

Lecture 5: Gradient Descent and Momentum

November 22, 2025

Lecturer: Lei Wu

Scribe: Lei Wu

1 Problem Setup

Let $f : \Omega \mapsto \mathbb{R}$ be an objective function, where $\Omega \subset \mathbb{R}^d$ denotes the input domain. The task of optimization is to solve the following problem:

$$\inf_{x \in \Omega} f(x), \quad (1)$$

where we use \inf instead of \min , as the minimum may not be attainable in Ω .

When Ω is a constrained domain, the problem is called **constrained optimization**; otherwise, it is referred to as **unconstrained optimization**. In this lecture, we focus on the unconstrained case for simplicity. Dealing with constraints can be quite complex, and whenever possible, it is generally advisable to reformulate the problem as an unconstrained one.

In most practical cases, the optimization problem (1) cannot be solved in closed form. Instead, one relies on iterative methods¹, e.g., the gradient descent (GD) algorithm:

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

where x_t represents the solution at the t -th step. A fundamental question in optimization is to determine the conditions under which these iterations converge, and the rate at which they do so.

Criteria for measuring convergence. Depending on the properties of the objective function, several criteria are commonly used to quantify convergence:

- **Point convergence.** If $x^* = \operatorname{argmin}_x f(x)$ exists, we can measure the convergence by $\|x_t - x^*\|$.
- **Function-value convergence.** We can also measure convergence using the objective gap $f(x_t) - \inf_x f(x)$, which remains meaningful even if the minimizer x^* is not attainable.
- **Gradient convergence.** When x^* or $f(x^*)$ are unknown, $\|\nabla f(x_t)\|$ is typically the only available indicator of convergence. Particularly, for non-convex problems, x_t may only converge to a critical point where $\nabla f(x) = 0$, in which case the magnitude of the gradient provides a natural measure of convergence.

It is worth emphasizing that, in machine learning (ML), the most relevant criterion is the function-value (loss) convergence, as it directly reflects model performance. Point convergence is less meaningful in modern ML, where models are typically over-parameterized or possess many degenerate solutions, leading to multiple equivalent minima. Gradient convergence is commonly used when analyzing convergence for non-convex landscapes. However, one should keep in mind that a small gradient norm does not necessarily imply good model performance, as the loss value may still remain high.

¹In machine learning, such methods are often referred to as optimizers.

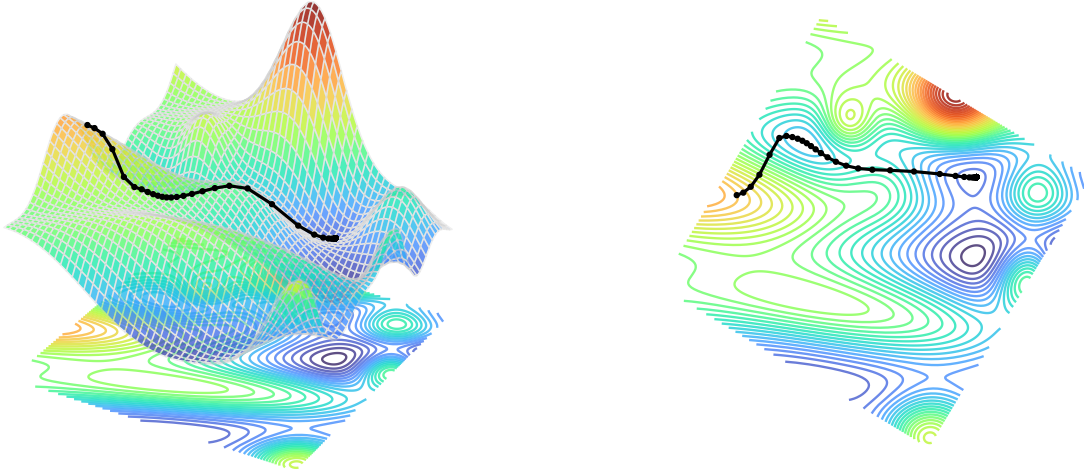


Figure 1: A visualization the landscape $f(\cdot)$ and how an optimization algorithm explores the landscape and finally locate the minimizer.

Optimization landscape. The *optimization landscape* refers to the geometric shape of the objective function $f(x)$ over its domain; see Figure 1 for an illustration. Intuitively, one can think of $f(x)$ as defining a surface where higher function values correspond to peaks and lower values correspond to valleys. An optimization algorithm, such as gradient descent, explores this landscape in search of a valley bottom—i.e., a minimizer of f . Viewing optimization through the lens of the landscape helps us understand *why and how* algorithms converge: the slope of the landscape determines the search direction, the curvature influences the speed of convergence, and the presence of multiple valleys explains challenges in nonconvex optimization. This geometric intuition will guide our analysis throughout the course.

2 Gradient Descent

Gradient descent (GD) generates a sequence of iterates according to

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

where η_t denotes the learning rate (or step size) of the t -th step. Intuitively, GD moves the iterate in the direction of the steepest descent, which corresponds to $-\nabla f(x_t)$ under the standard ℓ_2 metric.

Popular schedules of tuning learning rates include the following three ones.

- **Line search:** $\eta_t = \operatorname{argmin}_{\eta \geq 0} f(x_t - \eta \nabla f(x_t))$. This adaptive approach is common in classical numerical optimization but rarely used in machine learning.
- **Constant learning rate:** $\eta_t = \eta$. This is the simplest and most widely used choice in practice.
- **Decay learning rate:** e.g., $\eta_t = \eta_0 / (1 + t)$. This type of schedules are often used when $f(\cdot)$ is non-smooth.

Let us first illustrate why decaying learning rate is sometimes required.

Example 2.1. Consider $f(x) = |x|$. The GD iteration becomes $x_{t+1} = x_t - \eta_t \text{sign}(x_t)$. Since f is non-smooth at the origin, the gradient direction changes abruptly when x_t crosses zero. To ensure convergence of the iterates $\{x_t\}$, the learning rate must diminish to zero; otherwise, the sequence will oscillate indefinitely around the minimizer.

This phenomenon is specific to non-smooth objectives. When $f \in C^1(\mathbb{R}^d)$, as we shall see later, gradient descent can converge with a constant learning rate.

From discrete-time GD to continuous-time gradient flow. When the learning rate $\eta_t = \eta$ is small, we can view GD as a discrete approximation to a continuous-time process. Rewriting the GD update gives

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

If we interpret the iteration index t as a rescaled time variable $\tau = t\eta$, then the difference quotient approximates a time derivative:

$$\frac{x_{t+1} - x_t}{\eta} \approx \frac{dx(\tau)}{d\tau}.$$

Taking the limit $\eta \rightarrow 0$ yields the **gradient flow** (GF) equation

$$\dot{x}_\tau = -\nabla f(x_\tau),$$

which describes the continuous-time evolution of x_τ along the steepest descent direction.

Thus, gradient descent can be viewed as a forward–Euler discretization of the continuous-time gradient flow. Analyzing the gradient flow is often more tractable and offers clearer intuition about the underlying dynamics, which can then be translated back to the discrete GD algorithm.

2.1 One-Step Loss Descent

The convergence analysis relies on an estimate of the one-step loss descent. For the GF, it is easy to verify

$$\frac{df(x_t)}{dt} = \langle \nabla f(x_t), \dot{x}_t \rangle = -\|\nabla f(x_t)\|^2. \quad (3)$$

This provides an intuition behind GF/GD convergence. The rate of loss descent depends on the gradient's norm. For the discrete-time GD, we need to make further assumption on the objective function's smoothness.

Definition 2.2 (L -smoothness). A differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is said to be L -smooth if

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

If $f \in C^2(\mathbb{R}^d)$, the above condition is equivalent to $\sup_x \|\nabla^2 f(x)\|_2 \leq L$. The following lemma shows that smooth functions grow at most quadratically.

Lemma 2.3. If f is L -smooth, then for any $x, y \in \mathbb{R}^d$, $f(y) \leq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{L}{2}\|y - x\|^2$.

Proof. Omitted! □

Lemma 2.4 (One-step descent of GD). *Assume that f is L -smooth and the learning rate satisfies $\eta \leq 1/L$. Then the gradient descent satisfies*

$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2} \|\nabla f(x_t)\|^2.$$

Proof. Using the L -smoothness and Lemma (2.3), we have

$$\begin{aligned} f(x_{t+1}) &= f(x_t - \eta \nabla f(x_t)) \\ &\leq f(x_t) + \langle x_{t+1} - x_t, \nabla f(x_t) \rangle + \frac{L}{2} \|x_{t+1} - x_t\|^2 \\ &= f(x_t) + \left(-\eta + \frac{L\eta^2}{2} \right) \|\nabla f(x_t)\|^2 \leq f(x_t) - \frac{\eta}{2} \|\nabla f(x_t)\|^2, \end{aligned}$$

where the last inequality follows from $\eta \leq 1/L$. □

Interplay between learning rate and local curvature. In Lemma 2.4, we assumed the learning rate to be sufficiently small for simplicity. However, depending on the objective function, the dynamics of gradient descent can vary significantly under different learning rates. To illustrate this effect, consider the one-dimensional quadratic function

$$f(x) = \frac{\lambda}{2} x^2,$$

where $\lambda > 0$ characterizes the curvature of the loss landscape near the global minimizer $x = 0$. In this case, the GD updates follow

$$x_t = x_{t-1} - \eta \lambda x_{t-1} = (1 - \eta \lambda) x_{t-1} = (1 - \eta \lambda)^t x_0.$$

The resulting dynamics exhibit distinct behaviors depending on the value of learning rate:

- If $\eta \leq 1/\lambda$, then x_t converges monotonically to zero, resembling the behavior of gradient flow.
- If $1/\lambda < \eta < 2/\lambda$, then x_t still converges to zero but oscillates around the minimum, showing qualitatively different dynamical behavior from gradient flow.
- If $\eta = 2/\lambda$, then GD converges to a periodic orbit $(x_0, -x_0)$.
- If $\eta > 2/\lambda$, then GD diverges.

Figure 2 illustrates these regimes. To ensure monotone loss convergence, the learning rate must satisfy $\eta < 1/\lambda$. The key insight is that the appropriate choice of learning rate depends critically on the local curvature of the landscape. Similar behavior extends to general objective functions, as the landscape near a minimum can often be locally approximated by a quadratic function.

2.2 Non-convex Analysis

Theorem 2.5. *Let $f \in C^1(\mathbb{R}^d)$. Then the GF satisfies $\min_{s \in [0, t]} \|\nabla f(x_s)\| = O(1/\sqrt{t})$.*

This theorem shows that the gradient norm decreases to zero in a $O(1/\sqrt{t})$ rate.

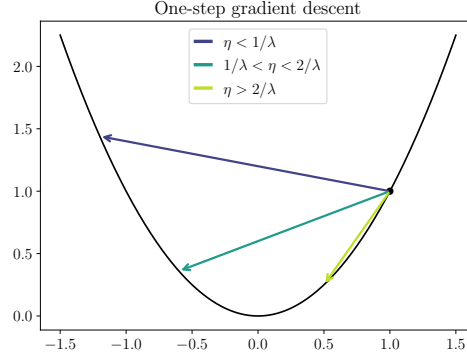


Figure 2: The effect of the learning rate size for gradient descent, where λ denotes the local curvature. The three cases correspond to $\eta < 1/\lambda$, $1/\lambda < \eta < 2/\lambda$, and $\eta > 2/\lambda$, respectively.

Proof. By Eq. (3), we have

$$f(x_t) - f(x_0) = \int_0^t \frac{df(x_s)}{ds} ds = - \int_0^t \|\nabla f(x_s)\|^2 ds.$$

Therefore,

$$\min_{s \in [0, t]} \|\nabla f(x_s)\| \leq \sqrt{\frac{f(x_0) - f(x_t)}{t}} \leq \sqrt{\frac{f(x_0) - \inf_x f(x)}{t}}$$

□

Theorem 2.6. Let f be L -smooth and $\{x_t\}$ be the GD solutions. Suppose $\eta \leq 1/L$. Then,

$$\min_{s=0,1,\dots,t-1} \|\nabla f(x_s)\| \leq \sqrt{\frac{f(x_0) - \inf_x f(x)}{2\eta t}}.$$

Proof. By Lemma (2.4), we have

$$f(x_{t+1}) - f(x_t) \leq -\frac{\eta}{2} \|\nabla f(x_t)\|^2.$$

Summing over t and noticing that the left side is a telescoping sum, we obtain

$$\inf_x f(x) - f(x_0) \leq f(x_t) - f(x_0) \leq -\frac{\eta}{2} \sum_{s=0}^{t-1} \|\nabla f(x_s)\|^2.$$

This implies that

$$\min_{s=0,1,\dots,t-1} \|\nabla f(x_s)\| \leq \sqrt{\frac{\sum_{s=0}^{t-1} \|\nabla f(x_s)\|^2}{t}} \leq \sqrt{\frac{f(x_0) - \inf_x f(x)}{2\eta t}}.$$

□

2.3 Convex Analysis

Let $S_f = \arg \min_x f(x)$ denote the set of minimizers of f , and define the distance from a point $x \in \mathbb{R}^d$ to a set $A \subset \mathbb{R}^d$ as $d(x, A) = \inf_{x' \in A} \|x - x'\|$. When f is not strongly convex, the minimizer set S_f may contain multiple points; in this case, S_f is often referred to as the *minimizer manifold*. For example, for $f(x_1, x_2) = (x_1 - 1)^2$, we have $S_f = \{x \in \mathbb{R}^2 : x_1 = 1\}$.

We start by considering gradient flow:

Theorem 2.7. *Suppose that $f \in C(\mathbb{R}^d)$ is convex. Then, we have*

$$f(x_t) - \inf_x f(x) \leq \frac{d^2(x_0, S_f)}{2t}$$

Proof. For any $\bar{x} \in \mathbb{R}^d$, consider the Lyapunov function

$$J(t) = t(f(x_t) - f(\bar{x})) + \frac{1}{2}\|x_t - \bar{x}\|^2. \quad (4)$$

Then, by the convexity, we have

$$\dot{J}(t) = f(x_t) - f(\bar{x}) - t\|\nabla f(x_t)\|^2 + \langle \bar{x} - x_t, \nabla f(x_t) \rangle \leq -t\|\nabla f(x_t)\|^2 \leq 0.$$

Then, we have $J(t) \leq J(0)$, which implies

$$t(f(x_t) - f(\bar{x})) + \frac{1}{2}\|x_t - \bar{x}\|^2 \leq \frac{1}{2}\|x_0 - \bar{x}\|^2. \quad (5)$$

Thus for any $\bar{x} \in S$, we have

$$f(x_t) - f(\bar{x}) \leq \frac{\|x_0 - \bar{x}\|^2}{2t}.$$

Taking $\bar{x} = \operatorname{argmin}_{x \in S_f} d(x_0, x)$ leads to the conclusion. \square

Remark 2.8 (Implicit bias). From (5), we have for any $\bar{x} \in \mathbb{R}^d$ that $\|x_t - \bar{x}\| \leq \|x_0 - \bar{x}\|$. For gradient flow (GF) with zero initialization $x_0 = 0$, it follows that $\|x_t\| \leq \|x_t - \bar{x}\| + \|\bar{x}\| \leq 2\|\bar{x}\|$. Taking $\bar{x} \in S_f$ yields

$$\|x_t\| \leq 2 \inf_{x \in S_f} \|x\|.$$

Hence, up to a constant factor, gradient flow with zero initialization converges to a minimizer with approximately the smallest norm.

Optimality. The convergence rate $O(1/t)$ is optimal for convex objective functions that admit a minimizer, that is, when $d(x_0, S_f) < \infty$.

Example 2.9. Let $f : \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = |x|^n$. This function is convex for $n \geq 1$ since $f''(x) = n(n-1)|x|^{n-2}$. By the energy dissipation identity, we have

$$\frac{d}{dt}f(x_t) = -f'(x_t)^2 = -n^2 x_t^{2n-2} = -n^2 f^{2-\frac{2}{n}}(x_t).$$

We denote $z_t = f(x_t)$ for brevity and solve

$$\dot{z} = -n^2 z^{2-\frac{2}{n}} \quad \Rightarrow \quad \frac{d}{dt} z^{\frac{2}{n}-1} = z^{\frac{2}{n}-2} \dot{z} = -\frac{n^2}{\frac{2}{n}-2}$$

so $z_t = \left(z_0 + \frac{n}{n-2}t\right)^{\frac{-n}{n-2}}$. Since $\frac{n}{n-2} \rightarrow 1$ as $n \rightarrow \infty$, there is no $\alpha > 1$ such that we could guarantee that $f(x_t) - \inf_x f(x) \leq \frac{C}{t^\alpha}$ for any $C > 0$ without making additional assumptions on f .

Remark 2.10. For convex functions whose minimizers lie at infinity (i.e., classification problem with cross entropy loss), the convergence rate of gradient descent can be substantially slower than $1/t$. An illustrative example is provided below.

Example 2.11. Consider $f_\alpha : (0, \infty) \rightarrow \mathbb{R}$, $f_\alpha(x) = x^{-\alpha}$ for $\alpha > 0$. Since $f'_\alpha(x) = -\alpha x^{-\alpha-1}$ and $f''_\alpha(x) = -\alpha(-\alpha-1)x^{-\alpha-2}$, the function f_α is convex. We can solve the gradient flow equation

$$\dot{x} = -f'_\alpha(x) = \alpha x^{-\alpha-1}$$

with initial condition $x_0 = 1$ explicitly since

$$\frac{d}{dt} x^{2+\alpha} = (2+\alpha)x^{1+\alpha}\dot{x} = C_\alpha \quad \Rightarrow \quad x_t = (1 + C_\alpha t)^{-\frac{1}{2+\alpha}},$$

which satisfies

$$f(x(t)) = (1 + C_\alpha t)^{-\frac{\alpha}{2+\alpha}} \sim t^{-\frac{\alpha}{2+\alpha}}.$$

If α is close to zero, the objective function decays very slowly. Intuitively, the reason is that the objective function is very flat, so the gradient is too small to induce significant changes in x over a short time, and small changes in x do not decrease f by a noticeable amount.

Next, we show that the same convergence rate also hold for the discrete-time GD.

Theorem 2.12. Assume that f is L -smooth and convex. If the learning rate $\eta \leq 1/L$, then GD satisfies

$$f(x_T) - \min_x f(x) \leq \frac{d^2(x_0, S_f)}{2T\eta}.$$

Proof. By Lemma 2.4, we have

$$f(x_{t+1}) \leq f(x_t) - \frac{\eta}{2} \|\nabla f(x_t)\|^2.$$

Since f is convex, it holds for any $x^* \in S_f$ that

$$f(x^*) \geq f(x_t) + \langle \nabla f(x_t), x^* - x_t \rangle.$$

Combining the two inequalities gives

$$\begin{aligned} f(x_{t+1}) - f(x^*) &\leq \langle \nabla f(x_t), x_t - x^* \rangle - \frac{\eta}{2} \|\nabla f(x_t)\|^2 \\ &= -\frac{1}{2\eta} (\|x_t - \eta \nabla f(x_t) - x^*\|^2 - \|x_t - x^*\|^2) \end{aligned}$$

$$= -\frac{1}{2\eta} (\|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2). \quad (6)$$

The sum telescopes, leaving

$$\sum_{t=0}^{T-1} (f(x_{t+1}) - f(x^*)) \leq -\frac{1}{2\eta} (\|x_T - x^*\|^2 - \|x_0 - x^*\|^2).$$

Since $f(x_t)$ is non-increasing, $f(x_T) \leq \frac{1}{T} \sum_{t=0}^{T-1} f(x_t)$. Hence,

$$f(x_T) - f(x^*) \leq \frac{1}{2\eta T} (\|x_0 - x^*\|^2 - \|x_T - x^*\|^2) \leq \frac{\|x_0 - x^*\|^2}{2\eta T}.$$

□

2.4 KL Analysis

Let us look at again the energy dissipation of GF: $\frac{df(x_t)}{dt} = -\|\nabla f(x_t)\|^2$. Obviously, the following condition ensures that the rate of energy dissipation does not degenerate.

Definition 2.13. $f \in C^1(\mathbb{R}^d)$ is said to satisfy the Kurtyak-Lojasiewicz (KL) inequality if there exist $\mu > 0$ such that

$$\|\nabla f(x)\|^2 \geq 2\mu \left(f(x) - \inf_x f(x) \right)^\alpha \quad \forall x \in \mathbb{R}^d,$$

where α is often called the Lojasiewicz exponent

An immediate consequence of this inequality is stated as follows.

Lemma 2.14. *If f satisfies the KL condition, then all stationary points are global minima.*

The KL condition captures how “sharp” or “flat” the landscape of f is around its critical points. It suggests that if $f(x)$ is close to $f(x^*)$, the gradient $\nabla f(x)$ must be also small, which provides a kind of control to the descent behavior of gradient-based methods.

The case of $\alpha = 1$ is referred to as the Polyak-Lojasiewicz (PL) condition. This case is important as it yields an exponential convergence:

Theorem 2.15. *If f satisfies the PL inequality, we have*

$$f(x_t) - \inf_x f(x) \leq e^{-2\mu t} (f(x_0) - \inf_x f(x)).$$

Proof. Note $\frac{d[f(x_t) - \inf_x f(x)]}{dt} = -\|\nabla f(x_t)\|^2 \leq -2\mu[f(x_t) - \inf_x f(x)]$, which implies the conclusion. □

If $f \in C^1(\mathbb{R}^d)$ is strongly convex, then it automatically satisfies the PL condition.

Definition 2.16. $f \in C^1(\mathbb{R}^d)$ is said to be strongly convex if there exist a $\mu > 0$ such that

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|y - x\|^2, \quad \forall x, y \in \mathbb{R}^d. \quad (7)$$

Note that if $f \in C^2(\mathbb{R}^d)$, the condition (7) is equivalent to $\inf_x \lambda_{\min}(\nabla^2 f(x)) \geq \mu$.

Lemma 2.17. *If f is strongly convex with constant μ , then f satisfies the PL condition:*

$$\|\nabla f(x)\|^2 \geq 2\mu (f(x) - f(x^*)).$$

Proof. Note that the minimum of the right hand side of (7) is attained in $\tilde{y} = x - \frac{1}{\mu} \nabla f(x)$. Thus,

$$\begin{aligned} f(y) &\geq f(x) + \langle \nabla f(x), \tilde{y} - x \rangle + \frac{\mu}{2} \|\tilde{y} - x\|^2 \\ &\geq f(x) - \frac{1}{2\mu} \|\nabla f(x)\|^2. \end{aligned}$$

Taking $y = x^*$ completes the proof. \square

Remark 2.18. It is more intuitive to check this property with $f(x) = \frac{1}{2} x^\top A x$.

However, the PL condition is *much weaker* than strong convexity and can hold even for non-convex or under-determined problems. Below we illustrate two representative examples.

- **Linear mapping of a strongly convex function.** Let $g : \mathbb{R}^k \rightarrow \mathbb{R}$ be μ_0 -strongly convex and $A \in \mathbb{R}^{k \times d}$. Denote by $\sigma_k(A)$ the smallest singular value of A . Define

$$f(x) := g(Ax).$$

Then f satisfies the PL condition. Indeed, $\nabla f(x) = A^\top \nabla g(Ax)$ and thus,

$$\|\nabla f(x)\|^2 = \|A^\top \nabla g(Ax)\|^2 \geq \sigma_k^2(A) \|\nabla g(Ax)\|^2 \geq 2\sigma_k^2(A) \mu_0 g(Ax) = 2\sigma_k^2(A) \mu_0 f(x).$$

When $k < d$, f is *not* strongly convex because its Hessian is singular, yet it still satisfies the PL condition.

A particularly important instance is the *over-parameterized linear regression*

$$F(\beta) = \frac{1}{n} \sum_{i=1}^n (\Phi(x_i)^\top \beta - y_i)^2 = \frac{1}{n} \|\Phi(X)\beta - y\|^2,$$

where $\Phi(X) = (\Phi(x_1), \dots, \Phi(x_n))^\top \in \mathbb{R}^{n \times d}$. In the over-parameterized regime ($d > n$), F is PL but not strongly convex. Consequently, training this over-parametrized model with GD enjoys exponential convergence even though the objective is degenerate.

- **A non-convex but PL example.** Consider

$$f(x) = x^2 + 3 \sin^2(x).$$

Although f is non-convex due to the oscillatory term, it satisfies the PL condition and thus admits exponential GD convergence. We refer to Figure 3 for an illustration.

The PL condition captures a broad class of objectives beyond strong convexity. It ensures fast convergence of GD even when the function is non-convex or the problem is over-parameterized.

For general KL conditions, we have the following result.

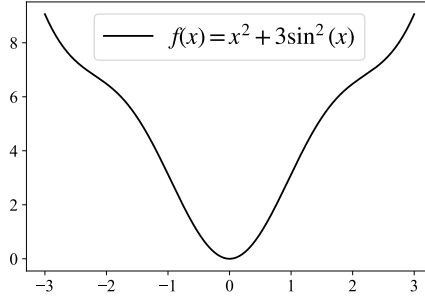


Figure 3: A non-convex yet PL function $f(x) = x^2 + 3 \sin^2(x)$.

Theorem 2.19. Assume that f satisfies the KL condition with parameter $\alpha \in [0, 1]$. Then, the convergence rate of the gradient flow depends on α as follows:

- If $\alpha > 1$, then $f(x_t) - \inf_x f(x) = O(t^{-1/(\alpha-1)})$.
- If $\alpha = 1$, then $f(x_t) - \inf_x f(x) = O(e^{-2\mu t})$.
- If $\alpha < 1$, then $f(x_t) - \inf_x f(x) \leq (f(x_0)^{1-\alpha} - \lambda(1-\alpha)t)^{\frac{1}{1-\alpha}}$ for all $t < \frac{f(x_0)^{1-\alpha}}{\lambda(1-\alpha)}$. In this case, the gradient flow converges in finite time.

The proof is omitted here and left as an exercise.

So depending on the exponent α in KL condition, three types of behaviors may occur: convergence at an algebraic rate (see also Example 2.11), convergence at an exponential rate, and in finite time. Note that the convergence in finite time cannot be recovered in practice, as the condition also prevents the objective function from being smooth close to a minimum. This requires choosing a decaying learning rate for GD, which pushes the time of convergence to infinity.

2.5 Quadratic Functions

We now consider the simplest setting, a quadratic function $f(x) = \frac{1}{2}x^\top Hx$, which enables a more precise characterization of GD convergence. When $\lambda_{\min}(H) > 0$, the function is strongly convex. However, strong convexity alone does not guarantee that the optimization problem is *easy*: the convergence behavior of GD can still vary significantly depending on the spectral properties of H . Thus, a more refined characterization is needed to distinguish *well-conditioned* (easy) and *ill-conditioned* (difficult) quadratic problems.

Consider a simple quadratic objective

$$f(x, y) = \frac{1}{2}x^2 + \frac{1}{2\epsilon}y^2, \quad (8)$$

where $\epsilon \ll 1$. The corresponding GD updates are

$$\begin{cases} x_{t+1} = x_t - \eta x_t, \\ y_{t+1} = y_t - \frac{\eta}{\epsilon} y_t, \end{cases} \quad \Rightarrow \quad \begin{cases} x_t = (1 - \eta)^t x_0, \\ y_t = \left(1 - \frac{\eta}{\epsilon}\right)^t y_0. \end{cases}$$

Hence,

$$f(x_t, y_t) = \frac{1}{2}(1 - \eta)^{2t}x_0^2 + \frac{1}{2\epsilon}\left(1 - \frac{\eta}{\epsilon}\right)^{2t}y_0^2.$$

For convergence, the step size must satisfy $\eta < 2\varepsilon$, since the curvature along the sharp y -direction is $1/\varepsilon$. This restriction forces GD to adopt a very small learning rate. While such a step size ensures stability in the sharp direction, it leads to extremely slow progress along the flat x -direction, with a convergence rate on the order of $O((1 - \varepsilon)^{2t})$. Figure 4 illustrates how the GD converges for small and large learning rates.

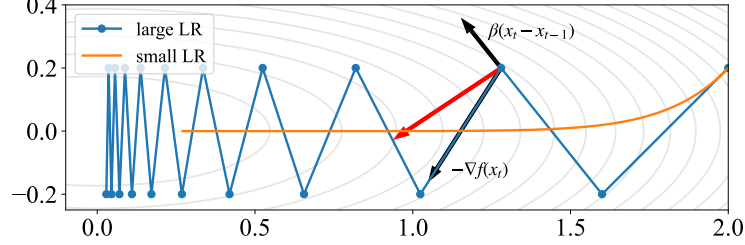


Figure 4: The GD trajectories for $f(x, y) := \frac{1}{2}(x^2 + 10y^2)$. Large-LR GD wastes time in oscillating around the valley. Small-LR GD converges well but move too little in each step. The red arrow denotes the direction proposed by heavy-ball momentum (HBM) method, which is better than the negative gradient direction.

The above analysis extends naturally to a general quadratic objective

$$f(x) = \frac{1}{2}x^\top Hx,$$

for which gradient descent (GD) updates as $x_{t+1} = (I - \eta H)x_t$. Let $H = U\Sigma U^\top = \sum_{j=1}^d \lambda_j u_j u_j^\top$ with eigenvalues $\lambda_1 \geq \dots \geq \lambda_d > 0$. Expanding $x_t = \sum_{j=1}^d \tilde{x}_j(t) u_j$ gives

$$\tilde{x}_j(t+1) = (1 - \eta\lambda_j)\tilde{x}_j(t) = (1 - \eta\lambda_j)^t \tilde{x}_j(0),$$

showing that GD evolves independently along each eigendirection. Moreover, the convergence along sharper direction is faster and for convergence, the updates must satisfy

$$\max_j |1 - \eta\lambda_j| < 1 \quad \Rightarrow \quad \eta \leq \frac{2}{\lambda_1}.$$

Hence, the sharpest direction u_1 determines the largest admissible step size. Meanwhile, the overall convergence is restricted by the flat directions. In particular, when $\lambda_1 \gg \lambda_d$, the small $\eta \propto \lambda_1^{-1}$ causes the slowest progress along the flattest direction u_d , producing the characteristic **zig-zag trajectory** of GD; see Figure 4 for an illustration.

The above intuition can be formalized by using the concept: **condition number** $\kappa = \lambda_1/\lambda_d$. Then

$$f(x_t) = \frac{1}{2} \sum_{j=1}^d \lambda_j (1 - \eta\lambda_j)^{2t} \tilde{x}_j^2(0) \lesssim C_0 \left(\frac{\kappa - 1}{\kappa + 1} \right)^{2t}.$$

To reach an ε -accurate solution, GD requires

$$T_\varepsilon = O(\kappa \log(1/\varepsilon))$$

iterations. A large κ thus implies slow convergence due to ill-conditioning.

3 Heavy-ball Momentum

To alleviate the zig-zag issue of large-LR GD and accelerate GD, one idea is to use past informations to construct a better update direction. By looking at Figure 4, this seems doable and in particular visually it seems that $-\nabla f(x_t) + \beta(x_t - x_{t-1})$ can yield a better direction if choosing β appropriately. It turns out that this is exactly the Heavy-Ball Momentum (HBM) method introduced by [Polyak, 1964]:

$$x_{t+1} = x_t - \eta \nabla f(x_t) + \beta(x_t - x_{t-1}). \quad (9)$$

By introduce the momentum $v_t = (x_t - x_{t-1})/\eta$, the above update can be also written as

$$\begin{aligned} v_{t+1} &= \beta v_t - \nabla f(x_t) \\ x_{t+1} &= x_t + \eta v_{t+1}. \end{aligned} \quad (10)$$

In this regard, β is called the momentum factor.

Other variants. In the literature, there are also two other formulations of heavy-ball momentum.

- By let $v_t = x_t - x_{t-1}$, (9) can be rewritten as

$$\begin{aligned} v_{t+1} &= \beta v_t - \eta \nabla f(x_t) \\ x_{t+1} &= x_t + v_{t+1}, \end{aligned} \quad (11)$$

- Let $\eta = (1 - \beta)\bar{\eta}$ and $v_t = (x_t - x_{t-1})/\bar{\eta}$. Then, (9) can be written as

$$\begin{aligned} v_{t+1} &= \beta v_t + (1 - \beta)(-\nabla f(x_t)) \\ x_{t+1} &= x_t + \bar{\eta} v_{t+1}, \end{aligned} \quad (12)$$

In this variant, the momentum is updated as a convex combination of the previous momentum and the current negative gradient.

These formulations are essentially equivalent but hyper-parameters may have different meanings. In all formulations, β 's are the same but the learning rates may scale differently with β . Different machine learning packages may use different formulations to implement HBM. For instance, PyTorch uses (10) but TensorFlow use (11). Therefore, one should be careful about the choice of learning rate.

3.1 Preliminary Analyses

Considering the update (10), we have

$$v_t = - \sum_{s=0}^t \beta^{t-s-1} \nabla f(x_s) + \beta^t v_0,$$

which implies that the momentum is just an *exponential moving average* (EMA) of past gradients. In particular, it tell us that we must set $\beta < 1$; otherwise, $\|v_t\|$ will blow up.

Continuous-time limit. Let $\beta = 1 - \alpha$. Then the HBM update (9) can be rewritten as

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

To reveal its continuous-time behavior, divide both sides by η :

$$\frac{x_{t+1} - 2x_t + x_{t-1}}{(\sqrt{\eta})^2} = -\frac{\alpha}{\sqrt{\eta}} \frac{x_t - x_{t-1}}{\sqrt{\eta}} - \nabla f(x_t).$$

The left-hand side can be viewed as a *central-difference approximation* of the second derivative:

$$\frac{x_{t+1} - 2x_t + x_{t-1}}{(\sqrt{\eta})^2} \approx \ddot{X}(t),$$

where $X(t)$ is the continuous-time interpolation of the discrete trajectory $\{x_t\}$, and the step size in physical time is $\sqrt{\eta}$. Similarly,

$$\frac{x_t - x_{t-1}}{\sqrt{\eta}} \approx \dot{X}(t)$$

approximates the first derivative.

To obtain a nontrivial continuous-time limit, we consider the scaling

$$1 - \beta, \eta \rightarrow 0, \quad \frac{1 - \beta}{\sqrt{\eta}} \rightarrow \gamma,$$

so that the discrete damping $1 - \beta$ and step size η balance each other. Let $X(t\sqrt{\eta}) = x_t$. Passing to the limit yields the second-order ordinary differential equation

$$\ddot{X} = -\gamma \dot{X} - \nabla f(X), \tag{13}$$

which is the continuous-time dynamics of the heavy-ball method. Let $V = \dot{X}$ be the velocity. Then, Eq. (13) can be written as

$$\begin{cases} \dot{X} &= V \\ \dot{V} &= -\gamma V - \nabla f(X). \end{cases} \tag{14}$$

This HBM-ODE is exactly the Newton's Law for the motion of a ball of mass 1, where $f(\cdot)$ is the potential energy and $-\gamma \dot{X}$ is the friction force. This explains why this method is called heavy-ball momentum method. We refer to Figure 5 for an illustration.

Lemma 3.1. *For the equation (14), define total energy $J(x, v) = f(x) + \frac{1}{2}v^2$. Then, the energy dissipation satisfies*

$$\frac{dJ(X_t, V_t)}{dt} = -\gamma \|V_t\|^2$$

Proof.

$$\frac{d}{dt} J(X_t, V_t) = \langle \nabla f(X_t), V_t \rangle + \langle V_t, -\gamma \dot{X}_t - \nabla f(X_t) \rangle = -\gamma \|V_t\|^2.$$

□

This shows that the energy dissipation rate depends on the effective friction $\gamma \approx \frac{1-\beta}{\sqrt{\eta}}$, which explains how the choice of (η, β) influences the dynamics of HBM.

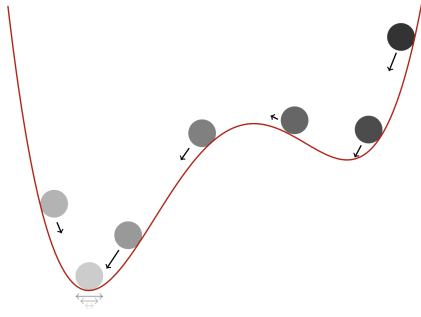


Figure 5: An illustration of the behavior of HBM dynamics.

The behavior of HBM dynamics. The continuous-time HBM-ODE and its physical interpretation provide valuable insight into the dynamics of HBM:

- HBM typically converges *non-monotonically*; in particular, a smaller friction coefficient γ leads to more pronounced oscillations.
- Momentum can help the algorithm escape saddle points, local minima, and flat plateaus.

The continuous-time analysis above indicates that, to realize these benefits, the momentum factor β to be close to 1, which aligns well with practical choice.

3.2 Acceleration for Strongly Convex Problem

Figure 6 shows the comparison of GD and HBM. One can see clearly that HBM converges to the minimizer with a better trajectory and the zig-zag issue is greatly alleviated. This property can lead substantial acceleration for optimizing strongly convex problem, improving the iteration complexity from $\kappa \log(1/\epsilon)$ to $\sqrt{\kappa} \log(1/\epsilon)$. For simplicity, we consider a quadratic objective $f(x) = \frac{1}{2}x^\top Hx$, whose minimizer is $x^* = 0$, to illustrate the underlying mechanism.

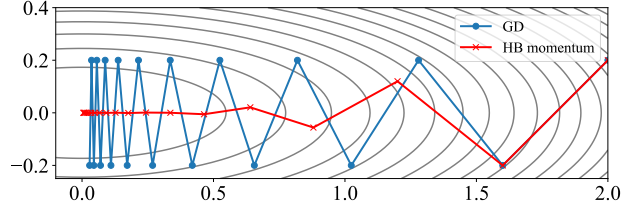


Figure 6: HBM can provide a better convergence direction without needing to reduce the learning rate.

First, analogous to the GD, the dynamics for HBM along different eigen directions is also decoupled. Still consider the decomposition $x_t = \sum_{j=1}^d x_j(t)u_j$. Then, multiplying both sides of Eq. (9) with u_j gives

$$\begin{aligned} x_j(t+1) &= x_j(t) - \eta\lambda_j x_j(t) + \beta(x_j(t) - x_j(t-1)) \\ &= (1 + \beta - \eta\lambda_j)x_j(t) - \beta x_j(t-1). \end{aligned}$$

Thus, the eigen component satisfies a second-order linear recurrence, whose solution is determined by the characteristic equation

$$\mu^2 - (1 + \beta - \eta\lambda_j)\mu + \beta = 0.$$

The two roots are given by

$$\mu_{j,\pm} = \frac{(1 + \beta - \eta\lambda_j) \pm \sqrt{\Delta_j}}{2}, \quad \Delta_j = (1 + \beta - \eta\lambda_j)^2 - 4\beta.$$

WLOG, assuming $\mu_{j,+} \neq \mu_{j,-}$. Then, the eigen component's dynamics follows

$$x_j(t) = C_+ \mu_{j,+}^t + C_- \mu_{j,-}^t,$$

where C_{\pm} are constants. To ensure convergence, (η, β) must satisfy

$$\sup_{j \in [d], \zeta \in \{+, -\}} |\mu_{j,\zeta}| < 1.$$

Accelerated convergence. Note that when $\Delta_j < 0$, the two roots are conjugates of each other and satisfy

$$|\mu_{j,+}| = |\mu_{j,-}| = \sqrt{\mu_{j,+}\mu_{j,-}} = \sqrt{\beta}.$$

Under this condition, we have $|x_j(t)| \leq C\sqrt{\beta}^t$. Surprisingly, the convergence rate of each direction is independent of the curvature λ_j . Consequently, if we can choose (η, β) such that $\max_{j \in [d]} \Delta_j < 0$, then

$$\|x_t - x^*\| = O(\beta^{t/2}). \quad (15)$$

The remaining task is to determine the smallest feasible value of β .

Note that

$$\begin{aligned} \Delta_j = (1 + \beta - \eta\lambda_j)^2 - 4\beta < 0 &\iff -1 \leq \frac{1 + \beta - \eta\lambda_j}{2\sqrt{\beta}} \leq 1 \\ &\iff (1 - \sqrt{\beta})^2 \leq \eta\lambda_j \leq (1 + \sqrt{\beta})^2. \end{aligned} \quad (16)$$

To satisfy this condition for all $j \in [d]$, it is required that

$$\eta\lambda_d \geq (1 - \sqrt{\beta})^2, \quad \eta\lambda_1 \leq (1 + \sqrt{\beta})^2.$$

Assuming equality in both cases, we obtain

$$\sqrt{\beta} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \quad \eta = \frac{2(1 + \beta)}{\lambda_1 + \lambda_d}.$$

Substituting this into (15) yields

$$\|x_t\| \leq C \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t. \quad (17)$$

Compared with GD, the dependence on the condition number is improved from κ to $\sqrt{\kappa}$.

Remark 3.2. We refer to <https://distill.pub/2017/momentum/> for a better visual explanation of how momentum works.

4 Nesterov Momentum

The improvement offered by the heavy-ball method (HBM) is limited to quadratic (or strongly convex) objectives. For general convex problems, both HBM and GD share the same convergence rate of $O(1/t)$. This naturally raises the question:

Can we achieve a faster rate than $O(1/t)$ by introducing momentum?

This question was fully answered by Nesterov in his seminal work [Nesterov, 1983]. Interestingly, Nesterov’s original motivation went beyond a specific algorithm—he aimed to understand the fundamental limits of all gradient-based methods. Specifically, he considered iterative schemes that rely solely on gradient information:

$$x_{t+1} = \mathcal{M}(\nabla f(x_t), \nabla f(x_{t-1}), \dots, \nabla f(x_0), x_0), \quad (18)$$

where \mathcal{M} denotes a general update rule. He then asked:

What is the fastest possible convergence rate that a gradient-based method can achieve when f is convex?

Nesterov provided a complete answer:

- The optimal convergence rate achievable by first-order (gradient-based) methods for convex functions is $O(1/t^2)$; this rate cannot be improved without using higher-order information.
- Moreover, he constructed an explicit method—now known as the **Nesterov Accelerated Gradient** (NAG) method—that attains this optimal rate:

$$\begin{aligned} y_t &= x_t + \beta_t(x_t - x_{t-1}), \\ x_{t+1} &= y_t - \eta \nabla f(y_t), \end{aligned} \quad (19)$$

where $\beta_t = \frac{t-1}{t+2}$. The term “accelerated gradient” is used due to the improved convergence.

Theorem 4.1 (Nesterov’s Theorem). *Let f be a L -smooth convex function. If x_t is generated by the NAG scheme with learning rate $\eta \leq \frac{1}{L}$, then*

$$f(x_t) - f(x^*) \leq \frac{\|x_0 - x^*\|^2}{\eta(t+1)^2}$$

The variant in deep learning. The NAG method becomes popular in deep learning following the influential work [Sutskever et al., 2013], which demonstrates its superior empirical performance compared to the heavy-ball momentum. In particular, [Sutskever et al., 2013] reformulated NAG to highlight its close connection with HBM:

$$\begin{aligned} v_{t+1} &= \beta v_t - \eta \nabla f(x_t + \beta v_t) \\ x_{t+1} &= x_t + v_{t+1} \end{aligned}$$

Figure 7 provides a visual comparison between Nesterov momentum and heavy ball momentum. The key distinction is that NAG evaluates the gradient at the **extrapolated point** $y_t = x_t + \beta v_t$ rather than at the current iterate x_t .

In deep learning practice, the momentum coefficient β is typically fixed to a constant value (e.g., 0.9 or 0.99). The choice $\beta_t = (t-1)/(t+2)$ is designed for convex optimization. However, in deep learning, objective functions are always non-convex and the delicate scheduling is not necessary.

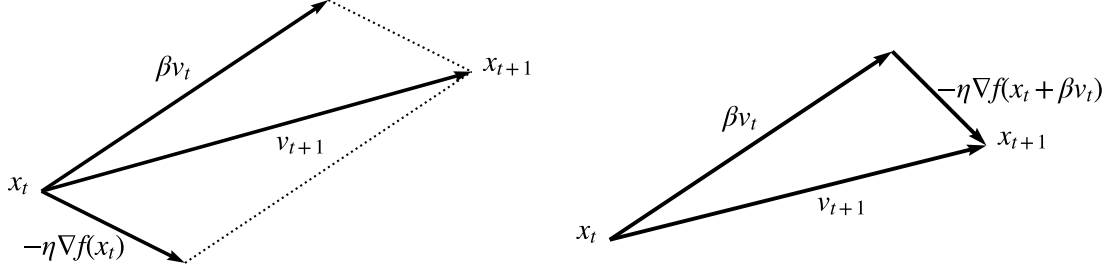


Figure 7: The comparison between heavy-ball momentum (left) and Nesterov momentum (right).

4.1 A Continuous-Time Analysis

The proof of Theorem 4.1 is technically involved and depends sensitively on the choice of β_t . Here we present a continuous-time argument [Su et al., 2016] that offers a simpler and more intuitive understanding.

Note that

$$\begin{aligned} x_{t+1} - x_t &= y_t - \eta \nabla f(y_t) - x_t \\ &= \frac{t-1}{t+2}(x_t - x_{t-1}) - \eta \nabla f(y_t) \\ &= x_t - x_{t-1} - \frac{3}{t+2}(x_t - x_{t-1}) - \eta \nabla f(y_t), \end{aligned}$$

which can be rephrased as

$$\frac{x_{t+1} - 2x_t + x_{t-1}}{(\sqrt{\eta})^2} = -\frac{3}{(t+2)\sqrt{\eta}} \frac{x_t - x_{t-1}}{\sqrt{\eta}} - \nabla f(y_t). \quad (20)$$

Let $\tau = t\sqrt{\eta}$ and $X(t\sqrt{\eta}) = x_t$. Then, the above

$$\frac{\ddot{X}(\tau)(\sqrt{\eta})^2}{(\sqrt{\eta})^2} + o(\sqrt{\eta}) = -\frac{3}{(t+2)\sqrt{\eta}} \dot{X}(\tau) - \nabla f(X(\tau)) + o(\sqrt{\eta}).$$

Taking $\eta \rightarrow 0$ and considering the leading term, we obtain the limiting ODE as follows

$$\ddot{X} = -\frac{3}{\tau} \dot{X} - \nabla f(X)$$

The above ODE is analogous to the HBM ODE (13). The difference is that in HBM, the friction factor is a constant, while in NAG the friction factor $3/\tau$ decays to zero as $\tau \rightarrow \infty$.

For brevity, we will still use t to denote the continuous time.

Theorem 4.2. Suppose $\dot{X}_t = 0$. Then,

$$f(X_t) - f^* \leq \frac{2\|X_0 - x^*\|^2}{t^2}$$

Proof. Consider the energy functional defined as

$$\mathcal{E}(t) := t^2 (f(X_t) - f^*) + 2 \left\| X_t + \frac{t}{2} \dot{X}_t - x^* \right\|^2$$

whose time derivative is

$$\dot{\mathcal{E}} = 2t(f(X) - f^*) + t^2 \langle \nabla f, \dot{X} \rangle + 4 \left\langle X + \frac{t}{2} \dot{X} - x^*, \frac{3}{2} \dot{X} + \frac{t}{2} \ddot{X} \right\rangle$$

Substituting $3\dot{X}/2 + t\ddot{X}/2$ with $-t\nabla f(X)/2$, (3.3) gives

$$\dot{\mathcal{E}} = 2t(f(X) - f^*) + 4 \left\langle X - x^*, -\frac{t}{2} \nabla f(X) \right\rangle = 2t(f(X) - f^*) - 2t \langle X - x^*, \nabla f(X) \rangle \leq 0,$$

where the inequality follows from the convexity of f . Hence by monotonicity of \mathcal{E} and non-negativity of $2 \left\| X + t\dot{X}/2 - x^* \right\|^2$, the gap obeys $f(X_t) - f^* \leq \mathcal{E}(t)/t^2 \leq \mathcal{E}(0)/t^2 = 2 \|x_0 - x^*\|^2 / t^2$. \square

References

- [Nesterov, 1983] Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. In *Dokl akad nauk Sssr*, volume 269, page 543.
- [Polyak, 1964] Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17.
- [Su et al., 2016] Su, W., Boyd, S., and Candès, E. J. (2016). A differential equation for modeling nesterov’s accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(153):1–43.
- [Sutskever et al., 2013] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147. PMLR.