Gradient descent with momentum

Irène Waldspurger*

December 5, 2022

Gradient descent is by far the most well-known optimization algorithm. Because of its simplicity and flexibility, it is a method of choice for many problems. However, it is oftentimes unconveniently slow. In this lecture, we will see that it is possible to speed up gradient descent by incorporating in it a term called *momentum*. We will present two forms of momentum, leading to the following two algorithms:

- heavy ball, which is the simplest form of gradient descent with momentum, and already provides significant speed-ups,
- Nesterov's method, which is slightly more complex, but performs much better than gradient descent on a larger range of problems than heavy ball.

1 Reminder on gradient descent

1.1 Definition

Here, we go back to the most basic form of optimization problems, discussed in the first lectures: unconstrained minimization of a convex and differentiable function over $\mathbb{R}^n$. In all the lecture, $f : \mathbb{R}^n \to \mathbb{R}$ is a convex and differentiable function, and we want to

$$\boxed{\text{find } x_* \text{ such that } f(x_*) = \min_{x\in\mathbb{R}^n} f(x).} \tag{1}$$

*waldspurger@ceremade.dauphine.fr

We assume that a minimizer exists, and denote it x_*[1].

Gradient descent is an iterative algorithm, which moves from one iterate to the next following the direction given by the (opposite of the) gradient. Its definition is recalled in Algorithm 1.

Input: Starting point x_0, number of iterations T, sequence of stepsizes $(\alpha_t)_{0 \leq t \leq T-1}$.
for $t = 0, \dots, T-1$ **do**
  Define $x_{t+1} = x_t - \alpha_t \nabla f(x_t)$.
end
return x_T

Algorithm 1: Gradient descent

The rationale behind this definition is that the gradient of f at a point $x \in \mathbb{R}^n$ provides a linear approximation of f in a neighborhood of x: informally,

$$\forall y \text{ close to } x, \qquad f(y) \approx f(x) + \langle \nabla f(x), y - x \rangle .$$

Consequently, $-\nabla f(x)$ is the direction along which f decays the fastest around x. It is therefore a reasonable direction to follow if we want to minimize f.

1.2 Convergence rate

The objective of this lecture is to propose ameliorations of gradient descent which are faster than the "basic" version. To make our discussion rigorous, we need to formally define the *speed* of an optimization algorithm. Several definitions are possible. Here, assuming the algorithm returns a sequence of iterates $(x_t)_{t \in \mathbb{N}}$, we will focus on the rate at which the sequence

$$f(x_t) - f(x_*)$$

goes to zero. This rate depends on the properties of the objective function f, so we must specify what we assume about f. In this lecture, we will assume either

1. f is *convex* and *smooth*;

[1] At least, we denote one of them x_*: the minimizer may not be unique.

2. f is *strongly convex* and *smooth*.

(See below for the definitions of *strongly convex* and *smooth*.)

These two possible sets of assumptions do not encompass all functions which appear in practical optimization problems, but are nevertheless satisfied in a number of settings. Therefore, they offer a good compromise between practical relevance and simplicity of theoretical analysis.

Definition 1: smoothness

For any $L > 0$, we say that a differentiable function f is L-smooth if ∇f is L-Lipschitz, that is

$$\forall x, y \in \mathbb{R}^n, \quad ||\nabla f(x) - \nabla f(y)|| \leq L||x - y||.$$

Definition 2: strong convexity

For any $\mu > 0$, we say that a differentiable function f is μ-strongly convex if

$$\forall x, y \in \mathbb{R}^n, \quad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}||y - x||^2.$$

Remark

A strongly convex function is necessarily convex.

When f is assumed to be smooth and convex, the convergence rate of gradient descent is, in the worst case, $O\left(\frac{1}{t}\right)$.

Theorem 1: gradient descent - smooth convex case

We assume that f is convex and L-smooth, for some $L > 0$. We consider gradient descent with constant stepsize

$$\forall t \in \mathbb{N}, \quad \alpha_t = \frac{1}{L}.$$

Then, for any $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq \frac{2L||x_0 - x_*||^2}{t + 4}.$$

If, in addition to being smooth, f is assumed to be strongly convex, then the convergence rate can be shown to be geometric. The geometric decay rate is $1 - \kappa$, where κ is the *conditioning* of the problem:

$$\kappa = \frac{\mu}{L},$$

where μ and L are, respectively, the strong convexity and smoothness constants.

Theorem 2: gradient descent - smooth strongly convex case

Let $0 < \mu < L$ be fixed. Let f be L-smooth and μ-strongly convex. We consider gradient descent with constant stepsize: $\alpha_t = \frac{1}{L}$ for all t. Then, for any $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^t ||x_0 - x_*||^2.$$

Theorems 1 and 2 are optimal, in the sense that there exist functions f for which the inequality is an equality (up to minor modifications in the constants).

2 Motivation of momentum

In this section, we motivate the introduction of momentum: we consider a simple function f for which gradient descent converges slowly, explain why convergence is slow, and why momentum can speed it up.

Let f be a simple quadratic function over $\mathbb{R}^2$:

$$\forall (x_1, x_2) \in \mathbb{R}^2, \quad f(x_1, x_2) = \frac{1}{2}\left(\lambda_1 x_1^2 + \lambda_2 x_2^2\right),$$

for parameters $0 < \lambda_1 < \lambda_2$. The unique minimizer of f is

$$x_* = (0, 0).$$

The gradient of f is

$$\forall (x_1, x_2) \in \mathbb{R}^2, \quad \nabla f(x_1, x_2) = (\lambda_1 x_1, \lambda_2 x_2).$$

If we run gradient descent with a constant stepsize $\alpha > 0$, the relation between iterates $x_t = (x_{t,1}, x_{t,2})$ and $x_{t+1} = (x_{t+1,1}, x_{t+1,2})$ is

$$\begin{aligned}(x_{t+1,1}, x_{t+1,2}) &= x_t - \alpha \nabla f(x_t) \\ &= (x_{t,1}, x_{t,2}) - \alpha(\lambda_1 x_{t,1}, \lambda_2 x_{t,2}) \\ &= ((1-\alpha\lambda_1)x_{t,1}, (1-\alpha\lambda_2)x_{t,2})\,.\end{aligned}$$

Since we want the iterates to go as fast as possible to zero, we would like to choose α such that

$$|1-\alpha\lambda_1| \ll 1 \quad \text{and} \quad |1-\alpha\lambda_2| \ll 1.$$

If λ_1 and λ_2 are of the same order, this is fine: it suffices to pick α of the order of $\frac{1}{\lambda_1} \sim \frac{1}{\lambda_2}$.

But if λ_1 is much smaller than λ_2 (that is, the problem is *ill-conditioned*), there is no good choice of α. If we set $\alpha \approx \frac{1}{\lambda_1}$, then

$$1 - \alpha\lambda_2 = 1 - \frac{\lambda_2}{\lambda_1} < -1$$

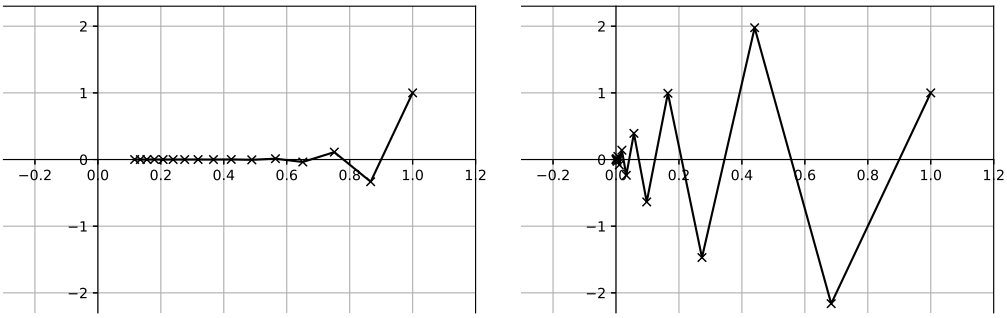
and the second coordinate of the iterates, $x_{t,2}$, diverges when $t \to \infty$. If, on the other hand, we set $\alpha \approx \frac{1}{\lambda_2}$, then the second coordinate goes to 0, and fast, but the first one converges very slowly:

$$1 - \alpha\lambda_1 = 1 - \frac{\lambda_1}{\lambda_2} \approx 1.$$

In this situation, gradient descent is slow. Figure 1a displays the first fifteen iterates in the case where $\lambda_1 = 0.1$ and $\lambda_2 = 1$, for $\alpha = 4/3$ (that is, of the order of $\frac{1}{\lambda_2}$). As expected, the second coordinate goes fast to zero, but the first one decays only slowly.

A possible remedy to this slow convergence is to use the information given by the past gradients when we define x_{t+1} from x_t: instead of moving in the direction given by $-\nabla f(x_t)$, we move in a direction m_{t+1} which is a (weighted) average between $-\nabla f(x_t)$ and the previous gradients $-\nabla f(x_0)$, ..., $-\nabla f(x_{t-1})$. Concretely, this yields the following iteration formula:

$$\begin{aligned} m_{t+1} &= \gamma_t m_t + (1-\gamma_t)\nabla f(x_t), \\ x_{t+1} &= x_t - \alpha_t m_{t+1}. \end{aligned}$$



(a) Standard gradient descent (b) Gradient descent with momentum

Figure 1: First 15 iterates of gradient descent, for $\lambda_1 = 0.1, \lambda_2 = 1$

Here, γ_t and α_t are respectively the momentum and stepsize parameters. The quantity m_t, which is the average of all gradients until step t, is called *momentum*.

Remark

An equivalent iteration formula is

$$x_{t+1} = x_t - \tilde{\alpha}_t \nabla f(x_t) + \tilde{\beta}_t (x_t - x_{t-1}), \tag{2}$$

with $\tilde{\alpha}_t = \alpha_t(1 - \gamma_t)$ and $\tilde{\beta}_t = \frac{\alpha_t \gamma_t}{\alpha_{t-1}}$.

Proof of the remark. From the second equation in the iteration formula:

$$\begin{aligned} &\forall t \in \mathbb{N}, \quad m_{t+1} = \frac{x_t - x_{t+1}}{\alpha_t}, \\ \Rightarrow \quad &\forall t \in \mathbb{N} - \{0\}, \quad m_t = \frac{x_{t-1} - x_t}{\alpha_{t-1}}. \end{aligned}$$

We plug these equalities into the first iteration formula:

$$\begin{aligned} &\forall t \in \mathbb{N} - \{0\}, \quad \frac{x_t - x_{t+1}}{\alpha_t} = \gamma_t \left(\frac{x_{t-1} - x_t}{\alpha_{t-1}} \right) + (1 - \gamma_t) \nabla f(x_t), \\ \Rightarrow \quad &\forall t \in \mathbb{N} - \{0\}, \quad x_{t+1} = x_t - \alpha_t (1 - \gamma_t) \nabla f(x_t) + \frac{\alpha_t \gamma_t}{\alpha_{t-1}} (x_t - x_{t-1}) . \end{aligned}$$

□

Using momentum instead of plain gradient in the iteration formula allows to use a larger stepsize. Indeed, for large stepsizes, $\alpha_t \nabla f(x_t)$ diverges when t grows, which causes the divergence of plain gradient descent. But it is possible that $\alpha_t m_t$ stays bounded, in which case gradient descent with momentum does not diverge: $\alpha_t m_t$ is an average of potentially large gradients pointing to different directions, which may therefore compensate each other. This can be seen in Figure 1b: compared to Figure 1a, the stepsize is larger; consequently, the first coordinate converges faster towards zero, but the second coordinate does not diverge.

3 Heavy ball

The simplest version of gradient descent with momentum is when the momentum and stepsize parameters are constant. It is due to Polyak, and often called *heavy ball*[2].

Input: Starting point x_0, number of iterations T, stepsize α, momentum parameter γ.

Set $m_0 = \nabla f(x_0)$;

for $t = 0, \ldots, T-1$ **do**

define

$$m_{t+1} = \gamma m_t + (1-\gamma)\nabla f(x_t);$$
$$x_{t+1} = x_t - \alpha m_{t+1}.$$

end

return x_T

Algorithm 2: Heavy ball

For proper choices of parameters, heavy ball exhibits a faster convergence rate than plain gradient descent on many natural problems. We will prove this fact for quadratic strongly convex functions.

[2]The name comes from the fact that the momentum term can be seen as an inertia term, which reminds of the movement of a heavy ball falling down a mountain towards a valley.

Theorem 3: heavy ball - quadratic case

Let $0 < \mu < L$ be fixed. Let f be a quadratic function, which is L-smooth and μ-strongly convex. We set

$$\alpha = \frac{1}{\sqrt{\mu L}}, \quad \gamma = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2 .$$

There exists a constant $C_{\mu,L} > 0$ such that, for any $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq C_{\mu,L} t^2 \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^{2t} ||x_0 - x_*||^2.$$

Before proving the theorem, let us compare the convergence rate with gradient descent. From Theorem 2, gradient descent converges geometrically, with decay rate

$$1 - \frac{\mu}{L}.$$

Theorem 3, on the other hand, guarantees for heavy ball a convergence with decay rate

$$\left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2 \approx 1 - 4\sqrt{\frac{\mu}{L}} \quad \text{when } \mu \ll L.$$

For ill-conditioned problems, $\sqrt{\frac{\mu}{L}}$ is much larger than $\frac{\mu}{L}$, resulting in a significant speed-up. As an example, if $\frac{\mu}{L} = 0.01$, dividing $f(x_t) - f(x_*)$ by a factor 10 necessitates around

$$\frac{\ln(10)}{-\ln\left(1 - \frac{\mu}{L}\right)} \approx 230$$

iterations with gradient descent, and only

$$\frac{\ln(10)}{-\ln\left(\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2\right)} \approx 6$$

with heavy ball.

Proof of Theorem 3. Up to a change of coordinates, we can assume that f is of the form

$$f(x_1, \dots, x_n) = \frac{1}{2}\left(\lambda_1 x_1^2 + \cdots + \lambda_n x_n^2\right),$$

where

$$L \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq \mu > 0$$

are the eigenvalues of the matrix representing f.

Denoting $x_t = (x_{t,1}, x_{t,2}, \dots, x_{t,n})$, we have, for each t,

$$\nabla f(x_t) = (\lambda_1 x_{t,1}, \dots, \lambda_n x_{t,n}),$$

hence the evolution equation of heavy ball is, for each $t \in \mathbb{N}$,

$$\begin{aligned} \forall k \leq n, \quad m_{t+1,k} &= \gamma m_{t,k} + (1-\gamma)\lambda_k x_{t,k}; \\ x_{t+1,k} &= x_{t,k} - \alpha m_{t+1,k} = (1 - \alpha(1-\gamma)\lambda_k)x_{t,k} - \alpha\gamma m_{t,k}. \end{aligned}$$

This can be written in matricial form: for each $t \in \mathbb{N}, k \in \{1, \dots, n\}$,

$$\begin{pmatrix} m_{t+1,k} \\ x_{t+1,k} \end{pmatrix} = M_k \begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix}, \quad \text{with } M_k = \begin{pmatrix} \gamma & (1-\gamma)\lambda_k \\ -\alpha\gamma & 1 - \alpha(1-\gamma)\lambda_k \end{pmatrix}$$
$$\Rightarrow \quad \begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix} = M_k^t \begin{pmatrix} m_{0,k} \\ x_{0,k} \end{pmatrix}.$$

For any k, the matrix M_k can be triangularized in a (complex) orthonormal basis: for some unitary matrix G_k, we can write it under the form

$$M_k = G_k \begin{pmatrix} \sigma_k^{(1)} & g_k \\ 0 & \sigma_k^{(2)} \end{pmatrix} G_k^{-1}.$$

For all $t \in \mathbb{N}$,

$$\begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix} = G_k \begin{pmatrix} (\sigma_k^{(1)})^t & g_{t,k} \\ 0 & (\sigma_k^{(2)})^t \end{pmatrix} G_k^{-1} \begin{pmatrix} m_{0,k} \\ x_{0,k} \end{pmatrix},$$
$$\text{with} \quad g_{t,k} = ((\sigma_k^{(1)})^{t-1} + (\sigma_k^{(1)})^{t-2}\sigma_k^{(2)} + \cdots + (\sigma_k^{(2)})^{t-1})g_k.$$

As G_k is unitary, it does not change the norm:

$$\left\| \begin{pmatrix} m_{t,k} \\ x_{t,k} \end{pmatrix} \right\| \leq \left\| \left\| \begin{pmatrix} (\sigma_k^{(1)})^t & g_{k,t} \\ 0 & (\sigma_k^{(2)})^t \end{pmatrix} \right\| \right\| \left\| \begin{pmatrix} m_{0,k} \\ x_{0,k} \end{pmatrix} \right\|.$$

(The triple bar denotes the spectral norm.)

For some constants $C, C' > 0$, the spectral norm can be upper bounded by

$$\begin{aligned}\left|\left|\left|\begin{pmatrix}(\sigma_k^{(1)})^t & g_{k,t}\\ 0 & (\sigma_k^{(2)})^t\end{pmatrix}\right|\right| &\leq C\max\left(|\sigma_k^{(1)}|^t, |\sigma_k^{(2)}|^t, |g_{k,t}|\right)\\ &\leq C' t\max\left(|\sigma_k^{(1)}|, |\sigma_k^{(2)}|\right)^t.\end{aligned}$$

We must compute $\max\left(|\sigma_k^{(1)}|, |\sigma_k^{(2)}|\right)$, where we recall that $\sigma_k^{(1)}, \sigma_k^{(2)}$ are the eigenvalues of M_k. These eigenvalues are the roots of the characteristic polynomial of M_k. A (slightly tedious) computation shows that the polynomial has a negative discriminant. The eigenvalues are therefore complex and conjugate one from each other:

$$|\sigma_k^{(1)}|^2 = |\sigma_k^{(2)}|^2 = \sigma_k^{(1)}\sigma_k^{(2)} = \det(M_k) = \gamma.$$

In particular, $\max\left(|\sigma_k^{(1)}|, |\sigma_k^{(2)}|\right) = \sqrt{\gamma}$, and we get

$$\begin{aligned}\forall k, \quad &\left|\left|\begin{pmatrix}m_{t,k}\\ x_{t,k}\end{pmatrix}\right|\right| \leq C' t\gamma^{t/2}\left|\left|\begin{pmatrix}m_{0,k}\\ x_{0,k}\end{pmatrix}\right|\right|\\ \Rightarrow \quad &|x_{t,k}| \leq C' t\gamma^{t/2}\sqrt{x_{0,k}^2 + m_{0,k}^2} \leq C' t\gamma^{t/2}\sqrt{1+L^2}|x_{0,k}|\\ \Rightarrow\ &f(x_t) - f(x_*) = \sum_{k=1}^{n}\lambda_k x_{t,k}^2 \leq L(1+L^2)C'^2 t^2\gamma^t||x_0||^2.\end{aligned}$$

If we set $C_{\mu,L} = L(1+L^2)C'^2$ and recall that

$$\gamma = \left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2,$$

we get the announced result:

$$f(x_t) - f(x_*) \leq C_{\mu,L}t^2\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^{2t}||x_0 - x_*||^2.$$

□

The theorem we just proved does not extend from strongly convex quadratic functions to general strongly convex functions. Indeed, there are unfavorable strongly convex functions, on which gradient descent with momentum is not faster than its standard version (or even where it diverges whereas plain gradient descent converges). Fortunately, many "interesting" functions are either quadratic or, more frequently, approximately quadratic in the neighborhood of a minimizer. For these functions, heavy ball is usually better than plain gradient descent.

4 Nesterov's method

In the previous section, we have said that heavy ball has a faster convergence rate than gradient descent for quadratic problems, but not for all strongly convex problems. In addition, it does not apply when the objective function is not strongly convex. In this final section, we present an algorithm which solves both these issues. As it has been found by Yurii Nesterov, it is often called "Nesterov's method".

The iteration formula for this algorithm is

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t + \beta_t(x_t - x_{t-1})) + \beta_t(x_t - x_{t-1}), \tag{3}$$

for a proper choice of parameters α_t, β_t. We see that it is very similar to the general form of gradient descent with momentum, as described in Equation (2), with the (important) difference that the gradient is not evaluated at point x_t, but at $x_t + \beta_t(x_t - x_{t-1})$.

If f is assumed to be L-smooth and μ-strongly convex, a simple choice is possible for coefficients α_t, β_t:

$$\forall t, \quad \alpha_t = \frac{1}{L} \quad \text{and} \quad \beta_t = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}.$$

This yields the following algorithm.

Input: Starting point x_0, number of iterations T, smoothness parameter L, strong convexity parameter μ.

Set $x_{-1} = x_0, \alpha = \frac{1}{L}, \beta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$;

for $t = 0, \dots, T-1$ **do**

define

$$x_{t+1} = x_t - \alpha \nabla f\left(x_t + \beta(x_t - x_{t-1})\right) + \beta(x_t - x_{t-1}).$$

end

return x_T

Algorithm 3: Nesterov's algorithm with constant parameters

With this choice, Nesterov's method converges to the minimizer linearly, with decay rate

$$1 - \sqrt{\frac{\mu}{L}},$$

which is similar to the convergence rate of heavy ball, but true for all strongly convex functions, not only quadratic ones!

Theorem 4: Nesterov's method: smooth strongly convex case

Let $0 < \mu < L$ be fixed. Let f be an L-smooth and μ-strongly convex function.

Let $(x_t)_{t \in \mathbb{N}}$ be the sequence computed by Algorithm 3. For all $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq 2\left(1 - \sqrt{\frac{\mu}{L}}\right)^t \left(f(x_0) - f(x_*)\right).$$

When f is not strongly convex, it is not possible to set parameters α_t and β_t to constant values. A more complicated (and admittedly mysterious, at first sight) definition must be used, described in the following algorithm.

Input: Starting point x_0, number of iterations T, smoothness parameter L.

Set $x_{-1} = x_0, \alpha = \frac{1}{L}, \lambda_{-1} = 0$;

for $t = 0, \ldots, T-1$ **do**

define

$$\lambda_t = \frac{1 + \sqrt{1 + 4\lambda_{t-1}^2}}{2};$$

$$\beta_t = \frac{\lambda_{t-1} - 1}{\lambda_t};$$

$$x_{t+1} = x_t - \alpha \nabla f\left(x_t + \beta_t(x_t - x_{t-1})\right) + \beta_t(x_t - x_{t-1}).$$

end

return x_T

Algorithm 4: Nesterov's algorithm with changing parameters

The convergence rate of this algorithm is given in the following theorem.

Theorem 5: Nesterov's method: smooth convex case

Let $L > 0$ be fixed. Let f be an L-smooth convex function.
Let $(x_t)_{t \in \mathbb{N}}$ be the sequence computed by Algorithm 4. For all $t \in \mathbb{N}$,

$$f(x_t) - f(x_*) \leq \frac{2L}{(t+1)^2} ||x_0 - x_*||^2.$$

Comparing the rates in Theorems 1 and 5 shows the superiority of Nesterov's method over gradient descent for smooth convex functions f:

$$\text{gradient descent rate: } O\left(\frac{1}{t}\right);$$

$$\text{Nesterov's method rate: } O\left(\frac{1}{t^2}\right).$$

Actually, it is possible to show that Nesterov's method is *optimal* for smooth convex functions among all first-order algorithms. In other words, for any first order algorithm (that is, an algorithm which only exploits gradient information about f), there exists an "adversarial" objective function f,

which is L-smooth and convex, such that, after t steps,

$$f(x_t) - f(x_*) \geq \frac{3L}{32(t+1)^2}||x_0 - x_*||^2.$$

This means that, up to the constant, no first-order algorithm can achieve a better convergence rate than the one in Theorem 5.

Nesterov's method is also optimal for smooth strongly convex functions among all first-order algorithms: no first-order algorithm can achieve a better convergence rate, for L-smooth and μ-strongly convex functions, than the one guaranteed by Theorem 4.

5 References

The main references used to prepare these notes are the original article where Polyak introduced the heavy ball algorithm,

- *Some methods of speeding up the convergence of iteration methods*, by B. T. Polyak, Ussr computational mathematics and mathematical physics, volume 4(5), pages 1-17 (1964),

two classical (classical to be, for the second one) books on optimization,

- *Introductory lectures on convex optimization: a basic course*, by Y. Nesterov, Springer Science & Business Media, volume 87 (2003),
- *Optimization for data analysis*, by S. J. Wright and B. Recht, Cambridge University Press (2022),

and two blog posts by S. Bubek on Nesterov's method for smooth convex functions,

- http://blogs.princeton.edu/imabandit/2013/04/01/acceleratedgradientdescent/,
- http://blogs.princeton.edu/imabandit/2018/11/21/a-short-proof-for-nesterovs-momentum/.

For another presentation of the advanced aspects of gradient descent, the reader can also refer to

- *Lecture notes on advanced gradient descent*, by C. Royer, https://www.lamsade.dauphine.fr/%7Ecroyer/ensdocs/GD/LectureNotesOML-GD.pdf (2021).