terms in the bracket cancel in a telescoping sum.

$$\begin{aligned}
\sum_{t=0}^{T-1} \mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{x}^\star) &= \frac{\gamma}{2}\sum_{t=0}^{T-1}\|\mathbf{g}_t\|^2 + \frac{1}{2\gamma}\left(\|\mathbf{x}_0 - \mathbf{x}^\star\|^2 - \|\mathbf{x}_T - \mathbf{x}^\star\|^2\right) \\
&\leq \frac{\gamma}{2}\sum_{t=0}^{T-1}\|\mathbf{g}_t\|^2 + \frac{1}{2\gamma}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2 \qquad (2.4)
\end{aligned}$$

So far, we have not used any properties of the function f or its gradient $\mathbf{g}_t$, except the definition of the update step $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma \mathbf{g}_t$. Now we invoke convexity of f, or more precisely the first-order characterization of convexity (1.3) with $\mathbf{x} = \mathbf{x}_t, \mathbf{y} = \mathbf{x}^\star$:

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \mathbf{g}_t^\top (\mathbf{x}_t - \mathbf{x}^\star). \tag{2.5}$$

Hence we further obtain

$$\sum_{t=0}^{T-1} (f(\mathbf{x}_t) - f(\mathbf{x}^\star)) \leq \frac{\gamma}{2} \sum_{t=0}^{T-1} \|\mathbf{g}_t\|^2 + \frac{1}{2\gamma} \|\mathbf{x}_0 - \mathbf{x}^\star\|^2. \tag{2.6}$$

This gives us an upper bound for the *average* error $f(\mathbf{x}_t) - f(\mathbf{x}^\star)$, $t = 0, \ldots, T-1$, hence in particular for the error incurred by the iterate with the smallest function value. The last iterate is not necessarily the best one: gradient descent with fixed stepsize γ will in general also make steps that overshoot and actually increase the function value; see Exercise 15(i).

The question is of course: is this result any good? In general, the answer is no. A dependence on $\|\mathbf{x}_0 - \mathbf{x}^\star\|$ is to be expected (the further we start from $\mathbf{x}^\star$, the longer we will take); the dependence on the squared gradients $\|\mathbf{g}_t\|^2$ is more of an issue, and if we cannot control them, we cannot say much.

2.4 Lipschitz convex functions: $\mathcal{O}(1/\varepsilon^2)$ steps

Here is the cheapest "solution" to squeeze something out of the vanilla analysis (2.4): let us simply assume that all gradients of f are bounded in norm. Equivalently, such functions are Lipschitz continuous over $\mathbb{R}^d$ by Theorem 1.9. (A small subtetly here is that in the situation of real-valued functions, Theorem 1.9 is talking about the spectral norm of the $(1 \times d)$-matrix (or row vector) $\nabla f(\mathbf{x})^\top$, while below, we are talking about the Euclidean norm of the (column) vector $\nabla f(\mathbf{x})$; but these two norms are the same; see Exercise 13.)

Assuming bounded gradients rules out many interesting functions, though. For example, $f(x) = x^2$ (a supermodel in the world of convex functions) already doesn't qualify, as $\nabla f(x) = 2x$—and this is unbounded as x tends to infinity. But let's care about supermodels later.

Theorem 2.1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\mathbf{x}^\star$; furthermore, suppose that $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$ and $\|\nabla f(\mathbf{x})\| \leq B$ for all $\mathbf{x}$. Choosing the stepsize*

$$\gamma := \frac{R}{B\sqrt{T}},$$

gradient descent (2.1) yields

$$\frac{1}{T}\sum_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}^\star)) \leq \frac{RB}{\sqrt{T}}.$$

Proof. This is a simple calculation on top of (2.6): after plugging in the bounds $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$ and $\|\mathbf{g}_t\| \leq B$, we get

$$\sum_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}^\star)) \leq \frac{\gamma}{2}B^2T + \frac{1}{2\gamma}R^2,$$

so want to choose γ such that

$$q(\gamma) = \frac{\gamma}{2}B^2T + \frac{R^2}{2\gamma}$$

is minimized. Setting the derivative to zero yields the above value of γ, and $q(R/(B\sqrt{T})) = RB\sqrt{T}$. Dividing by T, the result follows. □

This means that in order to achieve $\min_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}^\star)) \leq \varepsilon$, we need

$$T \geq \frac{R^2B^2}{\varepsilon^2}$$

many iterations. This is not particularly good when it comes to concrete numbers (think of desired error $\varepsilon = 10^{-6}$ when R, B are somewhat larger). On the other hand, the number of steps does not depend on d, the dimension of the space. This is very important since we often optimize in high-dimensional spaces. Of course, R and B may depend on d, but in many relevant cases, this dependence is mild.

What happens if we don't know R and/or B? An idea is to "guess" R and B, run gradient descent with T and γ resulting from the guess, check whether the result has absolute error at most ε, and repeat with a different guess otherwise. This fails, however, since in order to compute the absolute error, we need to know $f(\mathbf{x}^\star)$ which we typically don't. But Exercise 16 asks you to show that knowing R is sufficient.

2.5 Smooth convex functions: $\mathcal{O}(1/\varepsilon)$ steps

Our workhorse in the vanilla analysis was the first-order characterization of convexity: for all $\mathbf{x}, \mathbf{y} \in \mathbf{dom}(f)$, we have

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}). \tag{2.7}$$

Next we want to look at functions for which $f(\mathbf{y})$ can be bounded *from above* by $f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})$, up to at most quadratic error. The following definition applies to all differentiable functions, convexity is not required.

Definition 2.2. *Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be a differentiable function, $X \subseteq \mathbf{dom}(f)$ convex and $L \in \mathbb{R}_+$. Function f is called* smooth *(with parameter L) over X if*

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in X. \tag{2.8}$$

If $X = \mathbf{dom}(f)$, f is simply called smooth.

Recall that (2.7) says that for any $\mathbf{x}$, the graph of f is above its tangential hyperplane at $(\mathbf{x}, f(\mathbf{x}))$. In contrast, (2.8) says that for any $\mathbf{x} \in X$, the graph of f is below a not-too-steep tangential paraboloid at $(\mathbf{x}, f(\mathbf{x}))$; see Figure 2.2.

This notion of smoothness has become standard in convex optimization, but the naming is somewhat unfortunate, since there is an (older) definition of a smooth function in mathematical analysis where it means a function that is infinitely often differentiable.

Let us discuss some cases. If $L = 0$, (2.7) and (2.8) together require that

$$f(\mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbf{dom}(f),$$

meaning that f is an affine function. A simple calculation shows that our supermodel function $f(x) = x^2$ is smooth with parameter $L = 2$:

$$\begin{aligned} f(y) = y^2 &= x^2 + 2x(y - x) + (x - y)^2 \\ &= f(x) + f'(x)(y - x) + \frac{L}{2}(x - y)^2. \end{aligned}$$

More generally, we also claim that all quadratic functions of the form $f(\mathbf{x}) = \mathbf{x}^\top Q \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$ are smooth, where Q is a $(d \times d)$ matrix, $\mathbf{b} \in \mathbb{R}^d$ and $c \in \mathbb{R}$. Because $\mathbf{x}^\top Q \mathbf{x} = \mathbf{x}^\top Q^\top \mathbf{x}$, we get that $f(\mathbf{x}) = \mathbf{x}^\top Q \mathbf{x} = \frac{1}{2}\mathbf{x}^\top (Q + Q^\top)\mathbf{x}$, where $\frac{1}{2}(Q + Q^\top)$ is symmetric. Therefore, we can assume without loss of generality that Q is symmetric, i.e., it suffices to show that quadratic functions defined by symmetric functions are smooth.

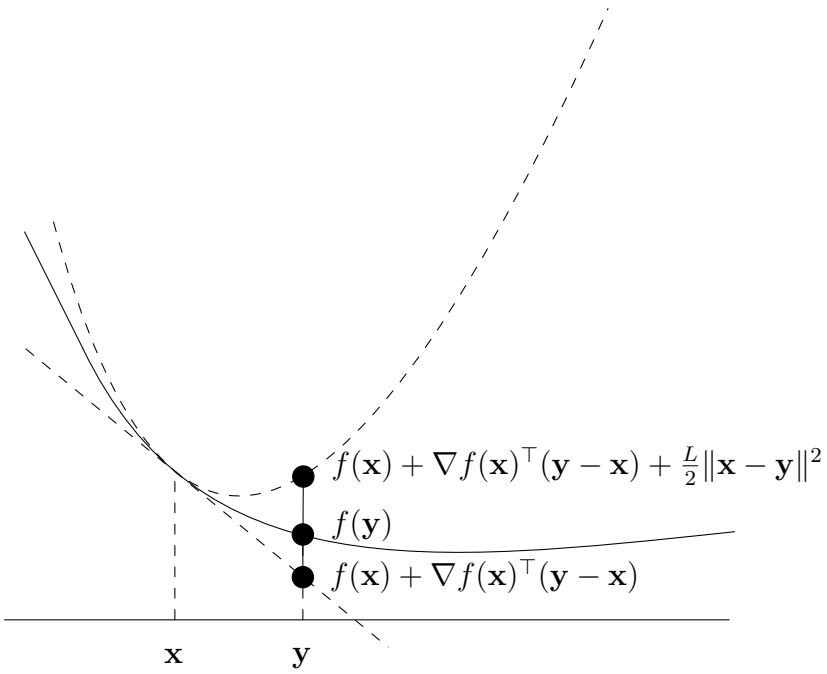

Figure 2.2: A smooth convex function

Lemma 2.3 (Exercise 14)**.** *Let $f(\mathbf{x}) = \mathbf{x}^\top Q\mathbf{x} + \mathbf{b}^\top\mathbf{x} + c$, where Q is a symmetric $(d \times d)$ matrix, $\mathbf{b} \in \mathbb{R}^d, c \in \mathbb{R}$. Then f is smooth with parameter $2\|Q\|$, where $\|Q\|$ is the spectral norm of Q (Definition 1.2).*

The (univariate) convex function $f(x) = x^4$ is not smooth (over $\mathbb{R}$): at $x = 0$, condition (2.8) reads as

$$y^4 \leq \frac{L}{2}y^2,$$

and there is obviously no L that works for all y. The function is smooth, however, over any bounded set X (Exercise 19).

In general—and this is the important message here—only functions of asymptotically at most quadratic growth can be smooth. It is tempting to believe that any such "subquadratic" function is actually smooth, but this is not true. Exercise 15(iii) provides a counterexample.

While bounded gradients are equivalent to Lipschitz continuity of f (Theorem 1.9), smoothness turns out to be equivalent to Lipschitz conti-

nuity of ∇f—if f is convex over the whole space. In general, Lipschitz continuity of ∇f implies smoothness, but not the other way around.

Lemma 2.4. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable. The following two statements are equivalent.*

(i) f is smooth with parameter L.

(ii) $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

We will derive the direction (ii)⇒(i) as Lemma 6.1 in Chapter 6 (which neither requires convexity nor domain $\mathbb{R}^d$). The other direction is a bit more involved. A proof of the equivalence can be found in the lecture slides of L. Vandenberghe, `http://www.seas.ucla.edu/~vandenbe/236C/lectures/gradient.pdf`.

The operations that we have shown to preserve convexity (Lemma 1.18) also preserve smoothness. This immediately gives us a rich collection of smooth functions.

Lemma 2.5 (Exercise 17)**.**

(i) Let $f_1, f_2, \ldots, f_m$ be smooth with parameters $L_1, L_2, \ldots, L_m$, and let $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{R}_+$. Then the function $f := \sum_{i=1}^m \lambda_i f_i$ is smooth with parameter $\sum_{i=1}^m \lambda_i L_i$ over $\mathbf{dom}(f) := \bigcap_{i=1}^m \mathbf{dom}(f_i)$.

(ii) Let $f : \mathbf{dom}(f) \to \mathbb{R}$ with $\mathbf{dom}(f) \subseteq \mathbb{R}^d$ be smooth with parameter L, and let $g : \mathbb{R}^m \to \mathbb{R}^d$ be an affine function, meaning that $g(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, for some matrix $A \in \mathbb{R}^{d \times m}$ and some vector $\mathbf{b} \in \mathbb{R}^d$. Then the function $f \circ g$ (that maps $\mathbf{x}$ to $f(A\mathbf{x} + \mathbf{b})$) is smooth with parameter $L\|A\|^2$ on $\mathbf{dom}(f \circ g) := \{\mathbf{x} \in \mathbb{R}^m : g(\mathbf{x}) \in \mathbf{dom}(f)\}$, where $\|A\|$ is the spectral norm of A (Definition 1.2).

We next show that for smooth convex functions, the vanilla analysis provides a better bound than it does under bounded gradients. In particular, we are now able to serve the supermodel $f(x) = x^2$.

We start with a preparatory lemma showing that gradient descent (with suitable stepsize γ) makes progress in function value on smooth functions in every step. We call this *sufficient decrease*, and maybe surprisingly, it does not require convexity.

Lemma 2.6. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and smooth with parameter L according to (2.8). With*

$$\gamma := \frac{1}{L},$$

gradient descent (2.1) satisfies

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2, \quad t \geq 0.$$

More specifically, this already holds if f is smooth with parameter L over the line segment connecting $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$.

Proof. We apply the smoothness condition (2.8) and the definition of gradient descent that yields $\mathbf{x}_{t+1} - \mathbf{x}_t = -\nabla f(\mathbf{x}_t)/L$. We compute

$$\begin{aligned} f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top(\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{L}{2}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2 \\ &= f(\mathbf{x}_t) - \frac{1}{L}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2 \\ &= f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2. \end{aligned}$$

□

Theorem 2.7. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\mathbf{x}^\star$; furthermore, suppose that f is smooth with parameter L according to (2.8). Choosing stepsize*

$$\gamma := \frac{1}{L},$$

gradient descent (2.1) yields

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{L}{2T}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2, \quad T > 0.$$

Proof. We apply sufficient decrease (Lemma 2.6) to bound the sum of the $\|\mathbf{g}_t\|^2 = \|\nabla f(\mathbf{x}_t)\|^2$ after step (2.6) of the vanilla analysis as follows:

$$\frac{1}{2L}\sum_{t=0}^{T-1}\|\nabla f(\mathbf{x}_t)\|^2 \leq \sum_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})) = f(\mathbf{x}_0) - f(\mathbf{x}_T). \tag{2.9}$$

With $\gamma = 1/L$, (2.6) then yields

$$\begin{aligned}\sum_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}^\star)) &\leq \frac{1}{2L}\sum_{t=0}^{T-1}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2 \\ &\leq f(\mathbf{x}_0) - f(\mathbf{x}_T) + \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2,\end{aligned}$$

equivalently

$$\sum_{t=1}^{T}(f(\mathbf{x}_t) - f(\mathbf{x}^\star)) \leq \frac{L}{2}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2. \tag{2.10}$$

Because $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t)$ for each $0 \leq t \leq T$ by Lemma 2.6, by taking the average we get that

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{1}{T}\sum_{t=1}^{T}(f(\mathbf{x}_t) - f(\mathbf{x}^\star)) \leq \frac{L}{2T}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2.$$

□

This improves over the bounds of Theorem 2.1. With $R^2 := \|\mathbf{x}_0 - \mathbf{x}^\star\|^2$, we now only need

$$T \geq \frac{R^2 L}{2\varepsilon}$$

iterations instead of R^2B^2/ε^2 to achieve absolute error at most ε.

Exercise 18 shows that we do not need to know L to obtain the same asymptotic runtime.

Interestingly, the bound in Theorem 2.7 can be improved—but not by much. Fixing L and $R = \|\mathbf{x}_0 - \mathbf{x}^\star\|$, the bound is of the form $O(1/T)$. Lee and Wright have shown that a better upper bound of $o(1/T)$ holds, but that for any fixed $\delta > 0$, a lower bound of $\Omega(1/T^{1+\delta})$ also holds [LW19].

2.6 Acceleration for smooth convex functions: $\mathcal{O}(1/\sqrt{\varepsilon})$ steps

Let's take a step back, forget about gradient descent for a moment, and just think about what we actually use the algorithm for: we are minimizing a

differentiable convex function $f : \mathbb{R}^d \to \mathbb{R}$, where we are assuming that we have acccess to the gradient vector $\nabla f(\mathbf{x})$ at any given point $\mathbf{x}$.

But is it clear that gradient descent is the best algorithm for this task? After all, it is just *some* algorithm that is using gradients to make progress locally, but there might be other (and better) such algorithms. Let us define a *first-order method* as an algorithm that only uses gradient information to minimize f. More precisely, we allow a first-order method to access f only via an oracle that is able to return values of f and ∇f at arbitrary points. Gradient descent is then just a specific first-order method.

For any class of convex functions, one can then ask a natural question: What is the best first-order method for the function class, the one that needs the smallest number of oracle calls in the worst case, as a function of the desired error ε? In particular, is there a method that asymptotically beats gradient descent?

There is an interesting history here: in 1979, Nemirovski and Yudin have shown that *every* first-order method needs in the worst case $\Omega(1/\sqrt{\varepsilon})$ steps (gradient evaluations) in order to achieve an additive error of ε on smooth functions [NY83]. Recall that we have seen an upper bound of $O(1/\varepsilon)$ for gradient descent in the previous section; in fact, this upper bound was known to Nemirovsky and Yudin already. Reformulated in the language of the previous section, there is a first-order method (gradient descent) that attains additive error $O(1/T)$ after T steps, and all first-order methods have additive error $\Omega(1/T^2)$ in the worst case.

The obvious question resulting from this was whether there actually exists a first-order method that has additive error $O(1/T^2)$ after T steps, on every smooth function. This was answered in the affirmative by Nesterov in 1983 when he proposed an algorithm that is now known as *(Nesterov's) accelerated gradient descent* [Nes83]. Nesterov's book (Sections 2.1 and 2.2) is a comprehensive source for both lower and upper bound [Nes18].

It is not easy to understand why the accelerated gradient descent algorithm is an optimal first-order method, and how Nesterov even arrived at it. A number of alternative derivations of optimal algorithms have been given by other authors, usually claiming that they provide a more natural or easier-to-grasp approach. However, each alternative approach requires some understanding of other things, and there is no well-established "simplest approach". Here, we simply throw the algorithm at the reader, without any attempt to motivate it beyond some obvious words. Then we present a short proof that the algorithm is indeed optimal.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex, differentiable, and smooth with parameter L. *Accelerated gradient descent* is the following algorithm: choose $\mathbf{z}_0 = \mathbf{y}_0 = \mathbf{x}_0$ arbitrary. For $t \geq 0$, set

$$\mathbf{y}_{t+1} := \mathbf{x}_t - \frac{1}{L}\nabla f(\mathbf{x}_t), \tag{2.11}$$

$$\mathbf{z}_{t+1} := \mathbf{z}_t - \frac{t+1}{2L}\nabla f(\mathbf{x}_t), \tag{2.12}$$

$$\mathbf{x}_{t+1} := \frac{t+1}{t+3}\mathbf{y}_{t+1} + \frac{2}{t+3}\mathbf{z}_{t+1}. \tag{2.13}$$

This means, we are performing a normal "smooth step" from $\mathbf{x}_t$ to obtain $\mathbf{y}_{t+1}$ and a more aggressive step from $\mathbf{z}_t$ to get $\mathbf{z}_{t+1}$. The next iterate $\mathbf{x}_{t+1}$ is a weighted average of $\mathbf{y}_{t+1}$ and $\mathbf{z}_{t+1}$, where we compensate for the more aggressive step by giving $\mathbf{z}_{t+1}$ a relatively low weight.

Theorem 2.8. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\mathbf{x}^\star$; furthermore, suppose that f is smooth with parameter L according to (2.8). Accelerated gradient descent (2.11), (2.12), and (2.13), yields*

$$f(\mathbf{y}_T) - f(\mathbf{x}^\star) \leq \frac{2L\left\|\mathbf{z}_0 - \mathbf{x}^\star\right\|^2}{T(T+1)}, \quad T > 0.$$

Comparing this bound with the one from Theorem 2.7, we see that the error is now indeed $O(1/T^2)$ instead of $O(1/T)$; to reach error at most ε, accelerated gradient descent therefore only needs $O(1/\sqrt{\varepsilon})$ steps instead of $O(1/\varepsilon)$.

Proof. The analysis uses a *potential function argument* [BG17]. We assign a potential $\Phi(t)$ to each time t and show that $\Phi(t+1) \leq \Phi(t)$. The potential is

$$\Phi(t) := t(t+1)\left(f(\mathbf{y}_t) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_t - \mathbf{x}^\star\right\|^2.$$

If we can show that the potential always decreases, we get

$$\underbrace{T(T+1)\left(f(\mathbf{y}_T) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_T - \mathbf{x}^\star\right\|^2}_{\Phi(T)} \leq \underbrace{2L\left\|\mathbf{z}_0 - \mathbf{x}^\star\right\|^2}_{\Phi(0)},$$

from which the statement immediately follows. For the argument, we need three well-known ingredients: (i) sufficient decrease (Lemma 2.6) for step (2.11) with $\gamma = 1/L$:

$$f(\mathbf{y}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2; \tag{2.14}$$

(ii) the vanilla analysis (Section 2.3) for step (2.12) with $\gamma = \frac{t+1}{2L}$, $\mathbf{g}_t = \nabla f(\mathbf{x}_t)$:

$$\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) = \frac{t+1}{4L}\|\mathbf{g}_t\|^2 + \frac{L}{t+1}\left(\|\mathbf{z}_t - \mathbf{x}^\star\|^2 - \|\mathbf{z}_{t+1} - \mathbf{x}^\star\|^2\right); \qquad (2.15)$$

(iii) convexity:

$$f(\mathbf{x}_t) - f(\mathbf{w}) \le \mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{w}), \quad \mathbf{w} \in \mathbb{R}^d. \qquad (2.16)$$

On top of this, we perform some simple calculations next. By definition, the potentials are

$$\begin{aligned}\Phi(t+1) &= t(t+1)\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + 2(t+1)\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_{t+1} - \mathbf{x}^\star\right\|^2 \\ \Phi(t) &= t(t+1)\left(f(\mathbf{y}_t\ \ ) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_t\ \ - \mathbf{x}^\star\right\|^2\end{aligned}$$

Now,

$$\Delta := \frac{\Phi(t+1) - \Phi(t)}{t+1}$$

can be bounded as follows.

$$\begin{aligned}\Delta &= t\left(f(\mathbf{y}_{t+1}) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + \frac{2L}{t+1}\left(\|\mathbf{z}_{t+1} - \mathbf{x}^\star\|^2 - \|\mathbf{z}_t - \mathbf{x}^\star\|^2\right) \\ &\overset{(2.15)}{=} t\left(f(\mathbf{y}_{t+1}) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + \frac{t+1}{2L}\|\mathbf{g}_t\|^2 - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\ &\overset{(2.14)}{\le} t\left(f(\mathbf{x}_t) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) - \frac{1}{2L}\|\mathbf{g}_t\|^2 - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\ &\le t\left(f(\mathbf{x}_t) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\ &\overset{(2.16)}{\le} t\mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{y}_t) + 2\mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{x}^\star) - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\ &= \mathbf{g}_t^\top((t+2)\mathbf{x}_t - t\mathbf{y}_t - 2\mathbf{z}_t) \\ &\overset{(2.13)}{=} \mathbf{g}_t^\top\mathbf{0} = 0.\end{aligned}$$

Hence, we indeed have $\Phi(t+1) \le \Phi(t)$. □

2.7 Interlude

Let us get back to the supermodel $f(x) = x^2$ (that is smooth with parameter $L = 2$, as we observed before). According to Theorem 2.7, gradient

descent (2.1) with stepsize $\gamma = 1/2$ satisfies

$$f(x_T) \leq \frac{1}{T} x_0^2. \tag{2.17}$$

Here we used that the minimizer is $x^\star = 0$. Let us check how good this bound really is. For our concrete function and concrete stepsize, (2.1) reads as

$$x_{t+1} = x_t - \frac{1}{2}\nabla f(x_t) = x_t - x_t = 0,$$

so we are always done after one step! But we will see in the next section that this is only because the function is particularly beautiful, and on top of that, we have picked the best possible smoothness parameter. To simulate a more realistic situation here, let us assume that we have not looked at the supermodel too closely and found it to be smooth with parameter $L = 4$ only (which is a suboptimal but still valid parameter). In this case, $\gamma = 1/4$ and (2.1) becomes

$$x_{t+1} = x_t - \frac{1}{4}\nabla f(x_t) = x_t - \frac{x_t}{2} = \frac{x_t}{2}.$$

So, we in fact have

$$f(x_T) = f\left(\frac{x_0}{2^T}\right) = \frac{1}{2^{2T}} x_0^2. \tag{2.18}$$

This is still vastly better than the bound of (2.17)! While (2.17) requires $T \approx x_0^2/\varepsilon$ to achieve $f(x_T) \leq \varepsilon$, (2.18) requires only

$$T \approx \frac{1}{2}\log\left(\frac{x_0^2}{\varepsilon}\right),$$

which is an exponential improvement in the number of steps.

2.8 Smooth and strongly convex functions: $\mathcal{O}(\log(1/\varepsilon))$ steps

The supermodel function $f(x) = x^2$ is not only smooth ("not too curved") but also *strongly convex* ("not too flat"). It will turn out that this is the crucial ingredient that makes gradient descent fast.

Definition 2.9. *Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be a convex and differentiable function, $X \subseteq \mathbf{dom}(f)$ convex and $\mu \in \mathbb{R}_+, \mu > 0$. Function f is called* strongly convex *(with parameter μ) over X if*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in X. \tag{2.19}$$

If $X = \mathbf{dom}(f)$, f is simply called strongly convex.

While smoothness according to (2.8) says that for any $\mathbf{x} \in X$, the graph of f is *below* a *not-too-steep* tangential paraboloid at $(\mathbf{x}, f(\mathbf{x}))$, strong convexity means that the graph of f is *above* a *not-too-flat* tangential paraboloid at $(\mathbf{x}, f(\mathbf{x}))$. The graph of a smooth *and* strongly convex function is therefore at every point wedged between two paraboloids; see Figure 2.3.

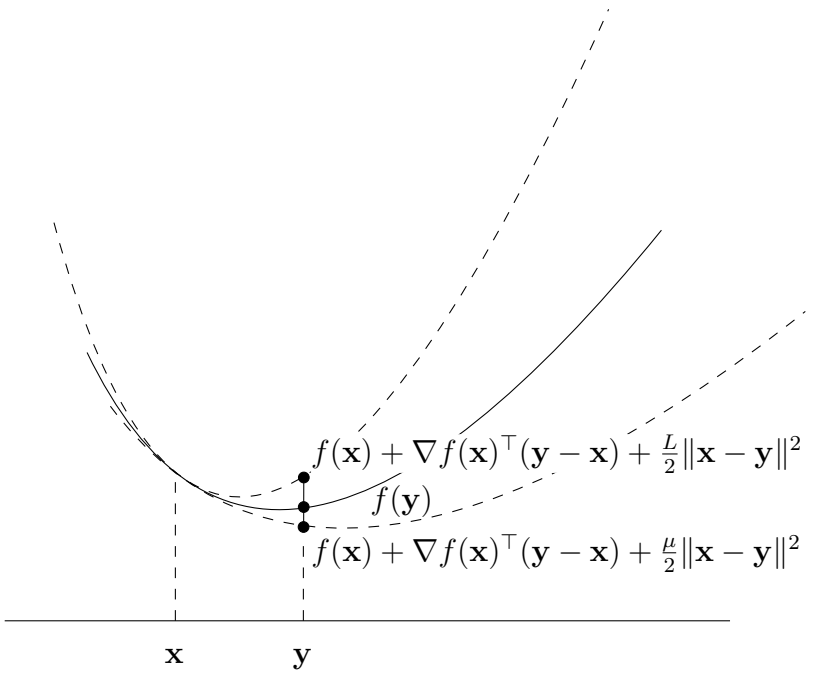

Figure 2.3: A smooth and strongly convex function

We can also interpret (2.19) as a strengthening of convexity. In the form of (2.7), convexity reads as

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathbf{dom}(f),$$

and therefore says that every convex function satisfies (2.19) with $\mu = 0$.

Lemma 2.10 (Exercise 20)**.** *If $f : \mathbb{R}^d \to \mathbb{R}$ is strongly convex with parameter $\mu > 0$, then f is strictly convex and has a unique global minimum.*

The supermodel $f(x) = x^2$ is particularly beautiful since it is both smooth and strongly convex with the same parameter $L = \mu = 2$ (going through the calculations in Exercise 14 will reveal this). We can easily characterize the class of particularly beautiful functions. These are exactly the ones whose sublevel sets are ℓ_2-balls.

Lemma 2.11 (Exercise 21)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be strongly convex with parameter $\mu > 0$ and smooth with parameter μ. Prove that f is of the form*

$$f(\mathbf{x}) = \frac{\mu}{2}\|\mathbf{x} - \mathbf{b}\|^2 + c,$$

where $\mathbf{b} \in \mathbb{R}^d, c \in \mathbb{R}$.

Once we have a unique global minimum $\mathbf{x}^\star$, we can attempt to prove that $\lim_{t\to\infty} \mathbf{x}_t = \mathbf{x}^\star$ in gradient descent. We start from the vanilla analysis (2.3) and plug in the lower bound $\mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{x}^\star) = \nabla f(\mathbf{x}_t)^\top(\mathbf{x}_t - \mathbf{x}^\star) \geq f(\mathbf{x}_t) - f(\mathbf{x}^\star) + \frac{\mu}{2}\|\mathbf{x}_t - \mathbf{x}^\star\|^2$ resulting from strong convexity. We get

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \frac{1}{2\gamma}\left(\gamma^2\|\nabla f(\mathbf{x}_t)\|^2 + \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\right) - \frac{\mu}{2}\|\mathbf{x}_t - \mathbf{x}^\star\|^2. \tag{2.20}$$

Rewriting this yields a bound on $\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2$ in terms of $\|\mathbf{x}_t - \mathbf{x}^\star\|^2$, along with some "noise" that we still need to take care of:

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq 2\gamma(f(\mathbf{x}^\star) - f(\mathbf{x}_t)) + \gamma^2\|\nabla f(\mathbf{x}_t)\|^2 + (1 - \mu\gamma)\|\mathbf{x}_t - \mathbf{x}^\star\|^2. \tag{2.21}$$

Theorem 2.12. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable. Suppose that f is smooth with parameter L according to (3.5) and strongly convex with parameter $\mu > 0$ according to (3.9). Exercise 24 asks you to prove that there is a unique global minimum $\mathbf{x}^\star$ of f. Choosing*

$$\gamma := \frac{1}{L},$$

gradient descent (2.1) with arbitrary $\mathbf{x}_0$ satisfies the following two properties.

(i) Squared distances to $\mathbf{x}^\star$ are geometrically decreasing:

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq \left(1 - \frac{\mu}{L}\right) \|\mathbf{x}_t - \mathbf{x}^\star\|^2, \quad t \geq 0.$$

(ii) The absolute error after T iterations is exponentially small in T:

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{L}{2} \left(1 - \frac{\mu}{L}\right)^T \|\mathbf{x}_0 - \mathbf{x}^\star\|^2, \quad T > 0.$$

Proof. For (i), we show that the noise in (2.21) disappears. By sufficient decrease (Lemma 2.6), we know that

$$f(\mathbf{x}^\star) - f(\mathbf{x}_t) \leq f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq -\frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2,$$

and hence the noise can be bounded as follows, using $\gamma = 1/L$, multiplying by 2γ and rearranging the terms, we get:

$$2\gamma \left(f(\mathbf{x}^\star) - f(\mathbf{x}_t)\right) + \gamma^2 \|\nabla f(\mathbf{x}_t)\|^2 \leq 0,$$

Hence, (2.21) actually yields

$$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq (1 - \mu\gamma)\|\mathbf{x}_t - \mathbf{x}^\star\|^2 = \left(1 - \frac{\mu}{L}\right) \|\mathbf{x}_t - \mathbf{x}^\star\|^2$$

and

$$\|\mathbf{x}_T - \mathbf{x}^\star\|^2 \leq \left(1 - \frac{\mu}{L}\right)^T \|\mathbf{x}_0 - \mathbf{x}^\star\|^2.$$

The bound in (ii) follows from smoothness (2.8), using $\nabla f(\mathbf{x}^\star) = \mathbf{0}$ (Lemma 1.22):

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \nabla f(\mathbf{x}^\star)^\top (\mathbf{x}_T - \mathbf{x}^\star) + \frac{L}{2}\|\mathbf{x}_T - \mathbf{x}^\star\|^2 = \frac{L}{2}\|\mathbf{x}_T - \mathbf{x}^\star\|^2.$$

□

From this, we can derivate a rate in terms of the number of steps required (T). Using the inequality $\ln(1 + x) \leq x$, it follows that after

$$T \geq \frac{L}{\mu} \ln\left(\frac{R^2 L}{2\varepsilon}\right),$$

iterations, we reach absolute error at most ε.

2.9 Exercises

Exercise 13. *Let $\mathbf{c} \in \mathbb{R}^d$. Prove that the spectral norm of $\mathbf{c}^\top$ equals the Euclidean norm of $\mathbf{c}$, meaning that*

$$\max_{\mathbf{x} \neq \mathbf{0}} \frac{|\mathbf{c}^\top \mathbf{x}|}{\|\mathbf{x}\|} = \|\mathbf{c}\| .$$

Exercise 14. *Prove Lemma 2.3: The quadratic function $f(\mathbf{x}) = \mathbf{x}^\top Q \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$ is smooth with parameter $2\|Q\|$.*

Exercise 15. *Consider the function $f(x) = |x|^{3/2}$ for $x \in \mathbb{R}$.*

(i) Prove that f is strictly convex and differentiable, with a unique global minimum $x^\star = 0$.

(ii) Prove that for every fixed stepsize γ in gradient descent (2.1) applied to f, there exists x_0 for which $f(x_1) > f(x_0)$.

(iii) Prove that f is not smooth.

(iv) Let $X \subseteq \mathbb{R}$ be a closed convex set such that $0 \in X$ and $X \neq \{0\}$. Prove that f is not smooth over X.

Exercise 16. *In order to obtain average error at most ε in Theorem 2.1, we need to choose iteration number and stepsize as*

$$T \geq \left(\frac{RB}{\varepsilon} \right)^2 , \quad \gamma := \frac{R}{B\sqrt{T}}.$$

If R or B are unknown, we cannot do this.

Suppose now that we know R but not B. This means, we know a concrete number R such that $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$; we also know that there exists a number B such that $\|\nabla f(\mathbf{x})\| \leq B$ for all $\mathbf{x}$, but we don't know a concrete such number.

Develop an algorithm that—not knowing B—finds a vector $\mathbf{x}$ such that $f(\mathbf{x}) - f(\mathbf{x}^\star) < \varepsilon$, using at most

$$\mathcal{O}\left(\left(\frac{RB}{\varepsilon} \right)^2 \right)$$

many gradient descent steps!

Exercise 17. *Prove Lemma 2.5! (Operations which preserve smoothness)*

Exercise 18. *In order to obtain average error at most ε in Theorem 2.7, we need to choose*

$$\gamma := \frac{1}{L}, \quad T \geq \frac{R^2 L}{2\varepsilon},$$

if $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$. If L is unknown, we cannot do this.

Now suppose that we know R but not L. This means, we know a concrete number R such that $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$; we also know that there exists a number L such that f is smooth with parameter L, but we don't know a concrete such number.

Develop an algorithm that—not knowing L—finds a vector $\mathbf{x}$ such that $f(\mathbf{x}) - f(\mathbf{x}^\star) < \varepsilon$, using at most

$$\mathcal{O}\left(\frac{R^2 L}{2\varepsilon}\right)$$

many gradient descent steps!

Exercise 19. *Let $a \in \mathbb{R}$. Prove that $f(x) = x^4$ is smooth over $X = (-a, a)$ and determine a concrete smoothness parameter L.*

Exercise 20. *Prove Lemma 2.10! (Strongly convex functions have unique global minimum)*

Exercise 21. *Prove Lemma 2.11! (Strongly convex and smooth functions)*

Bibliography

[ACGH18] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *CoRR*, abs/1810.02281, 2018.

[AE08] Herbert Amann and Joachim Escher. *Analysis II*. Birkhäuser, 2008.

[Ber05] Dimitri P. Bertsekas. Lecture slides on convex analysis and optimization, 2005. `http://athenasc.com/Convex_Slides.pdf`.

[BG17] Nikhil Bansal and Anupam Gupta. Potential-function proofs for first-order methods. *CoRR*, abs/1712.04581, 2017.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. `https://web.stanford.edu/~boyd/cvxbook/`.

[Dav59] William C. Davidon. Variable metric method for minimization. Technical Report ANL-5990, AEC Research and Development, 1959.

[Dav91] William C. Davidon. Variable metric method for minimization. *SIAM J. Optimization*, 1(1):1–17, 1991.

[Die69] J. Dieudonneé. *Foundations of Modern Analysis*. Academic Press, 1969.

[DSSSC08] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in

high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pages 272–279, 07 2008.

[FM91] M. Furi and M. Martelli. On the mean value theorem, inequality, and inclusion. *The American Mathematical Monthly*, 98(9):840–846, 1991.

[Gol70] D. Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970.

[Gre70] J. Greenstadt. Variations on variable-metric methods. *Mathematics of Computation*, 24(109):1–22, 1970.

[KSJ18] Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients. *arXiv*, 2018.

[LW19] Ching-Pei Lee and Stephen Wright. First-order algorithms converge faster than $o(1/k)$ on convex problems. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3754–3762, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[Nes83] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Math. Dokl.*, 27(2), 1983.

[Nes18] Yurii Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, second edition, 2018.

[Noc80] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[NP06] Yurii Nesterov and B.T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, Aug 2006.

[NY83] Arkady. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.

[Roc97] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics. Princeton University Press, 1997.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the LASSO. *J. R. Statist. Soc. B*, 58(1):267–288, 1996.

[Vis15] Nisheeth Vishnoi. A mini-course on convex optimization (with a view toward designing fast algorithms), 2015. `https://theory.epfl.ch/vishnoi/Nisheeth-VishnoiFall2014-ConvexOptimization.pdf`.

[Zim16] Judith Zimmermann. *Information Processing for Effective and Stable Admission*. PhD thesis, ETH Zurich, 2016. .