Breaking the Convergence Barrier: Optimization via Fixed-Time Convergent Flows

Param Budhraja[*,1] **Mayank Baranwal**[*,2] **Kunal Garg**[3] and **Ashish Hota**[1]

[1] Indian Institute of Technology Kharagpur
[2] Tata Consultancy Services Research, Mumbai
[3] University of California, Santa Cruz
budhrajap99@iitkgp.ac.in, baranwal.mayank@tcs.com, kunalgarg@ucsc.edu, ahota@ee.iitkgp.ac.in

Abstract

Accelerated gradient methods are the cornerstones of large-scale, data-driven optimization problems that arise naturally in machine learning and other fields concerning data analysis. We introduce a gradient-based optimization framework for achieving acceleration, based on the recently introduced notion of fixed-time stability of dynamical systems. The method presents itself as a generalization of simple gradient-based methods suitably scaled to achieve convergence to the optimizer in a fixed-time, independent of the initialization. We achieve this by first leveraging a continuous-time framework for designing fixed-time stable dynamical systems, and later providing a consistent discretization strategy, such that the equivalent discrete-time algorithm tracks the optimizer in a practically fixed number of iterations. We also provide a theoretical analysis of the convergence behavior of the proposed gradient flows, and their robustness to additive disturbances for a range of functions obeying strong convexity, strict convexity, and possibly nonconvexity but satisfying the Polyak-Łojasiewicz inequality. We also show that the regret bound on the convergence rate is constant by virtue of the fixed-time convergence. The hyperparameters have intuitive interpretations and can be tuned to fit the requirements on the desired convergence rates. We validate the accelerated convergence properties of the proposed schemes on a range of numerical examples against the state-of-the-art optimization algorithms. Our work provides insights on developing novel optimization algorithms via discretization of continuous-time flows.

Introduction and Related Work

Optimization algorithms lie at the heart of modern artificial intelligence and machine learning techniques (Sra, Nowozin, and Wright 2012). In most applications, fast and efficient algorithms are desired for solving the optimization problem at hand. This is particularly true in machine learning applications where large data sets lead to larger problem instances and potentially larger computational time. As a result, stochastic gradient descent (SGD), its variants such as mini-batch SGD (Shalev-Shwartz and Ben-David 2014), Adam (Kingma and Ba 2015), momentum-based, and accelerated stochastic methods have emerged as popular choices (Huo et al. 2018; Li, Fang, and Lin 2020).

In developing accelerated optimization algorithms, the discrete-time framework often proves non-intuitive and restrictive from an analytical standpoint. In contrast, continuous-time algorithms provide better intuition, and simpler and elegant proofs are often obtained by leveraging the tools of Lyapunov stability theory. Indeed, the connection between ordinary differential equations and optimization has been recognized for several decades. For instance, (Brown and Bartholomew-Biggs 1989) is one of the early works that leveraged the continuous-time framework to develop faster discrete-time algorithms. Similarly, the continuous-time version of gradient descent, termed *gradient flow* (GF) dynamics, was analyzed in (Su, Boyd, and Candès 2016). In (Wibisono, Wilson, and Jordan 2016), the family of Bregman-Lagrangians was used to generate second-order Lagrangian flows and exponential convergence rates were established. Despite much progress, there remain two main limitations for continuous-time algorithms: (1) most of the analysis has focused on asymptotic and exponential convergence, i.e., convergence as time tends to infinity; and (2) there have been few systematic studies on developing discrete-time implementations such that the accelerated convergence properties of the continuous-time algorithm are preserved.

In this paper, we focus on continuous-time (accelerated) gradient flow dynamics with fixed-time convergence guarantees. The notion of finite-time stability (FTS), which is a precursor to the notion of fixed-time stability, was proposed in the seminal work (Bhat and Bernstein 2000). A system is said to be finite-time stable if the trajectories converge to the equilibrium in a finite amount of time, called the *settling time*. The settling time may depend on the initial conditions, and can potentially grow unbounded as the initial conditions go farther away from the equilibrium point. Fixed-time stability (FxTS), on the other hand, is a stronger notion, which requires the settling time to be uniformly bounded for all initial conditions, i.e., convergence within a *fixed* time can be guaranteed (Polyakov 2011).

In the recent few years, continuous-time optimization methods under the notions of FTS and FxTS have gained significant interest. In (Cortés 2006), the author proposed a normalized version of GF and proved its finite-time stability. For convex optimization problems with equality constraints, the authors in (Chen and Ren 2018), designed discontinu-

[*]These authors contributed equally.

ous dynamical systems having the property of finite-time convergence. Recently in (Garg and Panagou 2021), FxTS gradient flows were proposed for unconstrained, constrained and min-max optimization problems. However, the aforementioned works only guarantee improved convergence in the continuous-time domain and do not provide a discrete-time implementation having accelerated convergence.

Recently, (Polyakov, Efimov, and Brogliato 2019) introduced the notion of *consistent* discretization for finite, and fixed-time stable dynamical systems. In particular, they proposed an implicit discretization scheme that preserves the convergence behavior of the continuous-time system. However, these results are of little use for the optimization community, since, a) the requirement of the dynamics being homogeneous cannot be satisfied unless the equilibrium point, in this case, the optimizer, is known, and b) implicit discretization schemes are not easy to implement, thus, making it difficult to use these schemes for iterative methods. The authors in (Benosman, Romero, and Cherian 2020) showed that the FTS flow, re-scaled gradient flow, and signed-gradient flow, all with a finite-time convergence, when discretized using various explicit schemes, such as Euler discretization or Runge-Kutta method, preserve the convergence behavior in the discrete-time, i.e., the minimizer could be computed within a finite number of iterations for a class of convex optimization problems. The authors evaluated their proposed methods for training neural networks and showed a significant improvement in the performance.

In this paper, a consistent discretization of FxTS-GF is proposed using the method proposed in (Garg et al. 2021). We then show the robustness of FxTS-GF to vanishing disturbance, under the assumption of Polyak-Łojasiewicz (PL) inequality. Note that a function satisfying the PL-inequality can be nonconvex and that PL-inequality is a weaker assumption than strong convexity. It was shown in (Karimi, Nutini, and Schmidt 2016) that PL inequality is one of the weakest assumptions under which linear convergence can be proven, which was earlier proven under the assumption of strong convexity. We then analyze the static regret of FxTS-GF and show that it is bounded by a constant. Finally, numerical experiments are conducted to compare the performance of FxTS-GF with state-of-the-art optimization algorithms. The proposed algorithm achieves lower training loss than traditional optimization methods, which also translates to better generalization on test instances.

Notation: The set of all real numbers is denoted by $\mathbb{R}$. The set of all positive reals is denoted by $\mathbb{R}_{>0}$. The zero vector belonging to $\mathbb{R}^n$ is denoted by $\mathbf{0}$. For $x \in \mathbb{R}^n$, its transpose is represented by $x^\intercal$. Unless otherwise specified, $\|\cdot\|$ denotes the Euclidean norm. The set of all functions $f : U \to V$, where $U \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{R}^m$, which are k-times continuously differentiable is denoted by $C^k(U, V)$. The set of functions $f : U \to V$ which are continuously differentiable with locally Lipschitz continuous gradient on U is denoted by $C^{loc}_{1,1}(U, V)$. For compactness, a function's argument might be omitted, whenever clear from the context. For $f \in C^1(\mathbb{R}^n, \mathbb{R})$, its gradient is denoted by ∇f. A set-valued mapping $\mathcal{F} : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ maps every $x \in \mathbb{R}^n$ to a set of $\mathbb{R}^m$.

Preliminaries

We start by presenting the problem setting, required assumptions and the fixed-time stability property of gradient flow dynamics. Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$. We make the following assumptions on the function f.

Assumption 1. *The function f attains the minimum value $f^\star > -\infty$ at $x^\star \in \mathbb{R}^n$, i.e., $f^\star := f(x^\star) > -\infty$.*

Assumption 2. *The function $f \in C^{loc}_{1,1}(\mathbb{R}^n, \mathbb{R})$ has a unique minimizer $x = x^\star$ and satisfies Polyak-Lojasiewicz (PL) inequality, or is gradient dominated, i.e. there exists $\mu > 0$, such that $\forall x \in \mathbb{R}^n$,*

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu(f(x) - f^\star). \tag{2}$$

Under the assumption of gradient dominance it was shown in ((Karimi, Nutini, and Schmidt 2016), Theorem 2) that the function $f(x)$ has a quadratic growth, i.e.

$$f(x) - f^\star \geq \frac{\mu}{2}\|x - x^\star\|^2, \tag{3}$$

for all $x \in \mathbb{R}^n$.

We now formally discuss the notion of fixed-time stability (FxTS). Consider the dynamical system

$$\dot{x} = g(x), \tag{4}$$

where $x \in \mathbb{R}^n$, $g(\mathbf{0}) = 0$ and let solution to (4) exist, is unique, and continuous for any initial condition $x(0) \in \mathbb{R}^n$, for all $t \geq 0$. As introduced in (Polyakov 2011), an equilibrium point of (4) is called as FxTS if (i) it is Lyapunov stable, and (ii) there exists a fixed-time $T < \infty$ (also known as settling time), such that for all initial conditions $x(0) \in \mathbb{R}^n$, the solution of (4) satisfies $x(t) = 0$ for all $t \geq T$. The following lemma provides sufficient conditions for FxTS of the origin.

Lemma 1 ((Polyakov 2011))**.** *Suppose there exists a positive definite, radially unbounded function $V \in C^1(\mathcal{D}, \mathbb{R})$, where $\mathcal{D} \subset \mathbb{R}^n$ is a neighbourhood of origin, such that*

$$\dot{V}(x) \leq -pV(x)^\alpha - qV(x)^\beta, \quad \forall x \in \mathcal{D} \setminus \{\mathbf{0}\}, \tag{5}$$

where $p, q > 0$, $\alpha \in (0, 1)$ and $\beta > 1$. Then, the origin of the system (4) is fixed-time stable with a settling time

$$T \leq \frac{1}{p(1-\alpha)} + \frac{1}{q(\beta - 1)}. \tag{6}$$

Fixed-Time Stable Gradient Flow (FxTS-GF)

We now introduce the following gradient flow dynamics and establish the FxTS of the optimizer. Specifically, consider the dynamics

$$\dot{x} = \begin{cases} -c_1 \frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_1-2}{p_1-1}}} - c_2 \frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_2-2}{p_2-1}}}, & \nabla f(x) \neq \mathbf{0}, \\ 0, & \nabla f(x) = \mathbf{0}, \end{cases} \tag{7}$$

where $c_1, c_2 > 0$, $p_1 > 2$ and $p_2 \in (1, 2)$. The flow (7), first introduced in (Garg and Panagou 2021), is henceforth called

as FxTS-GF. In (Garg and Panagou 2021), it was shown that the flow (7) converges to the optimizer $x^\star$ of (1) within a fixed time, irrespective of the initial condition, under the Assumptions 1 and 2. We now provide a self-contained proof of FxTS of (7) with a different candidate Lyapunov function.

Theorem 1 (FxTS-GF). *Suppose the function f satisfies Assumptions 1 and 2. Then, the flow given by* (7) *converges to the optimizer $x^\star$ in a fixed time for all $x(0) \in \mathbb{R}^n$.*

Proof. The existence and uniqueness of a continuous solution of (7) for initial conditions at all times was proved in (Garg and Panagou 2021, Proposition 1). Thus, we proceed with the proof of FxTS of the optimal point $x^\star$. Consider Lyapunov candidate $V(x) = f(x) - f^\star$. As $f^\star$ is the minimum value of f and $x^\star$ is the unique minimizer of f, it holds that $V(x) > 0$ for all $x \neq x^\star$. The time derivative of V is given by

$$\begin{aligned}\dot{V}(x) &= \nabla f(x)^\intercal \dot{x} \\ &= \nabla f(x)^\intercal \left(-c_1 \frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_1-2}{p_1-1}}} - c_2 \frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_2-2}{p_2-1}}} \right) \\ &= -c_1 \|\nabla f(x)\|^{\frac{p_1}{p_1-1}} - c_2 \|\nabla f(x)\|^{\frac{p_2}{p_2-1}}.\end{aligned}$$

Using inequality (2), we get

$$\begin{aligned}\dot{V} &\leq -c_1 (2\mu(f-f^\star))^{\frac{p_1}{2(p_1-1)}} - c_2 (2\mu(f-f^\star))^{\frac{p_2}{2(p_2-1)}} \\ &= -c_1 (2\mu)^{\frac{p_1}{2(p_1-1)}} V^{\frac{p_1}{2(p_1-1)}} - c_2 (2\mu)^{\frac{p_2}{2(p_2-1)}} V^{\frac{p_2}{2(p_2-1)}}.\end{aligned}$$

Define $p \coloneqq c_1 (2\mu)^{\frac{p_1}{2(p_1-1)}}$, $q \coloneqq c_2 (2\mu)^{\frac{p_2}{2(p_2-1)}}$. Since $p_1 > 2$ and $p_2 \in (1,2)$, we have $\alpha \coloneqq \frac{p_1}{2(p_1-1)} \in (0,1)$ and $\beta \coloneqq \frac{p_2}{2(p_2-1)} > 1$. Thus, the conditions for Lemma 1 are satisfied, and it follows that the equilibrium point $x^\star$ of (7) is FxTS with settling time $T \leq \frac{1}{p(1-\alpha)} + \frac{1}{q(\beta-1)}$. □

Robustness Analysis

The above result establishes the FxTS of the optimizer under flow dynamics (7). We now prove a stronger result that the FxTS property is preserved when the dynamics (7) is subjected to additive noises or disturbances, under mild assumptions on the noise or the disturbances. This is particularly important in data-driven learning where only a noisy estimate of the gradient is available. Specifically, we consider the following dynamical system:

$$\dot{x} = -c_1 \frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_1-2}{p_1-1}}} - c_2 \frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_2-2}{p_2-1}}} + \varepsilon(x), \quad (8)$$

where $\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ is the additive noise term. We assume that the noise $\varepsilon(\cdot)$ is a vanishing disturbance and it satisfies the following assumption.

Assumption 3. *There exists $l > 0$ such that the noise term satisfies $\|\varepsilon(x)\| \leq l\|x - x^\star\|^2$, for all $x \in \mathbb{R}^n$.*

Observe that Assumption 2 along with (3) yield

$$\|x - x^\star\|^2 \leq \frac{1}{\mu^2} \|\nabla f(x)\|^2.$$

Together with Assumption 3, we obtain the following bound on the noise ε:

$$\|\varepsilon(x)\| \leq \bar{l} \|\nabla f(x)\|^2, \quad (9)$$

where $\bar{l} = \frac{l}{\mu^2}$. Note that Assumption 3 also implies that the point $x^\star$ is an equilibrium point of the perturbed flow in (8). We now show that the FxTS property of $x^\star$ is robust to additive disturbance satisfying Assumption 3.

Theorem 2 (Robustness of FxTS-GF). *Under Assumptions 1, 2, 3, if c_1, c_2, p_2 are chosen so that $4\mu^2 \min\{c_1, c_2\} > l$ and $1 < p_2 \leq \frac{3}{2}$, then the equilibrium point $x^\star$ is FxTS for the flow given by* (8).

Proof. Consider the Lyapunov candidate $V(x) = f(x) - f^\star$ as before. The time derivative of V is given by

$$\dot{V}(x) = -c_1 \|\nabla f(x)\|^{\frac{p_1}{p_1-1}} - c_2 \|\nabla f(x)\|^{\frac{p_2}{p_2-1}} + \nabla f^\intercal \varepsilon(x).$$

We define the following constants for the ease notation: $\alpha_1 = \frac{p_1}{p_1-1}$, $\beta_1 = \frac{p_2}{p_2-1}$. Note that $\alpha_1 \in (0,2)$ and $\beta_1 > 2$. Now, we prove that

$$\dot{V}(x) \leq -(c_1 - \bar{l}) \|\nabla f(x)\|^{\alpha_1} - (c_2 - \bar{l}) \|\nabla f(x)\|^{\beta_1}, \quad (10)$$

for all $x \in \mathbb{R}^n$. Using Assumption 3 and triangle inequality, we obtain that $\nabla f^\intercal \varepsilon(x) \leq \|\nabla f(x)\| \|\varepsilon(x)\| \leq \bar{l} \|\nabla f(x)\|^3$. Thus, it follows that

$$\dot{V}(x) \leq -c_1 \|\nabla f(x)\|^{\alpha_1} - c_2 \|\nabla f(x)\|^{\beta_1} + \bar{l} \|\nabla f(x)\|^3. \quad (11)$$

Define $S = \{x \mid \|\nabla f(x)\| \leq 1\}$ and consider the two cases, namely, when $x \in S$ and $x \notin S$.

First, consider the case when $x \notin S$. Re-arranging the right-hand side of (11), we obtain:

$$\begin{aligned}\dot{V}(x) \leq &-c_1 \|\nabla f(x)\|^{\alpha_1} - (c_2 - \bar{l}) \|\nabla f(x)\|^{\beta_1} \\ &+ \bar{l} \left(\|\nabla f(x)\|^3 - \|\nabla f(x)\|^{\beta_1} \right).\end{aligned}$$

Since $p_2 \in \left(1, \frac{3}{2}\right]$, it follows that $\beta_1 > 3$. Furthermore, for all $x \notin S$, it holds that $\|\nabla f(x)\| > 1$ and thus, it follows that $\|\nabla f(x)\|^3 - \|\nabla f(x)\|^{\beta_1} \leq 0$. Hence, we obtain that

$$\begin{aligned}\dot{V}(x) &\leq -c_1 \|\nabla f(x)\|^{\alpha_1} - (c_2 - \bar{l}) \|\nabla f(x)\|^{\beta_1} \\ &\leq -(c_1 - \bar{l}) \|\nabla f(x)\|^{\alpha_1} - (c_2 - \bar{l}) \|\nabla f(x)\|^{\beta_1}.\end{aligned}$$

Next, consider the case when $x \in S$. Re-arranging the right-hand side of (11), we obtain

$$\begin{aligned}\dot{V}(x) \leq &-(c_1 - \bar{l}) \|\nabla f(x)\|^{\alpha_1} - c_2 \|\nabla f(x)\|^{\beta_1} \\ &+ \bar{l} \left(\|\nabla f(x)\|^3 - \|\nabla f(x)\|^{\alpha_1} \right).\end{aligned}$$

For all $x \in S$, we have $\|\nabla f(x)\|^3 - \|\nabla f(x)\|^{\alpha_1} \leq 0$ since $\alpha_1 \in (0,2)$ and $\|\nabla f(x)\| \leq 1$. Thus, it holds that

$$\begin{aligned}\dot{V}(x) &\leq -(c_1 - \bar{l}) \|\nabla f(x)\|^{\alpha_1} - c_2 \|\nabla f(x)\|^{\beta_1} \\ &\leq -(c_1 - \bar{l}) \|\nabla f(x)\|^{\alpha_1} - (c_2 - \bar{l}) \|\nabla f(x)\|^{\beta_1}.\end{aligned}$$

Thus, it follows from Assumption 2 that for all $x \in \mathbb{R}^n$

$$\begin{aligned}\dot{V}(x) \leq &-(c_1 - \bar{l}) (2\mu(f(x) - f^\star))^{\frac{p_1}{2(p_1-1)}} \\ &- (c_2 - \bar{l}) (2\mu(f(x) - f^\star))^{\frac{p_2}{2(p_2-1)}}.\end{aligned}$$

Define $p := (c_1 - \bar{l})(2\mu)^{\frac{p_1}{2(p_1-1)}}, q := (c_2 - \bar{l})(2\mu)^{\frac{p_2}{2(p_2-1)}}$, and $\alpha := \frac{p_1}{2(p_1-1)}, \beta := \frac{p_2}{2(p_2-1)}$ so that we have

$$\dot{V}(x) \leq -pV(x)^{\alpha} - qV(x)^{\beta},$$

where $p, q > 0$, $\alpha \in (0, 1)$ and $\beta > 1$. Using Lemma 1, we obtain that the equilibrium point $x^{\star}$ is fixed-time stable for the perturbed FxTS-GF flow given by (8). □

Thus, FxTS-GF flow in (7) is robust against a class of vanishing additive disturbances.

Regret Analysis

The regret analysis is used to evaluate the effectiveness of an algorithm (Sun and Hu 2020). In this section, we analyze the regret of FxTS-GF (7) in the offline setting. The static regret is defined as the accumulated difference between the objective function computed according to the state of the algorithm and the objective function computed according to the best fixed-point that minimizes the accumulated objective function. The regret at any time $T > 0$ is given by

$$\mathcal{R}_S(T, x_0) = \int_0^T (f(x(t)) - f(x^{\star}))\, dt,$$

where $x(0) = x_0$ is the initial condition. Note that in the offline setting the dynamic and static regret are equivalent. Also, observe that static regret is the time integral of the Lyapunov candidate used in the Theorem 1.

Theorem 3 (Regret Bound)**.** *Under the Assumptions 1 and 2, the static regret of the flow FxTS-GF is bounded by a constant $l_1 + l_2$, with*

$$l_1 = \frac{1}{p(2-\alpha)}, \quad l_2 = \frac{(1 + V(0)^{\beta-1})^{\frac{\beta-2}{\beta-1}} - 1}{qV(0)^{\beta-2}(\beta-2)},$$

where $V(0) = f(x_0) - f^{\star}$.

Proof. Considering the Lyapunov candidate $V(x) = f(x) - f^{\star}$, we have the following bound on its time derivative $\dot{V}(x)$, which was proved in Theorem 1,

$$\dot{V}(x) \leq -pV(x)^{\alpha} - qV(x)^{\beta}, \tag{12}$$

where $p = c_1(2\mu)^{\frac{p_1}{2(p_1-1)}}$, $q = c_1(2\mu)^{\frac{p_2}{2(p_2-1)}}$, $\alpha = \frac{p_1}{2(p_1-1)}$, and $\beta = \frac{p_2}{2(p_2-1)}$. When $V(t) > 1$ we use $\dot{V}(t) \leq -qV(t)^{\beta}$ and when $V(t) \leq 1$ we use $\dot{V}(t) \leq -pV(t)^{\alpha}$. The main idea is to apply these two approximations to obtain upper bound on $V(t)$, which in turn bounds the regret. We define the following constants:

$$T_1 := \frac{1}{p(1-\alpha)}, \quad T_2 := \frac{1}{q(\beta-1)}.$$

We divide the proof into two parts. First we analyze the case $V(x(0)) > 1$ and then the case $V(x(0)) \leq 1$.

Case 1: Consider $V(x(0)) > 1$, i.e. the initial conditions x_0 is such that $f(x_0) - f^{\star} > 1$. For $t \leq T_2$ we have

$$V(x(t)) \leq \frac{V(x(0))}{(1 + qV(x(0))^{\beta-1}(\beta-1)t)^{\frac{1}{\beta-1}}}. \tag{13}$$

For $T_2 \leq t \leq T_1 + T_2$, we get

$$V(x(t)) \leq (1 - p(1-\alpha)(t - T_2))^{\frac{1}{1-\alpha}}. \tag{14}$$

For $t \geq T_1 + T_2$, $V(x(t)) = 0$, i,e, the solution converges to optimizer. We integrate both sides of the inequality (13) to get the following for all $T \in [0, T_1]$:

$$\mathcal{R}_S(T, x_0) \leq \frac{(1 + q(\beta-1)V(x(0))T)^{\frac{\beta-2}{\beta-1}} - 1}{q(\beta-2)V(x(0))^{\beta-2}}.$$

Similarly for $T_2 \leq T \leq T_1 + T_2$, using both the inequalities (13) and (14), we get

$$\mathcal{R}_S(T, x_0) \leq l_2 + \frac{1 - (1 - p(1-\alpha)(T - T_2))^{\frac{2-\alpha}{1-\alpha}}}{p(2-\alpha)}.$$

As $V(t) = 0$ for $t \geq T_1 + T_2$, we get $\mathcal{R}_S(T, x_0) \leq l_1 + l_2$ for $T \geq T_1 + T_2$.

Case 2: Consider $V(x(0)) \leq 1$, i.e. the initial conditions x_0 is such that $f(x_0) - f^{\star} \leq 1$. In this scenario also we use the same procedure. For $0 \leq T \leq T_1$ we get

$$\mathcal{R}_S(T, x_0) \leq \frac{1 - (1 - p(1-\alpha)T)^{\frac{2-\alpha}{1-\alpha}}}{p(2-\alpha)}.$$

For $T \geq T_1$ we get $\mathcal{R}_S(T, x_0) \leq l_1$. Note that we can also say that $\mathcal{R}_S(T, x_0) \leq l_1 + l_2$ for all $T \geq 0$ and for all $V(0)$. □

Observe that instead of bounding by a constant we can also bound the regret $\mathcal{R}_S(T, x_0)$ by a step function as:

$$\mathcal{R}_S(T, x_0) = \begin{cases} l_1 & \text{if, } f(x_0) - f^{\star} \leq 1,\ T \geq 0, \\ l_1 & \text{if, } f(x_0) - f^{\star} > 1,\ T \leq T_1, \\ l_1 + l_2 & \text{if, } f(x_0) - f^{\star} > 1,\ T > T_1. \end{cases}$$

Discretization of FxTS-GF

In practice, continuous-time dynamical systems can be implemented using iterative discrete-time approximations. Following the work in (Garg et al. 2021; Benosman, Romero, and Cherian 2020), in this section, we provide the analysis of the Euler-discretization of (7), and show that when the FxTS-GF (7) is discretized using Euler discretization, it leads to a consistent discretization. We use the following result from (Garg et al. 2021).

Lemma 2 (**Consistent Discretization**)**.** *Consider the following differential inclusion:*

$$\dot{x} \in \mathcal{F}(x), \tag{15}$$

where $\mathcal{F} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is an upper semi-continuous set-valued map, taking non-empty, convex and compact values, with $0 \in \mathcal{F}(\bar{x})$ for some $\bar{x} \in \mathbb{R}^n$. Assume that there exists a positive definite, radially unbounded $V : \mathbb{R}^n \to \mathbb{R}$ such that $V(\bar{x}) = 0$ satisfying (5) *with $\alpha = 1 - \frac{1}{\xi}, \beta = 1 + \frac{1}{\xi}$ for some $\xi > 1$. If the function V satisfies $V(x) \geq m\|x - \bar{x}\|^2$ for all $x \in \mathbb{R}^n$, where $m > 0$ and $\bar{x}$ is the equilibrium point of* (15)*, then, for all $x_0 \in \mathbb{R}^n$ and $\epsilon > 0$, there exists $\eta^* > 0$ such that for any $\eta \in (0, \eta^*]$, the following holds:*

$$\|x_k - \bar{x}\| < \begin{cases} \frac{1}{\sqrt{m}} \left(\sqrt{\frac{p}{q}} \tan\left(\frac{\pi}{2} - \frac{\sqrt{pq}}{\xi} \eta k \right) \right)^{\frac{\xi}{2}} + \epsilon, & k \leq k^*; \\ \epsilon, & k > k^*, \end{cases} \tag{16}$$

where $k^* = \left\lceil \frac{\xi\pi}{2\eta\sqrt{pq}} \right\rceil$ *and* x_k *is a solution of the forward-Euler discretization of* (15)*:*

$$x_{k+1} \in x_k + \eta \mathcal{F}(x_k), \tag{17}$$

where $\eta > 0$ *is the time-step, starting from the point* x_0*.*

Thus, in order to prove that an Euler discretization scheme of (7) leads to a consistent discretization, it is sufficient to show that (7) satisfies the conditions of Lemma 2.

Lemma 3. *If* p_1, p_2 *satisfy*

$$2 + \frac{1}{p_1 - 2} = \frac{1}{2 - p_2}, \tag{18}$$

with $\frac{3}{2} < p_2 < 2$*, then, the function* $V(x) = (f(x) - f^\star)$ *satisfies conditions of Lemma 2.*

Proof. Consider the Lyapunov candidate $V(x) = (f(x) - f^\star)$. Its time derivative along the trajectories of (7) reads:

$$\dot{V}(x) = -c_1 \nabla f(x)^\intercal \frac{\nabla f(x)}{\|\nabla f\|^{\frac{p_2-2}{p_2-1}}} - c_2 \nabla f(x)^\intercal \frac{\nabla f(x)}{\|\nabla f\|^{\frac{p_1-2}{p_1-1}}}.$$

Following the analysis in Theorem 1, it follows that

$$\dot{V} \leq -pV^{\frac{p_1}{2(p_1-1)}} - qV^{\frac{p_2}{2(p_2-1)}},$$

where $p = c_1(2\mu)^{\frac{p_1}{2(p_1-1)}}$ and $q = c_2(2\mu)^{\frac{p_2}{2(p_2-1)}}$. Note that under the condition (18), it holds that there exists $\xi = -\frac{2p_2-2}{p_2-2} = \frac{2p_1-2}{p_1-2} > 2$, so that the above equation reads

$$\dot{V}(x) = -pV(x)^{1+\frac{1}{\xi}} - qV(x)^{1-\frac{1}{\xi}}.$$

Thus, the candidate function V satisfies the conditions of Lemma 1 with $\alpha = 1 - \frac{1}{\xi}$ and $\beta = 1 + \frac{1}{\xi}$. Finally, note that under Assumption 2, it holds that $\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu(f(x) - f^\star) \geq 2\mu^2\|x - x^\star\|^2$, i.e., the function V has quadratic growth, and thus, the function V is radially unbounded, satisfying all the conditions of Lemma 2. □

Theorem 4. *Assume that the functions* f *satisfy Assumptions 1-2. Consider the discrete-time system*

$$x_{k+1} = x_k - \eta c_1 \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|^{\frac{p_1-2}{p_1-1}}} - \eta c_2 \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|^{\frac{p_2-2}{p_2-1}}}, \tag{19}$$

obtained from discretizing the dynamics in (7) *using Euler's method with time step* $\eta > 0$*, where* p_1, p_2 *satisfy* (18)*. Then, for all* $\epsilon > 0$*, there exists* $\eta^* > 0$ *such that for all* $\eta \in (0, \eta^*]$*, the trajectories of* (19) *satisfy*

$$\|x_k - x^\star\| \leq \begin{cases} \frac{1}{\sqrt{2}\mu}\left(\sqrt{\frac{p}{q}}\tan\left(\frac{\pi}{2} - \frac{\eta k\sqrt{pq}}{2\mu}\right)\right)^{\mu} + \epsilon\,; & k \leq k^*, \\ \epsilon\,; & k > k^*, \end{cases} \tag{20}$$

where $k^* = \lceil \frac{\mu\pi}{\sqrt{pq}\eta} \rceil$*, and* $a, b, c_1, \mu > 0$*.*

Proof. The proof is based on Lemma 2. First, note that per Lemma 3, there exists a function V, namely $V(x) = (f(x) - f^\star)$, that satisfies the conditions of Lemma 2 with $\xi = \frac{2p_1-2}{p_1-2}$ and $\beta = 2\mu^2$. Next, note that the right-hand side of (7) is single-valued and continuous, and thus, $\mathcal{F}(x) = -c_1\frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_1-2}{p_1-1}}} - c_2\frac{\nabla f(x)}{\|\nabla f(x)\|^{\frac{p_2-2}{p_2-1}}}$ satisfies the assumptions of Lemma 2 with $\bar{x} = x^\star$. Thus, it holds that all the conditions of Lemma 2 are satisfied and the proof is complete. □

FxTS Gradient Flow with Momentum

Training of neural networks entails computing gradients on mini-batches. These gradients are not exact and serve as only *noisy* estimates of the true gradient of the loss function, leading to optimization algorithms not descending in optimal directions. This can be partially alleviated using the momentum method (Polyak 1964), which employs exponentially weighted averages to provide a better estimate of the true gradient. Gradient descent with momentum is defined by:

$$\begin{aligned} v_t &= \beta v_{t-1} + (1-\beta)\nabla f(x_{t-1}) \\ x_t &= x_{t-1} - \alpha v_t, \end{aligned} \tag{21}$$

where $\alpha > 0$ and $\beta \in [0, 1)$. Here x_t represents the t^{th}-iterate of the state x. In the limit that α is sufficiently small, the derivative $\dot{x}(t)$ can be approximated as $\dot{x}(t) \approx (x_t - x_{t-1})/\alpha$. Using this analogy, the continuous-time variant of the momentum method can be expressed as:

$$\begin{aligned} \dot{x}(t) &= -v(t) \\ \dot{v}(t) &= \lambda(\nabla f(x(t)) - v(t)), \end{aligned} \tag{22}$$

where $\lambda = (1-\beta)/\alpha$. We now propose a suitable modification of (22), such that the resulting dynamics converges to equilibrium $(x^\star, \mathbf{0})$ in a fixed-time. We refer to this modified fixed-time stable dynamical system as *FxTS(M)-GF* which is an acronym for fixed-time stable gradient flow with momentum. The continuous-time dynamics for the FxTS(M)-GF for $(x, v) \neq (x^\star, \mathbf{0})$ is described as:

$$\begin{aligned} \dot{x} &= -v \cdot h(x, v) \\ \dot{v} &= \lambda(\nabla f(x) - v) \cdot g_{p,q}(\|\nabla f(x) - v\|), \end{aligned} \tag{23}$$

where $g_{p,q} : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ is defined as $g_{p,q}(s) := 1/(s)^{\frac{p-2}{p-1}} + 1/(s)^{\frac{q-2}{q-1}}$ with $p > 2$, $q \in (1, 2)$. The function $h(x, v)$ is defined as:

$$h(x,v) = \begin{cases} g_{p,q}(\|\nabla f(x)\|), & \text{if } \|\nabla f(x)\| > \|\nabla f(x) - v\| \\ 1, & \text{else} \end{cases}.$$

It is possible to choose the exponents p, q such that $g_{p,q}$ is monotonically decreasing on $\mathbb{R}_{>0}$. For instance, choosing $p = 2.1$ and $q = 1.98$ results in a monotonically decreasing $g_{p,q}(\cdot)$. We now state our main result.

Assumption 4. *The function* f *is* μ*-strongly convex and has an* L*-Lipschitz continuous gradient, i.e.,* $\mu \leq \|\nabla^2 f(x)\| \leq L$ *for all* $x \in \mathbb{R}^n$*.*

Theorem 5. *Under Assumptions 1 and 4, if exponents* p *and* q *are chosen such that* $g_{p,q}(\cdot)$ *is monotonically decreasing on* $\mathbb{R}_{>0}$ *and* $\lambda > L$*, then the equilibrium point* $(x^\star, \mathbf{0})$ *is fixed-time stable for the flow described in* (23)*.*

Proof. We consider the candidate Lyapunov function,

$$V(x, v) = \underbrace{\frac{1}{2}\|\nabla f(x)\|^2}_{V_1(x)} + \underbrace{\frac{1}{2}\|\nabla f(x) - v\|^2}_{V_2(x,v)}. \tag{24}$$

Clearly, $V(x, v) > 0$ for all $(x, v) \neq (x^\star, \mathbf{0})$. Additionally, define a set $\mathcal{S} = \{(x, v) \mid \|\nabla f(x)\| \leq \|\nabla f(x) - v\|\}$, and observe that

$$\begin{aligned} x \in \mathcal{S} \quad &\implies V \leq 2V_2 \\ x \notin \mathcal{S} \quad &\implies V \leq 2V_1. \end{aligned} \tag{25}$$

Taking time derivative of V along trajectories of (23) yields

$$\begin{aligned}\dot{V} &= (\nabla f)^\intercal(\nabla^2 f)\dot{x} + (\nabla f - v)^\intercal((\nabla^2 f)\dot{x} - \dot{v}) \\ &= -h\cdot(\nabla f)^\intercal(\nabla^2 f)v - h\cdot(\nabla f - v)^\intercal(\nabla^2 f)v \\ &\quad - \lambda(\nabla f - v)^\intercal(\nabla f - v)\cdot g_{p,q}(\|\nabla f - v\|) \\ &= -h\cdot(\nabla f)^\intercal(\nabla^2 f)(\nabla f) + h\cdot(\nabla f - v)^\intercal(\nabla^2 f)(\nabla f - v) \\ &\quad - \lambda(\nabla f - v)^\intercal(\nabla f - v)\cdot g_{p,q}(\|\nabla f - v\|) \\ &\leq -h\cdot\mu\|\nabla f\|^2 + \|\nabla f - v\|^2\left(h\cdot L - \lambda\cdot g_{p,q}(\|\nabla f - v\|)\right),\end{aligned}$$

where the last inequality follows from μ-strong convexity and L-Lipschitz gradient conditions.

Case 1: $x \notin \mathcal{S}$: In this case, h is given by $g_{p,q}(\|\nabla f\|)$. Moreover, $\|\nabla f - v\| \leq \|\nabla f\|$, which leads to $g_{p,q}(\|\nabla f\|) \leq g_{p,q}(\|\nabla f - v\|)$ due to the monotonicity of $g_{p,q}$. Using this and the fact that $\lambda > L$, $\dot{V}(x)$ can be upper-bounded as:

$$\begin{aligned}\dot{V} &\leq -g_{p,q}(\|\nabla f\|)\mu\|\nabla f\|^2 - g_{p,q}(\|\nabla f - v\|)(\lambda - L)\|\nabla f - v\|^2 \\ &\leq -\mu\left(2V_1\right)^{\frac{p}{2(p-1)}} - \mu\left(2V_1\right)^{\frac{q}{2(q-1)}}, \\ &\leq -\mu V^{\frac{p}{2(p-1)}} - \mu V^{\frac{q}{2(q-1)}}. \end{aligned} \tag{26}$$

Case 2: $x \in \mathcal{S}$: In this case, $h = 1$ and thus, we have

$$\begin{aligned}\dot{V} &\leq -\mu\|\nabla f\|^2 + \left(L - \lambda\cdot g_{p,q}(\|\nabla f - v\|)\right)\|\nabla f - v\|^2 \\ &\leq L\left(2V_2\right) - \lambda\left(2V_2\right)^{\frac{p}{2(p-1)}} - \lambda\left(2V_2\right)^{\frac{q}{2(q-1)}} \\ &\leq 2LV - \lambda V^{\frac{p}{2(p-1)}} - \lambda V^{\frac{q}{2(q-1)}}. \end{aligned} \tag{27}$$

Since $p > 2$ and $q \in (1, 2)$, using the similar arguments as in the proof of Theorem 2, we obtain that there exists $\gamma > 0$ such that for all $x \in \mathcal{S}$

$$\dot{V}(x) \leq -\gamma V(x)^{\frac{p}{2(p-1)}} - \gamma V(x)^{\frac{q}{2(q-1)}}. \tag{28}$$

Thus, from (26) and (28), it follows that

$$\dot{V}(x) \leq -\min\{\gamma, \mu\}\left(V(x)^{\frac{p}{2(p-1)}} + V(x)^{\frac{q}{2(q-1)}}\right), \tag{29}$$

for all $x \in \mathbb{R}^n$ i.e., the FxTS(M)-GF is fixed-time convergent gradient flow following Lemma 1. □

The proposed FxTS(M)-GF inherits some of the desirable properties of the aforementioned FxTS-GF, such as robustness and constant regret, however, a detailed analysis of the FxTS(M)-GF is left for future work.

Experiments

In this section, we present empirical results on optimizing (non-convex) functions that satisfy PL-inequality and training deep neural networks. The algorithms were implemented using PyTorch 0.4.1 on a 16GB Core-i7 2.8GHz CPU and NVIDIA GeForce GTX-1060 GPU.

Algorithms. We compare the proposed FxTS-GF and FxTS(M)-GF algorithms against the state-of-the-art Adam (Kingma and Ba 2015) and the Nesterov accelerated gradient (NAG) descent (Polyak 1964; Sutskever et al. 2013). The hyperparameters for different optimizers are tuned for optimal performance. We use constant step-size for all the algorithms.

Datasets. For the purpose of implementing deep neural networks, we examine the performances of the aforementioned algorithms on two widely used datasets: MNIST (60000 training samples, 10000 test samples) (LeCun et al. 1998), and CIFAR10 (50000 training samples, 10000 test samples) (Krizhevsky and Hinton 2009).

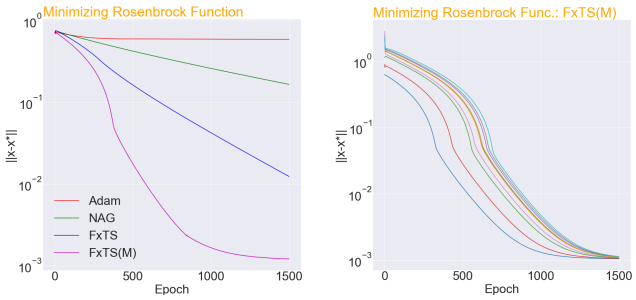

Figure 1: Minimization of Rosenbrock function. (a) Comparison of various optimization algorithms for the initial condition $(0.3, 0.8)$. (b) Performance of the FxTS(M)-GF algorithm at varying initial conditions.

Optimizing Rosenbrock Function

Rosenbrock function (Rosenbrock 1960) is a *non-convex* function with a global minimum at $(1, 1)$, and is often used to benchmark optimization algorithms. The global minimum resides in a long, narrow, parabolic-shaped flat valley. While it is easy to locate the valley, the convergence of optimization algorithms to a global minimum is difficult. The Rosenbrock function is given by:

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2.$$

Despite it being a non-convex function, the Rosenbrock function can be shown to satisfy PL-inequality (2) with modulus $\mu = 0.1$ in the region $[-1, 1] \times [-1, 1]$. We evaluate the convergence behavior of various optimization algorithms for the initial point $(x_1, x_2) = (0.3, 0.8)$ and constant learning rate 10^{-3}. We use the following hyperparameters for the optimization algorithms:

Adam: $\beta's = (0.9, 0.999)$, $\epsilon = 10^{-8}$
NAG: Momentum = 0.5
FxTS-GF: $\beta's = (1.25, 1.25)$, $\alpha's = (20, 1.98)$
FxTS(M)-GF: $\beta's = (1.25, 1.25)$, $\alpha's = (20, 1.98)$, Momentum = 0.18

Figure 1a plots the evolution of the norm of the error term $x - x^\star$ for various optimization algorithms. It can be seen that the proposed FxTS-GF and FxTS(M)-GF algorithms converge much faster than the Adam and NAG optimizers. The detailed evolution of descent trajectories of the aforementioned optimization algorithms for varying initial conditions can be found in the supplementary material. The performance of the FxTS(M)-GF algorithm for randomly chosen initial conditions is shown in Figure 1b on a semilog-scale. A straight line on a semilog-scale depicts exponentially fast convergence. However, the proposed FxTS(M)-GF is shown to achieve faster than exponential convergence to the global minimum, independent of initialization. Additional results can be found in the supplementary material.

Training Deep Neural Networks

The accelerated convergence behavior of the FxTS(M)-GF is further evaluated by training deep neural networks on MNIST and CIFAR10 datasets, respectively. As before, the performance of the FxTS(M)-GF algorithm is benchmarked

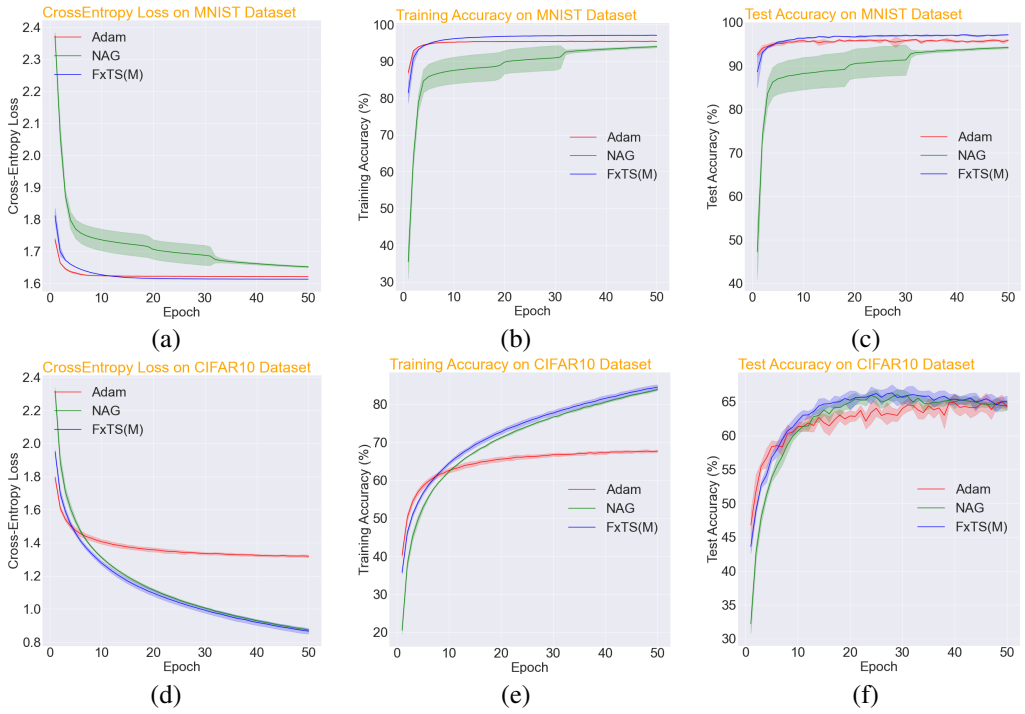

Figure 2: Comparison of several optimization algorithms for training deep neural networks on MNIST and CIFAR datasets across five random seeds. FxTS(M) outperforms Adam and NAG optimizers on various performance measures.

against the Adam and NAG optimizers for three criteria: (i) minimization of training loss, (ii) training accuracy, (iii) accuracy on the test set (generalization). The neural network architecture for MNIST consists of a convolutional layer (with 32 filters of size 3×3), followed by a dense layer of output size 128. The final layer consists transforms the 128-dimensional input into a 10-dimensional output (corresponding to ten classes). We employ the ReLU activation function for the convolutional and the first linear layer, and the softmax activation function for the output layer. The loss function is the cross-entropy along with l_2-regularization (coefficient 0.01). The learning rates for Adam and NAG are kept at 10^{-3}. A larger learning rate for Adam and NAG seems to destabilize the learning curve. On the other hand, the learning rate for the FxTS(M)-GF is chosen as 0.005. The momentum parameters for the NAG and FxTS(M)-GF algorithms are chosen as 0.5 and 0.3, respectively. The training loss vs epoch is presented in Figure 2a, while the training and testing accuracies are depicted in Figures 2b and 2c, respectively. These figures depict average performances across five random seeds. As can be seen, our FxTS(M)-GF achieves the lowest training loss on the MNIST dataset. Moreover, this performance gain also translates into better performance on training and testing accuracies.

For evaluating optimizers on the CIFAR10 dataset, we consider a neural network architecture with two convolutional layers (6 filters of size 5×5, 16 filters of size 5×5), each followed by a max-pooling layer (with a 2×2 window). The architecture also consists of three fully connected layers of output sizes 120, 84, and 10 (number of classes), respectively. We employ the ReLU activation function for the convolutional and the first two linear layers, and the softmax activation function for the output linear layer. The learning rates for all the optimizers are chosen as 10^{-3}, while the momentum parameters for the NAG and the FxTS(M)-GF are chosen as 0.5. The training loss (averaged across five random seeds) vs epoch is presented in Figure 2d, while the training and testing accuracies are depicted in Figures 2e and 2f, respectively. As can be seen, the proposed FxTS(M)-GF achieves the lowest training loss on the CIFAR10 dataset, too, along with better performance on training and testing accuracies. Interestingly, the training curve with Adam optimizer plateaus quite early during the training.

Conclusion

In this paper, we leverage continuous-time stability theory to develop novel optimization algorithms with accelerated convergence guarantees. In particular, we demonstrate that a class of continuous-time dynamical systems, suitably designed to track the minimum of convex objective functions, can do so in a fixed time independent of initialization. The resulting continuous-time dynamics are shown to be consistent upon discretization. The continuous-time dynamical system also comprises two desirable characteristics: (a) robustness to additive perturbations, (b) constant regret bounds. As an extension to data-driven learning, we also develop a momentum-based fixed-time convergent gradient flow scheme. The equivalent discretized algorithm is validated on several examples consisting of training of neural networks and minimization of invex functions. The proposed FxTS(M)-GF scheme outperforms Adam and NAG optimizers on several performance measures.

References

Benosman, M.; Romero, O.; and Cherian, A. 2020. Optimizing deep neural networks via discretization of finite-time convergent flows. *arXiv preprint arXiv:2010.02990*.

Bhat, S. P.; and Bernstein, D. S. 2000. Finite-Time Stability of Continuous Autonomous Systems. *SIAM Journal on Control and Optimization*, 38(3): 751–766.

Brown, A.; and Bartholomew-Biggs, M. 1989. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *Journal of Optimization Theory and Applications*, 62: 211–224.

Chen, F.; and Ren, W. 2018. Convex Optimization via Finite-Time Projected Gradient Flows. In *2018 IEEE Conference on Decision and Control (CDC)*, 4072–4077.

Cortés, J. 2006. Finite-time convergent gradient flows with applications to network consensus. *Automatica*, 42(11): 1993–2000.

Garg, K.; Baranwal, M.; Gupta, R.; and Benosman, M. 2021. Fixed-Time Stable Proximal Dynamical System for Solving MVIPs. ArXiv e-Print.

Garg, K.; and Panagou, D. 2021. Fixed-Time Stable Gradient Flows: Applications to Continuous-Time Optimization. *IEEE Transactions on Automatic Control*, 66(5): 2002–2015.

Huo, Z.; Gu, B.; Liu, J.; and Huang, H. 2018. Accelerated method for stochastic composition optimization with nonsmooth regularization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Karimi, H.; Nutini, J.; and Schmidt, M. 2016. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 795–811. Springer.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations San Diego*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, H.; Fang, C.; and Lin, Z. 2020. Accelerated first-order optimization algorithms for machine learning. *Proceedings of the IEEE*, 108(11): 2067–2082.

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

Polyakov, A. 2011. Nonlinear feedback design for fixed-time stabilization of linear control systems. *IEEE Transactions on Automatic Control*, 57(8): 2106–2110.

Polyakov, A.; Efimov, D.; and Brogliato, B. 2019. Consistent discretization of finite-time and fixed-time stable systems. *SIAM Journal on Control and Optimization*, 57(1): 78–103.

Rosenbrock, H. 1960. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3): 175–184.

Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Sra, S.; Nowozin, S.; and Wright, S. J. 2012. *Optimization for machine learning*. MIT Press.

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.

Sun, C.; and Hu, G. 2020. A Continuous-Time Nesterov Accelerated Gradient Method for Centralized and Distributed Online Convex Optimization. *arXiv preprint arXiv:2009.12545*.

Sutskever, I.; Martens, J.; Dahl, G.; and Hinton, G. 2013. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, 1139–1147. PMLR.

Wibisono, A.; Wilson, A. C.; and Jordan, M. I. 2016. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences of the United States of America*, 113(47): E7351—E7358.