An Inertial Newton Algorithm for Deep Learning

Camille Castera, Jérôme Bolte, Cédric Févotte, Edouard Pauwels

An Inertial Newton Algorithm for Deep Learning

Camille Castera CAMILLE.CASTERA@IRIT.FR
IRIT - CNRS
Université de Toulouse
Toulouse, France

Jérôme Bolte JEROME.BOLTE@TSE-FR.EU
Toulouse School of Economics
Université Toulouse 1 Capitole
Toulouse, France

Cédric Févotte CEDRIC.FEVOTTE@IRIT.FR
IRIT - CNRS
Université de Toulouse
Toulouse, France

Edouard Pauwels EDOUARD.PAUWELS@IRIT.FR
IRIT - CNRS
Université de Toulouse
DEEL, IRT Saint Exupery
Toulouse, France

Abstract

We introduce a new second-order inertial method for machine learning called INDIAN, exploiting the geometry of the loss function while only requiring stochastic approximations of the function values and the generalized gradients. This makes the method fully implementable and adapted to large-scale optimization problems such as the training of a deep neural network. The algorithm combines both gradient-descent and Newton-like behaviors as well as inertia. We prove the convergence of INDIAN to critical points for most deep learning problems. To do so, we provide a well-suited framework to analyze deep learning losses involving tame optimization in which we study the continuous dynamical system together with the discrete stochastic approximations. On the theoretical side, we also prove a sublinear convergence rate for the continuous time differential inclusion which underlies the algorithm. From an empirical point of view the algorithm shows promising results on popular DNN training benchmark problems.

1. Introduction

Can we devise a learning algorithm for general/nonsmooth deep neural networks (DNNs) featuring inertia and Newtonian directional intelligence only by means of a backpropagation oracle? In an optimization jargon: can we use second order ideas in time and space for *nonsmooth nonconvex* optimization by uniquely using a subgradient oracle? Before providing some answers to this question, let us have a glimpse at some of the fundamental optimization algorithms for training deep networks.

The backpropagation algorithm is, to this day, the fundamental block to compute gradients in DNNs training. It is used in instances of the Stochastic Gradient Descent algorithm (SGD, Robbins and Monro (1951)). The latter is powerful, flexible, capable of handling huge size problems, noise, and further comes with theoretical guarantees of many kinds. We refer to Bottou and Bousquet

(2008); Moulines and Bach (2011) in a convex machine learning context and Bottou et al. (2018) for a recent account highlighting the importance of deep learning (DL) applications and their challenges. In the nonconvex setting, recent works of Adil (2018); Davis et al. (2019) follow the *Ordinary Differential Equations (ODE) approach* introduced in Ljung (1977), and further developed in Benaïm (1999); Kushner and Yin (2003); Benaïm et al. (2005); Borkar (2009). SGD is however a raw first-order algorithm requiring manual tuning and whose convergence rate can sometimes be low on some DL instances. In the recent literature two improvement lines have been explored:

– Use local geometry of empirical losses to improve on steepest descent directions.

– Use past steps history to design clever steps in the present.

The first approach is akin to quasi-Newton methods while the second revolves around Polyak's inertial method (Polyak, 1964). The latter is inspired by the following appealing mechanical thought-experiment. Consider a heavy ball evolving on the graph of the loss (the loss *landscape*), subject to gravity and stabilized by some friction effects. Friction generates energy dissipation, so that the particle will eventually reach a steady state which one hopes to be a local minimum. These two approaches are already present in the DL literature: among the most popular algorithms for training DNNs, ADAGRAD (Duchi et al., 2011) features local geometrical aspects while ADAM (Kingma and Ba, 2014) combines inertial ideas with step sizes similar to the ones of ADAGRAD. Stochastic Newton and quasi-Newton algorithms have been considered by Martens (2010); Byrd et al. (2011, 2016) and recently reported to perform efficiently on several problems (Berahas et al., 2017; Xu et al., 2017). The work of Wilson et al. (2017) demonstrates that carefully tuned SGD and heavy-ball algorithms are competitive with concurrent methods.

However, deviating from the simplicity of Stochastic Gradient Descent also comes with major challenges because of the size and the severe absence of regularity in Deep Learning (differential regularity is generally absent, but even weaker regularity such as semi-convexity or Clarke regularity are not available). All sorts of practical and theoretical hardships are met: defining and even computing the Hessians is delicate, inverting them is unthinkable at this day, first and second-order Taylor approximation are unavailable due to nonsmoothness, and one has to deal with shocks which are inherent to inertial approaches in a nonsmooth context ("corners, walls" indeed generate velocity discontinuity). This makes the study of ADAGRAD and ADAM in full generality quite difficult. Some recent progresses are reported in Barakat and Bianchi (2018).

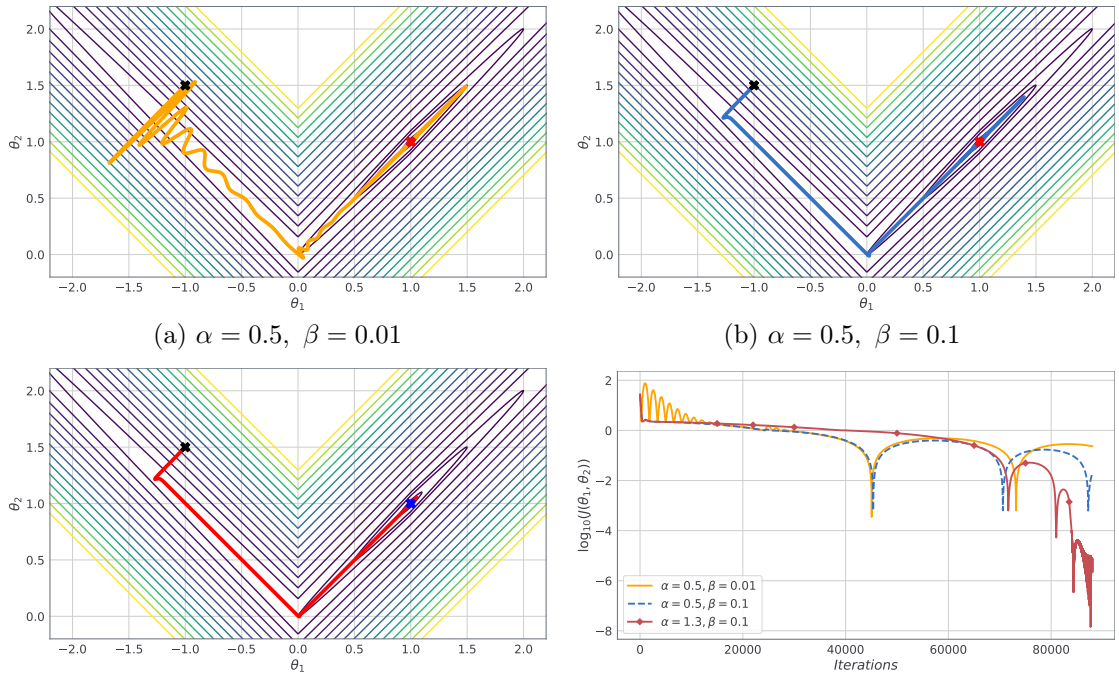
Our approach is inspired by the following dynamical system (DIN)[1] introduced in Alvarez et al. (2002):

$$\underbrace{\ddot{\theta}(t)}_{\text{Inertial term}} + \underbrace{\alpha\,\dot{\theta}(t)}_{\text{Friction term}} + \underbrace{\beta\,\nabla^2\mathcal{J}(\theta(t))\dot{\theta}(t)}_{\text{Newtonian effects}} + \underbrace{\nabla\mathcal{J}(\theta(t))}_{\text{Gravity effect}} = 0,\ t \geq 0, \tag{1}$$

where t is the time parameter which acts as a continuous epoch counter, $\mathcal{J}$ is a given loss function (usually empirical loss in DL applications) assumed C^2 just for now. Furthermore, $\nabla\mathcal{J}$ and $\nabla^2\mathcal{J}$ denote the gradient of $\mathcal{J}$ and its Hessian respectively. This system blends inertial ideas with Newton's method.

To adapt this dynamics to DL and overcome the computational difficulties generated by second-order objects occurring in (1), we work with the Clarke subdifferential,[2] combine a phase space lifting

1. For "dynamical inertial Newton" dynamics.
2. Which is obtained through convex combination of limiting derivatives.



(a) $\alpha = 0.5,\ \beta = 0.01$ (b) $\alpha = 0.5,\ \beta = 0.1$

(c) $\alpha = 1.3,\ \beta = 0.1$ (d) Objective function $\mathcal{J}$ (in log-scale)

Figure 1: Illustration of INDIAN applied to the nonsmooth function $\mathcal{J}(\theta_1, \theta_2) = 100(\theta_2 - |\theta_1|)^2 + |1 - \theta_1|$. Subplots (a-c) represent the trajectories of the parameters θ_1 and θ_2 in $\mathbb{R}^2$ for three choices of hyper-parameters α and β, see Equation (1) for an intuitive explanation. Subplot (d) displays the values of the objective function $\mathcal{J}(\theta_1, \theta_2)$ for the three settings considered.

method with of a small-step discretization mini-batch process. This overcomes the nonsmoothness while allowing the use of stochastic (mini-batches) approaches which are often required due to large data sets. An important difficulty is met when dealing with networks having nonsmooth activation functions such as ReLU (Glorot et al., 2011). Indeed subsampled versions of the algorithm, which are absolutely necessary in practice, must be treated with great care since the sum of subdifferentials no longer coincides with the subdifferential of the sum. We address this delicate issue by using new notions of steady states and by providing adequate calculus rules.

The resulting algorithm, called INDIAN, shows great efficiency in practice. For the same computational price than other tested methods (including ADAM and ADAGRAD), INDIAN avoids parasite oscillations (see Section 5.1), often achieves better training accuracy and shows robustness to hyper-parameter setting. A first illustration of the behaviour of the induced dynamics is given in Figure 1 for a simple nonsmooth and nonconvex function in $\mathbb{R}^2$.

Our theoretical results are general and simple. Using Lyapunov analysis from Alvarez et al. (2002) we combine tame nonsmooth results *à la* Sard (Sard, 1942) and the differential inclusion approximation method (Benaïm et al., 2005) to characterize the asymptotics of our algorithm

similarly to Davis et al. (2019); Adil (2018). This provides a strong theoretical ground to our study since we can prove that our method converges to a connected component of the set of steady states even in the ReLU case where the optimization problem is nonsmooth. For the smooth deterministic dynamics, we also show that convergence in values is of the form $O(1/t)$ where t is the running time. For doing so we provide a general result for curves having certain good Lyapunov functions.

The algorithm is described in details in Section 2 and its convergence proof is given in Section 3. Convergence rates of the underlying continuous time differential inclusion are derived in Section 4. Section 5 describes experimental results on synthetic and real data sets.

2. INDIAN: an Inertial Newton Algorithm for Deep Neural Networks

We first make precise the functional framework and then we describe, step by step, the process of building INDIAN from Equation (1).

2.1 Neural Networks with Lipschitz Continuous Prediction Function and Losses

We consider DNNs of a general type represented by a locally Lipschitz continuous function $f : (x, \theta) \in \mathbb{R}^M \times \mathbb{R}^P \mapsto y \in \mathbb{R}^D$, as for instance, a composition of feed-forward, convolutional, recurrent networks with ReLU, sigmoid, or tanh activation functions.

Recall that a function $F : \mathbb{R}^P \to \mathbb{R}$ is *locally Lipschitz continuous*, if for any $\theta \in \mathbb{R}^P$, there exists a neighborhood V of θ and a constant $C > 0$ such that for any $\theta_1, \theta_2 \in V$,

$$|F(\theta_1) - F(\theta_2)| \leq C \|\theta_1 - \theta_2\|,$$

where $\|\cdot\|$ is any norm on $\mathbb{R}^P$. A function $F : \mathbb{R}^P \to \mathbb{R}^D$ is locally Lipschitz continuous if each of its coordinates is locally Lipschitz continuous.

The variable $\theta \in \mathbb{R}^P$ is the parameter of the model (P can be very large), while $x \in \mathbb{R}^M$ and $y \in \mathbb{R}^D$ represent input and output data. For instance, the vector x may embody an image while y is a label explaining its content. Consider further a data set of N samples $(x_n, y_n)_{n=1,\ldots,N}$. Training the network amounts to finding a value of the parameter θ such that, for each input data x_n of the data set, the output $f(x_n, \theta)$ of the model predicts the real value y_n with good accuracy.

To do so, we follow the traditional approach of minimizing an empirical risk loss function

$$\mathbb{R}^P \ni \theta \mapsto \mathcal{J}(\theta) = \sum_{n=1}^{N} l(f(x_n, \theta), y_n), \tag{2}$$

where $l : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a locally Lipschitz continuous dissimilarity measure.

2.2 Neural Networks and Tameness in a Nutshell

Tameness refers to an ubiquitous geometrical property of losses and constraints encompassing most finite dimensional optimization problems met in practice. Prominent classes of tame objects are piecewise linear or piecewise polynomial objects (with finitely many pieces), semi-algebraic objects. But the notion is much more general as we intent to convey below; the formal definition is recalled at the end of this subsection (Definition 1).

Informally, sets or functions are called tame when they can be described by a finite number of basic formulas, inequalities, or Boolean operations involving standard functions such as polynomial,

exponential, or max functions. We refer to Attouch et al. (2010) for illustrations, recipes and examples within a general optimization setting or Davis et al. (2019) for illustrations in the context of neural networks. One is referred to Van den Dries (1998); Coste (2000); Shiota (2012) for foundational material. To apprehend the strength behind tameness it is convenient to remember that it models nonsmoothness by confining the study to sets and functions which are union of smooth pieces. This is the so-called *stratification* property of tame sets and functions. It was this property which motivated the vocable of *tame topology*,[3] see Van den Dries (1998). In a nonconvex optimization settings, the stratification property is crucial to generalize qualitative algorithmic results to nonsmooth objects.

All finite dimensional deep learning optimization models we are aware of yield tame losses $\mathcal{J}$. To understand this assertion and convey the wide scope of tameness assumptions, let us provide concrete examples (see also Davis et al. (2019)). If one assumes that the neural networks under consideration are built from the following traditional components:

- The network architecture describing f is fixed with an arbitrary, number of layers of arbitrary dimensions and arbitrary Directed Acyclic Graph (DAG) representing computation.
- The activation functions are among classical ones: ReLU, sigmoid, SQNL, RReLU, tanh, APL, soft plus, soft clipping, and many others including multivariate activations (norm, sorting), or activations defined piecewise with polynomials, exponential and logarithm.
- The dissimilarity function $l(x, y)$ is a standard loss such as ℓ_p norms, logistic loss or cross-entropy, more generally defined piecewise using polynomials, exponential and logarithm.

Then one easily shows, by elementary quantifier elimination arguments (property (iii) below), that the corresponding loss, $\mathcal{J}$, is tame.

For the sake of completeness, we provide below the formal definition of tameness and o-minimality.

Definition 1 [o-minimal structure] (Coste, 2000, Definition 1.5) An *o-minimal* structure on $(\mathbb{R}, +, .)$ is a countable collection of sets $\mathcal{O} = \{\mathcal{O}_q\}_{q\geq 1}$ where each $\mathcal{O}_q$ is itself a collection of subsets of $\mathbb{R}^q$, called *definable* subsets. They must have the following properties, for each $q \geq 1$:

(i) (Boolean properties) $\mathcal{O}_q$ contains the empty set, is stable by finite union, finite intersection and complementation;

(ii) (Lifting property) if A belongs to $\mathcal{O}_q$, then $A \times \mathbb{R}$ and $\mathbb{R} \times A$ belong to $\mathcal{O}_{q+1}$.

(iii) (Projection or quantifier elimination property) if $\Pi : \mathbb{R}^{q+1} \to \mathbb{R}^q$ is the canonical projection onto $\mathbb{R}^q$ then for any A in $\mathcal{O}_{q+1}$, the set $\Pi(A)$ belongs to $\mathcal{O}_q$.

(iv) (Semi-algebraicity) $\mathcal{O}_q$ contains the family of algebraic subsets of $\mathbb{R}^q$, that is, every set of the form

$$\{\theta \in \mathbb{R}^q : \zeta(\theta) = 0\},$$

where $\zeta : \mathbb{R}^q \to \mathbb{R}$ is a polynomial function.

(v) (Minimality property), the elements of $\mathcal{O}_1$ are exactly the finite unions of intervals and points.

3. "*La topologie modérée*" wished for by Grothendieck.

A mapping $F : S \subset \mathbb{R}^m \to \mathbb{R}^q$ is said to be *definable in* $\mathcal{O}$ if its graph is definable in $\mathcal{O}$ as a subset of $\mathbb{R}^m \times \mathbb{R}^q$.

For illustration of o-minimality in the context of optimization one is referred to Attouch et al. (2010); Davis et al. (2019).

From now on we fix an o-minimal structure $\mathcal{O}$, a set or a mapping definable in $\mathcal{O}$ will be called *tame*.

2.3 From the Second-order ODE (DIN) to the Algorithm INDIAN with Mini-batches

We describe the construction of an algorithm adapted from (1).

2.3.1 Handling Nonsmoothness and Nonconvexity

In a first time we show how the formalism offered by Clarke's subdifferential is applicable in order to generalize (1), also known as (DIN), to the nonsmooth nonconvex setting.

Recall that this dynamical system (1) writes

$$\ddot{\theta}(t) + \alpha\dot{\theta}(t) + \beta\nabla^2\mathcal{J}(\theta(t))\dot{\theta}(t) + \nabla\mathcal{J}(\theta(t)) = 0, \tag{3}$$

where $\mathcal{J}$ is a twice differentiable potential, and $\alpha > 0$, $\beta > 0$ are two hyper-parameters. We cannot exploit Equation (3) directly since in most DL applications $\mathcal{J}$ is not twice differentiable, not even once. We first overcome the explicit use of the Hessian matrix $\nabla^2\mathcal{J}$ by introducing an auxiliary variable like in Alvarez et al. (2002). Let $\theta : \mathbb{R}_+ \to \mathbb{R}^P$ be a solution of (3). We define the auxiliary variable as $\psi = -\beta\dot{\theta} + \beta^2\nabla\mathcal{J}$. For a $\mathcal{J}$ merely differentiable, we introduce the following dynamical system:

$$\begin{cases} \dot{\theta}(t) + \beta\nabla\mathcal{J}(\theta(t)) & +(\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) = 0 \\ \dot{\psi}(t) & +(\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) = 0 \end{cases}, \quad \text{for a.e. } t \in (0, +\infty). \tag{4}$$

As explained in Alvarez et al. (2002), when $\mathcal{J}$ is twice differentiable, (3) is equivalent to (4) but the latter does not require the existence of second order derivatives.

Let us now introduce a new *nonconvex nondifferentiable* version of (4). By Rademacher's theorem, locally Lipschitz continuous deep learning losses $\mathcal{J} : \mathbb{R}^P \to \mathbb{R}$ are differentiable almost everywhere. Denote by R the set of points where $\mathcal{J}$ is differentiable. Then, $\mathbb{R}^P \setminus \mathsf{R}$ has zero Lebesgue measure so for any $\theta^\star \in \mathbb{R}^P \setminus \mathsf{R}$, we have a sequence of points of R whose limit is this $\theta^\star$. This motivates the introduction of the following definition due to Clarke (1990).

Definition 2 (Clarke subdifferential of Lipschitz functions) *For any locally Lipschitz continuous function $F : \mathbb{R}^P \to \mathbb{R}$, the Clarke subgradient of F at $\theta \in \mathbb{R}^P$, denoted $\partial F(\theta)$, is the set defined by,*

$$\partial F(\theta) = \operatorname{conv}\left\{ v \in \mathbb{R}^P, \exists(\theta_k)_{k\in\mathbb{N}} \in \mathbb{R}^{\mathbb{N}}, \text{ such that } \theta_k \xrightarrow[k\to\infty]{} \theta \text{ and } \nabla F(\theta_k) \xrightarrow[k\to\infty]{} v \right\}, \tag{5}$$

where conv *denotes the convex hull operator.*

The Clarke subdifferential is a nonempty compact convex set. Thanks to this definition, we can extend (4) to nondifferentiable functions. As $\partial\mathcal{J}(\theta)$ is a set, we do no longer study a differential

equation but rather a *differential inclusion*,

$$\begin{cases} \dot{\theta}(t) + \beta \partial \mathcal{J}(\theta(t)) & +(\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \ni 0 \\ \dot{\psi}(t) & +(\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \ni 0 \end{cases}, \quad \text{for a.e. } t \in (0, +\infty). \tag{6}$$

For a given initial condition $(\theta_0, \psi_0) \in \mathbb{R}^P \times \mathbb{R}^P$, we call a solution (or trajectory) of this system any absolutely continuous curve (θ, ψ) from $\mathbb{R}_+$ to $\mathbb{R}^P \times \mathbb{R}^P$ for which $(\theta(0), \psi(0)) = (\theta_0, \psi_0)$ and (6) holds. We recall that absolute continuity amounts to the fact that θ is differentiable almost everywhere with integrable derivative and

$$\theta(t) - \theta(0) = \int_0^t \dot{\theta}(s) ds, \text{ for all } t \geq 0.$$

Due to the properties of the Clarke subdifferential existence of a solution to differential inclusions such as (6) is ensured, see Aubin and Cellina (2012); note however that uniqueness does not hold in general. We will now use the form of this differential inclusion to build a new algorithm to train DNNs.

2.3.2 Discretization of the Differential Inclusion

To obtain the basic form of our algorithm, we discretize (6) according to the classical explicit Euler method. Given (θ, ψ) a solution of (6) and any time t_k, set $\theta_k = \theta(t_k)$ and $\psi_k = \psi(t_k)$. Then, at time $t_{k+1} = t_k + \gamma_k$ with γ_k positive small, one can approximate $\dot{\theta}(t_{k+1})$ and $\dot{\psi}(t_{k+1})$ by

$$\dot{\theta}(t_{k+1}) \simeq \frac{\theta_{k+1} - \theta_k}{\gamma_k}, \qquad \dot{\psi}(t_{k+1}) \simeq \frac{\psi_{k+1} - \psi_k}{\gamma_k}.$$

This discretization yields the following algorithm:

$$\begin{cases} v_k & \in \partial \mathcal{J}(\theta_k) \\ \theta_{k+1} & = \theta_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k - \beta v_k \right) \\ \psi_{k+1} & = \psi_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k \right) \end{cases} \tag{7}$$

Although the algorithm above is well defined for our problem, it is almost useless for deep learning purposes. Indeed, due to the absence of an operational sum rule for the Clarke's subdifferential, it is not possible to consider mini-batch versions of (7). The next section is meant to overcome this problem and to state formally our central algorithm. The reader not familiar with nonsmooth analysis may skip this next part at first and think INDIAN as being the Algorithm 7 under study.

2.3.3 INDIAN Algorithm with a New Notion of Steady States

In order to compute or approximate the subdifferential of $\mathcal{J}$ at each iteration and to cope with large data sets, $\mathcal{J}$ can be approximated by mini-batches, reducing the memory footprint and computational cost of evaluation. For any $\mathsf{B} \subset \{1, \ldots, N\}$, define

$$\mathcal{J}_{\mathsf{B}} \colon \theta \mapsto \sum_{n \in \mathsf{B}} l(f(x_n, \theta), y_n). \tag{8}$$

Unlike in the differentiable case, subgradients do not in general sum up to a subgradient of the sum, that is $\partial \mathcal{J}_{\mathsf{B}}(\theta) \neq \sum_{n\in\mathsf{B}} \partial l(f(x_n,\theta),y_n)$ in general. To see this, take for example $0 = |\cdot| - |\cdot|$, the Clarke subgradient of this function at 0 is $\{0\}$, whereas $\partial(|0|) + \partial(-|0|) = [-1,1] + [-1,1] = [-2,2]$. In order to circumvent this failure, we introduce a notion of steady states that corresponds to the stationary points generated by a generic mini-batch approach. As we shall see, this allows both for practical applications and convergence analysis (despite the sum rule failure for Clarke subdifferential).

For any $\mathsf{B} \subset \{1,\dots,N\}$, we introduce the following objects:

$$D\mathcal{J}_{\mathsf{B}} = \sum_{n\in\mathsf{B}} \partial \left[l(f(x_n,\cdot),y_n) \right], \quad D\mathcal{J} = \sum_{n=1}^{N} \partial \left[l(f(x_n,\cdot),y_n) \right]. \tag{9}$$

Observe that, for each B, we have $D\mathcal{J}_{\mathsf{B}} \supset \partial \mathcal{J}_{\mathsf{B}}$ and that $\mathcal{J}_{\mathsf{B}}$ is differentiable almost everywhere with $D\mathcal{J}_{\mathsf{B}} = \partial \mathcal{J}_{\mathsf{B}} = \{\nabla \mathcal{J}_{\mathsf{B}}\}$, see Clarke (1990). When $\mathcal{J}$ is tame the equalities even hold on the complement of a finite union of manifolds of dimension strictly lower than P—use the classical stratification results for o-minimal structures, Coste (2000). For convenience, a point satisfying $D\mathcal{J}(\theta) \ni 0$ will be called *D-critical*. This vocable is motivated by favourable properties: sum and chain rules along curves (see Lemmas 5 and 6 below) and the existence of a tame Sard's theorem (see Lemma 7). To our knowledge, this notion of a steady state has not previously been used in the literature. While this notion is needed for the theoretical analysis, one should keep in mind that $D\mathcal{J}$ is actually what is computed numerically provided that the basic oracle returns a Clarke subgradient. This computation is usually done with a backpropagation algorithm, similarly to the seminal method of Rumelhart and Hinton (1986).

Ultimately, one can rewrite (6) replacing $\partial \mathcal{J}$ by $D\mathcal{J}$, which yields a differential inclusion adapted to study mini-batch approximations of nonsmooth losses $\mathcal{J}$, it reads

$$\begin{cases} \dot\theta(t) + \beta D\mathcal{J}(\theta(t)) & +(\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \ni 0 \\ \dot\psi(t) & +(\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \ni 0 \end{cases}, \quad \text{for a.e. } t \in (0,+\infty). \tag{10}$$

Remark that this is simply (6) with $D\mathcal{J}$. Discretizing this system gives an implementable version of INDIAN. We consider a sequence $(\mathsf{B}_k)_{k\in\mathbb{N}}$ of nonempty subsets of $\{1,\dots,N\}$ chosen independently, uniformly at random with replacement and a sequence of positive step sizes $(\gamma_k)_{k\in\mathbb{N}}$. For a given initialization $(\theta_0,\psi_0) \in \mathbb{R}^P \times \mathbb{R}^P$, at iteration $k \geq 1$, it reads:

$$\text{(INDIAN)} \quad \begin{cases} v_k & \in D\mathcal{J}_{\mathsf{B}_k}(\theta_k) \\ \theta_{k+1} & = \theta_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k - \beta v_k \right) \\ \psi_{k+1} & = \psi_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k \right) \end{cases} \tag{11}$$

Here again $\alpha > 0$ and $\beta > 0$ are the hyper-parameters of the algorithm. The whole process is a stochastic approximation of the deterministic dynamics obtained by choosing $\mathsf{B}_k \equiv \{1,\dots,N\}$, that is $\mathcal{J}_{\mathsf{B}_k} \equiv \mathcal{J}$ (batch version). This can be seen by observing that the vectors v_k above may be written $v_k = \tilde v_k + \eta_k$, where $\tilde v_k \in D\mathcal{J}(\theta_k)$ and η_k compensates for the missing subgradients and can be seen as a zero-mean noise.

Hence, INDIAN admits the following general abstract stochastic formulation:

$$\begin{cases} w_k & \in D\mathcal{J}(\theta_k) \\ \theta_{k+1} & = \theta_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k - \beta w_k + \xi_k \right) \\ \psi_{k+1} & = \psi_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k \right) \end{cases} \tag{12}$$

where $(\xi_k)_{k\in\mathbb{N}}$ is a martingale difference noise sequence adapted to the filtration induced by (random) iterates up to k, and θ_0, ϕ_0 are arbitrary initial conditions. While (11) is the version implemented in practice, its equivalent form (12) is more convenient for the convergence analysis of the next section. We stress that the equivalence between (11) and (12) relies on the use of $D\mathcal{J}$ and would not hold with $\partial\mathcal{J}$ in Algorithm 7.

Let us gather the previous derivations and assumptions:

Inertial Newton Algorithm for Deep Learning (INDIAN)

Objective function: $\mathcal{J} = \sum_{n=1}^{N} \mathcal{J}_n, \quad \mathcal{J}_n \colon \mathbb{R}^P \mapsto \mathbb{R}$ locally Lipschitz, $n = 1, \dots N$.

Algorithm parameters: (α, β) positive.

Mini batches: $(\mathsf{B}_k)_{k\in\mathbb{N}}$, nonempty subsets of $\{1, \dots, N\}$.

Step sizes: $(\gamma_k)_{k\in\mathbb{N}}$ positive.

Initialization: $(\theta_0, \psi_0) \in \mathbb{R}^P \times \mathbb{R}^P$.

For $k \in \mathbb{N}$**:**

$$\begin{cases} v_k & \in & \sum_{n\in\mathsf{B}_k} \partial\left[\mathcal{J}_n(\theta_k)\right] \\ \theta_{k+1} & = & \theta_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k - \beta v_k \right) \\ \psi_{k+1} & = & \psi_k + \gamma_k \left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k \right) \end{cases}$$

3. Convergence Results for INDIAN

3.1 Main result: Accumulation Points of INDIAN are Critical

We now provide convergence guarantees for the algorithm INDIAN in a DL settings. The main idea here is to prove that the discrete algorithm (11) asymptotically behaves like the solutions of the continuous differential inclusion (10) Out of tameness, our main assumption is the following:

Assumption 1 (Stochastic approximation) *The sets $(\mathsf{B}_k)_{k\in\mathbb{N}}$ are taken independently uniformly at random with replacement. The step size sequence γ_k is positive, $\sum \gamma_k = +\infty$ and satisfies $\gamma_k = o\left(\frac{1}{\log k}\right)$, that is $\limsup_{k\to+\infty} |\gamma_k \log k| = 0$.*

Typical admissible choices are $\gamma_k = C(k+1)^{-a}$ with $a \in (0,1]$, $C > 0$. The main theoretical result of this paper follows.

Theorem 3 (INDIAN converges to the set of D-critical points of $\mathcal{J}$) *Assume that $\mathcal{J}$ is locally Lipschitz continuous, tame and that the step sizes satisfy Assumption 1. Fix an initial condition (θ_0, ψ_0) and assume that there exists $M > 0$ such that $\sup_{k\geq 0} \|(\theta_k, \psi_k)\| \leq M$ almost surely.*
Then, almost surely, any accumulation point $\bar{\theta}$ of the sequence $(\theta_k)_{k\in\mathbb{N}}$ satisfies $D\mathcal{J}(\bar{\theta}) \ni 0$. In addition $(\mathcal{J}(\theta_k))_{k\in\mathbb{N}}$ converges.

Remark 4 (a) [Step sizes]: Assumption 1 offers much more flexibility than the usual $\mathcal{O}(1/\sqrt{k})$ assumption commonly used for SGD. We leverage the boundedness assumption, local Lipschitz continuity and finite sum structure of $\mathcal{J}$, so that the noise is actually uniformly bounded, hence sub-Gaussian, allowing for much larger step sizes than in the more common bounded second moment setting. See (Benaïm et al., 2005, Remark 1.5) and Benaïm (1999) for more details. The interest of this slow decrease is highlighted in Figure 4.
(b) [Convergence of INDIAN]: Apart from the uniform boundedness of the noise, we do not use the specific structure of DL losses. Thus our result actually holds for general locally Lipschitz continuous tame functions with finite sum structure and for the general stochastic process under uniformly bounded martingale increment noise. Other variants could be considered depending on the assumptions on the noise, see Benaïm et al. (2005).
(c) [Convergence to critical points]: Observe that when $\mathcal{J}$ is differentiable, limit points are simply critical points.
(d) [Local minima]: Let us mention that for general $\mathcal{J}$, being D-critical is a necessary condition for being a local minima.

3.2 Preliminary Variational Results

We extend some results known for the Clarke subdifferential of tame functions to the operator D that we previously introduced. First, we recall a useful result of Davis et al. (2019) which follows from the projection formula in Bolte et al. (2007).

Lemma 5 (Chain rule) *Let $\mathcal{J} : \mathbb{R}^P \to \mathbb{R}$ be a locally Lipschitz continuous, tame function, then $\mathcal{J}$ admits a chain rule, meaning that for all absolutely continuous curves $\theta : \mathbb{R}_+ \to \mathbb{R}^P$, and for almost all $t \in \mathbb{R}_+$,*

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}t}(\theta(t)) = \langle \dot{\theta}(t), \partial\mathcal{J}(\theta(t))\rangle = \langle \dot{\theta}(t), v\rangle, \quad \forall v \in \partial\mathcal{J}(\theta(t)). \tag{13}$$

Consider now a function with an additive composite structure (such as in deep learning):

$$\mathcal{J} \colon \mathbb{R}^P \ni \theta \mapsto \sum_{n=1}^{N} \mathcal{J}_n(\theta), \tag{14}$$

where each $\mathcal{J}_n \colon \mathbb{R}^P \mapsto \mathbb{R}$ is locally Lipschitz and tame. We set for any $\theta \in \mathbb{R}^P$

$$D\mathcal{J}(\theta) = \sum_{n=1}^{N} \partial\mathcal{J}_n(\theta).$$

The following lemma is a direct generalization of the above chain rule.

Lemma 6 (Chain rule) *Let $\mathcal{J}$ be a sum of tame functions like in Equation (14). Let $c\colon [0,1] \mapsto \mathbb{R}^P$ be an absolutely continuous curve so that $t \mapsto \mathcal{J}(c(t))$ is differentiable almost everywhere. For almost all $t \in [0,1]$, and for all $v \in D\mathcal{J}(c(t))$,*

$$\frac{d}{dt}\mathcal{J}(c(t)) = \langle v, \dot{c}(t)\rangle\,.$$

Proof By local Lipschitz continuity and absolute continuity, each $\mathcal{J}_n$ is differentiable almost everywhere and Lemma 5 can be applied:

$$\frac{d}{dt}\mathcal{J}_n(c(t)) = \langle v_n, \dot{c}(t)\rangle\,, \text{for all } v_n \in \partial\mathcal{J}_n(c(t)) \text{ and for almost all } t \in \mathbb{R}_+.$$

Thus

$$\frac{d}{dt}\mathcal{J}(c(t)) = \sum_{n=1}^{N}\frac{d}{dt}\mathcal{J}_n(c(t)) = \sum_{n=1}^{N}\langle v_n, \dot{c}(t)\rangle\,,$$

for any $v_n \in \partial\mathcal{J}_n(c(t))$, for all $n = 1,\ldots,N$, and for almost all $t \in \mathbb{R}_+$. This proves the desired result. ■

We finish this section with a Sard Lemma for D-critical values, in the spirit of Bolte et al. (2007).

Lemma 7 (A Sard's theorem for tame D-critical values) *Set*

$$\mathsf{S} = D - \text{crit } := \left\{\theta \in \mathbb{R}^P : D\mathcal{J}(\theta) \ni 0\right\},$$

then $\mathcal{J}(\mathsf{S})$ is finite.

Proof The set S is tame and hence it has a finite number of connected components. It is sufficient to prove that $\mathcal{J}$ is constant on each connected component of S. Without loss of generality, assume that S is connected and consider $\theta_0, \theta_1 \in \mathsf{S}$. By Whitney regularity (Van den Dries, 1998, 4.15), there exist a tame continuous path Γ joining θ_0 to θ_1. Because of the tame nature of the result, we should here conclude with only tame arguments and use the projection formula in Bolte et al. (2007), but for convenience of readers who are not familiar with this result we use Lemma 5. Since Γ is tame, the monotonicity lemma (see for example (Kurdyka, 1998, Lemma 2)) gives the existence of a finite collection of real numbers $0 = a_0 < a_1 < \ldots < a_q = 1$, such that Γ is C^1 on each segment (a_{j-1}, a_j), $j = 1,\ldots,q$. Applying Lemma 5 to each $\Gamma_{|(a_i,a_{i+1})}$, we see that $\mathcal{J}$ is constant save perhaps on a finite number of points, it is thus constant by continuity. ■

3.3 Proof of Convergence for INDIAN

Our approach follows the stochastic method for differential inclusion developed in Benaïm et al. (2005) and thus the differential system (10) and its Lyapunov properties play fundamental roles.

The steady states of (10) are given by

$$\mathsf{S} = \left\{(\theta,\psi) \in \mathbb{R}^P \times \mathbb{R}^P : 0 \in D\mathcal{J}(\theta), \psi = (1-\alpha\beta)\theta)\right\}, \tag{15}$$

they are initialization values for which the system does not evolve and remains constant. Observe that the first coordinates of these points are D-critical for $\mathcal{J}$ and that conversely any D-critical point of $\mathcal{J}$ corresponds to a unique rest point in S.

Definition 8 (Lyapunov function) *Let* A *be a subset of* $\mathbb{R}^P \times \mathbb{R}^P$*, we say that* $E : \mathbb{R}^P \times \mathbb{R}^P \to \mathbb{R}$ *is a Lyapunov function for the set* A *and the dynamics* (10) *if*

(i) for any solution (θ, ψ) *of* (DIN) *with initial condition* (θ_0, ψ_0)*, we have:* $E(\theta(t), \psi(t)) \leq E(\theta_0, \psi_0)$ *a.e. on* $\mathbb{R}$*.*

(ii) for any solution (θ, ψ) *of* (DIN) *with initial condition* $(\theta_0, \psi_0) \notin \mathsf{A}$*, we have:* $E(\theta(t), \psi(t)) < E(\theta_0, \psi_0)$ *a.e. on* $\mathbb{R}$*.*

In practice, to establish that a functional is Lyapunov, one can simply use differentiation through chain rule results, with in particular Lemma 5 in Davis et al. (2019)[Theorem 5.8] based on the projection formula in Bolte et al. (2007). In the context of INDIAN , we will use Lemma 6.

To build a Lyapunov function for the dynamics (10) and the set S, consider the two following energy-like functions:

$$\begin{cases} E_{\min}(\theta(t), \psi(t)) &= (1 - \sqrt{\alpha\beta})^2 \mathcal{J}(\theta(t)) + \frac{1}{2} \left\| (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \right\|^2 \\ E_{\max}(\theta(t), \psi(t)) &= (1 + \sqrt{\alpha\beta})^2 \mathcal{J}(\theta(t)) + \frac{1}{2} \left\| (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \right\|^2 . \end{cases} \tag{16}$$

Then the following lemma applies.

Lemma 9 (Differentiation along DIN trajectories) *Let* (θ, ψ) *be a solution of* (10) *with initial condition* (θ_0, ψ_0)*. For almost all* $t > 0$*,* θ *and* ψ *are differentiable at* t*,* (10) *holds,* $\frac{\dot\theta(t) - \dot\psi(t)}{\beta} \in D\mathcal{J}(\theta(t))$ *and*

$$\frac{\mathrm{d}E_{\min}}{\mathrm{d}t}(\theta(t), \psi(t)) = -\left\| \sqrt{\alpha}\dot\theta(t) - \frac{1}{\sqrt{\beta}} \left(\dot\psi(t) - \dot\theta(t) \right) \right\|^2$$
$$\frac{\mathrm{d}E_{\max}}{\mathrm{d}t}(\theta(t), \psi(t)) = -\left\| \sqrt{\alpha}\dot\theta(t) + \frac{1}{\sqrt{\beta}} \left(\dot\psi(t) - \dot\theta(t) \right) \right\|^2$$

Proof Define $E_\lambda(\theta, \psi) = \lambda \mathcal{J}(\theta) + \frac{1}{2} \left\| (\alpha - \frac{1}{\beta})\theta + \frac{1}{\beta}\psi \right\|^2$. We aim at choosing λ so that E_λ is a Lyapunov function. Because $\mathcal{J}$ is tame and locally Lipschitz continuous, using Lemma 6 we know that for any absolutely continuous trajectory $\theta : \mathbb{R}_+ \to \mathbb{R}^P$ and for almost all $t > 0$,

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}t}(\theta(t)) = \langle \dot\theta(t), D\mathcal{J}(\theta(t)) \rangle = \langle \dot\theta(t), v(t) \rangle, \quad \forall v(t) \in D\mathcal{J}(\theta(t)). \tag{17}$$

Let θ and ψ be solutions of (DIN). For almost all $t \in \mathbb{R}_+$, we can differentiate $E_\lambda(\theta, \psi)$ to obtain

$$\begin{aligned} \frac{\mathrm{d}E_\lambda}{\mathrm{d}t}(\theta(t), \psi(t)) =& \lambda \langle \dot\theta(t), v(t) \rangle + (\alpha - \frac{1}{\beta}) \langle \dot\theta(t), (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \rangle \\ &+ \frac{1}{\beta} \langle \dot\psi(t), (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) \rangle \end{aligned} \tag{18}$$

for all $v(t) \in D\mathcal{J}(\theta(t))$. Using (10), we get $\frac{1}{\beta}(\dot\theta(t) - \dot\psi(t)) \in D\mathcal{J}(\theta(t))$ and $-\dot\psi(t) = (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t)$ a.e. Choosing $v(t) = \frac{1}{\beta}(\dot\theta(t) - \dot\psi(t))$ yields:

$$\frac{\mathrm{d}E_\lambda}{\mathrm{d}t}(\theta(t), \psi(t)) = \lambda \left\langle \dot\theta(t), \frac{\dot\theta(t) - \dot\psi(t)}{\beta} \right\rangle - (\alpha - \frac{1}{\beta}) \left\langle \dot\theta(t), \dot\psi(t) \right\rangle - \frac{1}{\beta} \left\langle \dot\psi(t), \dot\psi(t) \right\rangle .$$

Then, expressing everything as a function of $\dot\theta$ and $\frac{1}{\beta}(\psi-\theta)$, one can show that a.e. on $\mathbb{R}_+$:

$$\begin{aligned}\frac{\mathrm{d}E_\lambda}{\mathrm{d}t}(\theta,\psi)(t) &= -\alpha\|\dot\theta(t)\|^2 - \beta\left\|\frac{\dot\theta(t)-\dot\psi(t)}{\beta}\right\|^2 + (\lambda-\alpha\beta-1)\,\langle\dot\theta(t),\frac{\dot\theta(t)-\dot\psi(t)}{\beta}\rangle \\ &= -\left\|\sqrt{\alpha}\dot\theta(t)+\frac{\alpha\beta+1-\lambda}{2\sqrt{\alpha}}\frac{\dot\theta(t)-\dot\psi(t)}{\beta}\right\|^2 - \left(\beta-\frac{(\alpha\beta+1-\lambda)^2}{4\alpha}\right)\left\|\frac{\dot\theta(t)-\dot\psi(t)}{\beta}\right\|^2.\end{aligned}$$

We aim at choosing λ so that E_λ is decreasing that is $\left(\beta-\frac{(\alpha\beta+1-\lambda)^2}{4\alpha}\right)>0$. This holds whenever $\lambda\in\left[(1-\sqrt{\alpha\beta})^2,(1+\sqrt{\alpha\beta})^2\right]$. We choose $\lambda_{\min}=(1-\sqrt{\alpha\beta})^2$, and $\lambda_{\max}=(1+\sqrt{\alpha\beta})^2$, for these two values we obtain for almost all $t>0$:

$$\begin{cases}\dot E_{\lambda_{\min}}(\theta(t),\psi(t)) &= -\left\|\sqrt{\alpha}\dot\theta(t)+\frac{1}{\sqrt{\beta}}\left(\dot\theta(t)-\dot\psi(t)\right)\right\|^2 \\ \dot E_{\lambda_{\max}}(\theta(t),\psi(t)) &= -\left\|\sqrt{\alpha}\dot\theta(t)-\frac{1}{\sqrt{\beta}}\left(\dot\theta(t)-\dot\psi(t)\right)\right\|^2\end{cases} \tag{19}$$

Remark finally that by definition $E_{\min}=E_{\lambda_{\min}}$ and $E_{\max}=E_{\lambda_{\max}}$. ■

Define $E=E_{\min}+E_{\max}$ and recall that $\mathsf{S}=\left\{(\theta,\psi)\in\mathbb{R}^P\times\mathbb{R}^P : 0\in DJ(\theta),\psi=(1-\alpha\beta)\theta)\right\}$. By a direct integration argument, we obtain the following.

Lemma 10 (E is Lyapunov function for (INDIAN) with respect to S) *For any $(\theta_0,\psi_0)\notin\mathsf{S}$ and any solution (θ,ψ) with initial condition (θ_0,ψ_0),*

$$E(\theta(t),\psi(t))<E(\theta_0,\psi_0),\ \textit{for almost all } t>0. \tag{20}$$

We are now in position to provide the desired proof.

Proof of Theorem 3 Lemmas 9 and 10 entail that E is a Lyapunov function for the set S and the dynamics (10). Set $\mathsf{C}=\{\theta\in\mathbb{R}^P:(\theta,\psi)\in\mathsf{S}\}$ which is actually the set of D-critical points of $\mathcal{J}$. Using Lemma 7 of Section 3.2, $\mathcal{J}(\mathsf{C})$ is finite. Moreover, since $E(\theta,\psi)=2(1+\alpha\beta)\mathcal{J}(\theta)$ for all $(\theta,\psi)\in\mathsf{S}$, E takes a finite number of values on S, and in particular, $E(\mathsf{S})$ has empty interior.

Denote by L the set of accumulation points of a realizations of the sequences $((\theta_k,\psi_k))_{k\in\mathbb{N}}$ produced by (11) starting at (θ_0,ψ_0) and L_1 its projection on $\mathbb{R}^P\times\{0\}$. We have the 3 following properties:

– By assumption, we have $\|(\theta_k,\psi_k)\|\le M$ almost surely, for all $k\in\mathbb{N}$.
– By local Lipschitz continuity $\partial\mathcal{J}_{\mathsf{B}}(\theta)$ is uniformly bounded for $\|\theta\|\le M$ and any $\mathsf{B}\subset\{1,\dots,N\}$, hence the centered noise $(\xi_k)_{k\in\mathbb{N}}$ is a uniformly bounded martingale difference sequence.
– By Assumption 1, the sequence $(\gamma_k)_{k\in\mathbb{N}}$ are chosen such that $\gamma_k=o(\frac{1}{\log k})$ (see Remark 4) (a)).

Combining Theorem 3.6, Remark 1.5 and Proposition 3.27 of Benaïm et al. (2005) to obtain that $\mathsf{L}\subset\mathsf{S}$ and $E(\mathsf{L})$ is a singleton. Hence $\mathcal{J}(\mathsf{L}_1)$ is also a singleton and the theorem follows. □

4. Towards Convergence Rates for INDIAN

In the previous section connecting INDIAN to (10) was one of the keys to prove the convergence of the discrete dynamics. Let us now focus on the continuous dynamical system (6) in the deterministic

case where $\mathcal{J}$ and $\partial\mathcal{J}$ are not approximated anymore – we thus no longer use $D\mathcal{J}$ although this would be possible to the prize of a more technical proof. In this section and in this section only, we pertain to losses $\mathcal{J}$ that are real semi-algebraic (a particular case of tame functions).[4] Recall that a set is called semi-algebraic if it is a finite union of sets of the form

$$\{\theta \in \mathbb{R}^P, \zeta(\theta) = 0, \zeta_i(\theta) < 0\}$$

where ζ, ζ_i are real polynomial functions. A function is called semi-algebraic if its graph is semi-algebraic.

We will prove that the continuous time system (6) is actually a quasi-gradient dynamic, and that we can characterize the convergence rate to critical points. Let us first introduce an essential mechanism to obtain such convergence rates: the Kurdyka-Łojasiewicz (KL) property.

4.1 The Nonsmooth Kurdyka-Łojasiewicz Property for the Clarke Subdifferential

The nonsmooth Kurdyka-Łojasiewicz (KL) property, as introduced in (Bolte et al., 2010), is a measure of "amenability to sharpness". Here we provide a uniform version for the Clarke subdifferential of semi-algebraic functions as in Bolte et al. (2007) and Bolte et al. (2014). The details of the proof are left to the reader.[5]

Lemma 11 (Uniform Nonsmooth KL Property for the Clarke Subdifferential) *Let K be a nonempty compact set and let $L : \mathbb{R}^P \to \mathbb{R}$ be a semi-algebraic locally Lipschitz continuous function. Assume that L is constant on K, with value $L^\star$. Then there exist $\varepsilon > 0$, $\delta > 0$, $a \in (0, 1)$ and $\rho > 0$ such that, for all*

$$v \in \left\{v \in \mathbb{R}^P, \operatorname{dist}(v, \mathsf{K}) < \varepsilon\right\} \cap \left\{v \in \mathbb{R}^P, L^\star < L(v) < L^\star + \delta\right\},$$

it holds

$$\rho(1-a)\left(L(v) - L(\bar{v})\right)^{-a} \operatorname{dist}\left(0, \partial L(v)\right) > 1. \tag{21}$$

In the sequel, we make an abuse of notation by writing $\|\partial\mathcal{J}(\cdot)\| \triangleq \operatorname{dist}(0, \partial\mathcal{J}(\cdot))$. To obtain a convergence rate we will use inequality (21) on the Lyapunov function E. But first we state a general result of convergence that is built around the KL property.

4.2 A General Asymptotic Rate Result

We state a general theorem that leads to the existence of a convergence rate. This Theorem will hold in particular for (6). First, we start by stating the result.

Theorem 12 *Let $X : [0, +\infty) \to \mathbb{R}^P$ be a bounded absolutely continuous trajectory and let $L : \mathbb{R}^P \to \mathbb{R}$ be a semi-algebraic locally Lipschitz continuous function. If there exist $c_1 > 0$ such that for a.e. $t > 0$,*

$$\frac{\mathrm{d}L}{\mathrm{d}t}(X(t)) \leq -c_1 \|(\partial L)(X(t))\|^2. \tag{i}$$

4. We could extend the results of this section to more general objects including analytic functions on bounded sets, semi-algebraicity assumption is made for the sake of clarity.
5. One may directly use the general inequality provided in Bolte et al. (2007) or combine the local result of Bolte et al. (2007) with the compactness arguments Bolte et al. (2014, Lemma 6).

Then $L(X(t))$ converges to a limit value $L^\star$ and,

$$|L(X(t)) - L^\star| = \mathcal{O}\left(\frac{1}{t}\right).$$

If in addition there exists $c_2 > 0$ such that for a.e. $t > 0$,

$$c_2\|\dot{X}(t)\| \leq \|(\partial L)(X(t))\|, \tag{ii}$$

then, X converges to a critical point of L with a rate[6] of the form $O(1/t^b)$ with $b > 0$.

Proof We first prove the convergence of $L(X(\cdot))$. Suppose that (i) holds. Since X is bounded and L is continuous, $L(X(\cdot))$ is bounded. Moreover from (i), $L(X(\cdot))$ is decreasing, so it converges to some value $L^\star$. To simplify suppose $L \geq 0$ and $L^\star = 0$. Define,

$$\mathsf{I} = \left\{x \in \mathbb{R}^P \;\middle|\; L(x) = 0\right\}.$$

Suppose first that there exists $s \geq 0$, such that $X(s) \in \mathsf{I}$. Since $L(X(\cdot))$ is decreasing with limit 0, then for all $t \geq s$, $L(X(t)) = 0$ and the convergence rate holds true.

Let us thus assume that for all $t \geq 0$, $L(X(t)) > 0$. The trajectory X is bounded in $\mathbb{R}^P$, hence there exists a compact set $\mathsf{C} \subset \mathbb{R}^P$ such that $X(t) \in \mathsf{C}$ for all $t \geq 0$. Define $\mathsf{K} = \mathsf{I} \cap \mathsf{C}$, it is a compact set since I is closed (by continuity of L) and C is compact. Moreover, L is constant on K so by Lemma 11, there exist $\varepsilon > 0$, $\delta > 0$, $a \in (0, 1)$ and a constant $\rho > 0$ such that for all

$$v \in \left\{v \in \mathbb{R}^P, \mathrm{dist}(v, \mathsf{K}) < \varepsilon\right\} \cap \left\{0 < L(v) < \delta\right\},$$

it holds,

$$\rho(1-a)\left(L(v)\right)^{-a} \mathrm{dist}\left(0, \partial L(v)\right) > 1.$$

We have $L(X(t)) \to 0$ so there exists $t_0 \geq 0$ such that for all $t \geq t_0$, $0 < L(X(t)) < \delta$. Without loss of generality, we assume $t_0 = 0$. Similarly, we have $\mathrm{dist}(X(t), K) \to 0$, so we may assume that for all $t \geq 0$, $\mathrm{dist}\left(X(t), \mathsf{K}\right) < \varepsilon$. Thus, for all $t \geq 0$,

$$\rho(1-a)L(X(t))^{-a}\|\partial L(X(t))\| > 1.$$

Going back to assumption (i), for a.e. $t > 0$, one has,

$$\frac{\mathrm{d}L}{\mathrm{d}t}(X(t)) \leq -c_1\|(\partial L)(X(t))\|^2,$$

but the KL property implies that for a.e. $t > 0$,

$$-\|\partial L(X(t))\|^2 < -\frac{1}{\rho^2(1-a)^2}L(X(t))^{2a}.$$

Therefore,

$$\frac{\mathrm{d}L}{\mathrm{d}t}(X(t)) < -\frac{c_1}{\rho^2(1-a)^2}L(X(t))^{2a}.$$

6. Linear rate or finite convergence are possible.

We split into two cases depending on the value of a. If $0 < a \leq 1/2$, then for t large, $L(X(t)) < 1$ so $-L(X(t))^{2a} < -L(X(t))$ and hence,

$$\frac{\mathrm{d}L}{\mathrm{d}t}(X(t)) < -\frac{c_1}{\rho^2(1-a)^2}L(X(t)),$$

so we get a linear rate. When $1/2 < a < 1$, we have for a.e. $t > 0$,

$$L(X(t))^{-2a}\frac{\mathrm{d}}{\mathrm{d}t}L(X(t)) = \frac{1}{1-2a}\frac{\mathrm{d}}{\mathrm{d}t}L(X(t))^{1-2a} < -\frac{c_1}{(\rho^2(1-a)^2)}, \tag{22}$$

with $1 - 2a < 0$. We can integrate from 0 to $t > 0$:

$$L(X(t))^{1-2a} > \frac{(2a-1)c_1}{\rho^2(1-a)^2}t + L(X(0))^{1-2a} > \frac{(2a-1)c_1}{\rho^2(1-a)^2}t.$$

Since $\frac{1}{1-2a} < -1$, one obtains a convergence rate of the form $\mathcal{O}\left(t^{\frac{1}{1-2a}}\right)$. In both cases the rate is at least $\mathcal{O}\left(\frac{1}{t}\right)$.

We assume now that both (i) and (ii) holds and prove convergence of the trajectory with a convergence rate. Let $t > s > 0$, by the fundamental theorem of calculus and the triangular inequality:

$$\|X(t) - X(s)\| \leq \left\| \int_s^t \dot{X}(\tau)\,\mathrm{d}\tau \right\| \leq \int_s^t \|\dot{X}(\tau)\|\,\mathrm{d}\tau. \tag{23}$$

We wish to bound $\|\dot{X}\|$ using L. Using the chain rule (Lemma 5 of Section 3.2), for a.e. $\tau > 0$,

$$\frac{\mathrm{d}}{\mathrm{d}\tau}L(X(\tau))^{1-a} = (1-a)L(X(\tau))^{-a}\langle \dot{X}(\tau), (\partial L)(X(\tau))\rangle. \tag{24}$$

Then, from (i), we deduce that for a.e. $\tau > 0$,

$$\langle \dot{X}(\tau), (\partial L)(X(\tau))\rangle = \frac{\mathrm{d}L}{\mathrm{d}\tau}(X(\tau)) \leq -c_1\|(\partial L)(X(\tau))\|^2, \tag{25}$$

so

$$\frac{\mathrm{d}}{\mathrm{d}\tau}L(X(\tau))^{1-a} \leq -c_1(1-a)L(X(\tau))^{-a}\|(\partial L)(X(\tau))\|^2. \tag{26}$$

The KL property (21) implies that for a.e. $\tau > 0$,

$$-(1-a)L(X(\tau))^{-a}\|(\partial L)(X(\tau))\| < -\frac{1}{\rho}. \tag{27}$$

Putting this in (26) and using assumption (ii) we finally obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}L(X(\tau))^{1-a} < -\frac{c_1}{\rho}\|(\partial L)(X(\tau))\| \leq -\frac{c_1c_2}{\rho}\|\dot{X}(\tau)\|. \tag{28}$$

We can use that in (23):

$$\begin{aligned}\|X(t) - X(s)\| &\leq -\frac{\rho}{c_1c_2}\int_s^t \frac{\mathrm{d}}{\mathrm{d}t}L(X(\tau))^{1-a}\,\mathrm{d}\tau \\ &= \frac{\rho}{c_1c_2}(L(X(s)^{1-a} - L(X(t)^{1-a}).\end{aligned} \tag{29}$$

Then, using the convergence rate we already proved for L, we deduce that the Cauchy criterion holds for X inside the compact (hence complete) subset $\mathsf{C} \subset \mathbb{R}^P$ containing the trajectory. Thus, X converges, and from (i), we have that $\liminf_{t\to+\infty} \|\partial L(X(t))\| = 0$, since ∂L has closed graph, this shows that the limit is a critical points of L. Finally, taking the limit in (29) and using the convergence rate of L we obtain a rate for X as well. ■

Remark 13 Theorem (12) takes the form of a general recipe to obtain a convergence rate since it may be applied in many cases, to curves or flows, provided that a convenient Lyapunov function is given. Note also that it is sufficient for assumptions (i) and (ii) to hold only after some time $t_0 > 0$, in such case, simply do a time shift to use the theorem.

4.3 Application to INDIAN

We now apply Theorem 12 to the deterministic continuous dynamical model of INDIAN (6).

Theorem 14 (Convergence rates) *Suppose that $\mathcal{J}$ is semi-algebraic locally Lipschitz continuous and lower bounded. Then, any bounded trajectory (θ, ψ) solution to (6) converges to a point $(\bar{\theta}, \bar{\psi}) \in \mathsf{S}$, with a convergence rate of the form $\mathcal{O}\left(t^{-b}\right)$ with $b > 0$. Moreover $\mathcal{J}(\theta(t))$ converges to its limit $\bar{\mathcal{J}}$ with a rate: $\left|\mathcal{J}(\theta(t)) - \bar{\mathcal{J}}\right| = \mathcal{O}\left(\frac{1}{t}\right)$.*

Proof Let (θ, ψ) be a bounded solution of (6). We would like to use Theorem 12 with $X \equiv (\theta, \psi)$, and a well chosen function. In the proof of Theorem 3 we proved a descent property along the trajectory for the function $E(\theta, \psi) = 2(1 + \alpha\beta)\mathcal{J}(\theta, \psi) + \left\|(\alpha - \frac{1}{\beta})\theta + \frac{1}{\beta}\psi\right\|^2$. This function is semi-algebraic, locally Lipschitz continuous, so it remains to prove that (i) and (ii) hold for E along (θ, ψ).

For $t \geq 0$, denote $w(t) = (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t)$, then according to Lemma 9 for a.e. $t > 0$,

$$\begin{aligned}
\frac{\mathrm{d}E}{\mathrm{d}t}(\theta(t), \psi(t)) &= -\|\sqrt{\alpha}\dot{\theta}(t) - \frac{1}{\sqrt{\beta}}\left(\dot{\psi}(t) - \dot{\theta}(t)\right)\|^2 - \|\sqrt{\alpha}\dot{\theta}(t) + \frac{1}{\sqrt{\beta}}\left(\dot{\psi}(t) - \dot{\theta}(t)\right)\|^2 \\
&= -2\alpha\|\dot{\theta}(t)\|^2 - \frac{2}{\beta}\|\dot{\psi}(t) - \dot{\theta}(t)\|^2 = -2\alpha\|\dot{\theta}(t)\|^2 - \frac{2}{\beta}\|\beta\partial\mathcal{J}(\theta(t))\|^2 \\
&= -2\alpha\| - \beta\partial\mathcal{J}(\theta(t)) - w(t)\|^2 - 2\beta\|\partial\mathcal{J}(\theta(t))\|^2.
\end{aligned} \tag{30}$$

On the other hand, by standard results on the sum rule, we have for all $(\theta, \psi) \in \mathbb{R}^P \times \mathbb{R}^P$:

$$\partial E(\theta, \psi) = 2\begin{pmatrix} (1 + \alpha\beta)\partial\mathcal{J}(\theta) + (\alpha - \frac{1}{\beta})\left((\alpha - \frac{1}{\beta})\theta + \frac{1}{\beta}\psi\right) \\ \frac{1}{\beta}\left((\alpha - \frac{1}{\beta})\theta + \frac{1}{\beta}\psi\right) \end{pmatrix}, \tag{31}$$

so for a.e. $t > 0$,

$$\frac{\|\partial E(\theta(t), \psi(t))\|^2}{4} = \left\|(1 + \alpha\beta)\partial\mathcal{J}(\theta(t)) + (\alpha - \frac{1}{\beta})w(t)\right\|^2 + \left\|\frac{1}{\beta}w(t)\right\|^2. \tag{32}$$

We wish to find $c_1 > 0$, such that $\frac{1}{2}\frac{\mathrm{d}E}{\mathrm{d}t} + \frac{c_1}{4}\|\partial E\|^2 < 0$. This follows from the following claim.

Claim: let $r_1 > 0$, $r_2 \in \mathbb{R}$, $r_3 > 0$, Then there exist C_1 and C_2 two positive constants such that for any $a, b \in \mathbb{R}$:

$$C_1(a^2 + b^2) \leq (r_1 a + r_2 b)^2 + r_3 b^2 \leq C_2(a^2 + b^2) \tag{33}$$

Indeed, the function $Q : (a, b) \mapsto (r_1 a + r_2 b)^2 + r_3 b^2$ is a positive definite quadratic form, C_1 and C_2 can be taken to be two eigenvalues of the positive definite matrix which represents Q. Whence (33) holds for all a and b.

Using the previous claim to (32) and (30) leads to the existence of $c_1 > 0$ such that for a.e. $t > 0$,

$$\frac{\mathrm{d}E}{\mathrm{d}t}(\theta(t), \psi(t)) \leq -c_1 \|\partial E(\theta(t), \psi(t))\|^2,$$

so assumption (i) holds for INDIAN.

It now remains to show that (ii) of Theorem 12 holds: which means that there exists $c_2 > 0$ such that for (θ, ψ) solution of (6) and for a.e. $t > 0$, $\|\partial E(\theta(t), \psi(t))\|^2 \geq c_2 \left(\|\dot{\theta}(t)\|^2 + \|\dot{\psi}(t)\|^2 \right)$. Using (6) and (32) we get:

$$\frac{\|\partial E(\theta(t), \psi(t))\|^2}{4} = \left\| \frac{1}{\beta}(1 + \alpha\beta)\dot{\theta}(t) + \left[(\alpha - \frac{1}{\beta}) - \frac{1}{\beta}(1 + \alpha\beta) \right] \dot{\psi}(t) \right\|^2 + \frac{1}{\beta^2} \|\dot{\psi}(t)\|^2, \tag{34}$$

and applying the claim (33) again to (34) one can show that there exist $c_2 > 0$, such that for a.e. $t > 0$,

$$\|\partial E(\theta(t), \psi(t))\|^2 \geq c_2 \left(\|\dot{\theta}(t)\|^2 + \|\dot{\psi}(t)\|^2 \right).$$

So assumption (ii) holds for (6). To conclude, we can apply Theorem 12 to (6) and the proof is complete. ■

Remark 15 (a) These result suggest that similar behaviours and rates could be observed for INDIAN itself.
(b) The proof above is significantly simpler when $\alpha\beta > 1$ since Alvarez et al. (2002) proved that in this case, (6) is equivalent to a gradient system, thus assumptions (i) and (ii) of Theorem 12 instantly holds.
(c) Theorems 12 and 14 can be adapted to the case when the Clarke subdifferential is replaced by $D\mathcal{J}$, but we do not state it here for the sake of simplicity.

5. Experiments

In this section we first provide some insights into the hyper-parameters. This is done through some explanations of the 2D examples Figure 1 given in the introduction. We then compare the performance of INDIAN with those of other algorithms on DNN for image recognition.

5.1 Understanding the Role of the Hyper-parameters

Both hyper-parameters α and β can be seen as damping coefficients from the viewpoint of mechanics as discussed by Alvarez et al. (2002) and sketched in the introduction. Recall the second-order time continuous dynamics which served as a model to the design INDIAN :

$$\text{(DIN)} \quad \ddot{\theta}(t) + \alpha\, \dot{\theta}(t) + \beta\, \nabla^2 \mathcal{J}(\theta(t))\dot{\theta}(t) + \nabla \mathcal{J}(\theta(t)) = 0.$$

This differential equation was inspired by Newton's Second Law of dynamics asserting that the acceleration of a material point coincides with the sum of forces applied to the particle. As recalled in the introduction three forces are at stake: the gravity and two friction terms. The parameter α calibrates the *viscous damping* intensity as in the Heavy Ball friction method of Polyak (1964). It acts as a dissipation term but it can also be seen as a proximity parameter of the system with the usual gradient descent: the higher α is, the more DIN behaves like a pure gradient descent.[7] On the other hand the parameter β can be seen as a *Newton damping* which takes into account the geometry of the landscape to brake or accelerate the dynamics in an adaptive anisotropic fashion, see Alvarez and Pérez (1998); Alvarez et al. (2002) for further insights.

We now turn our attention to INDIAN, and illustrate the versatility of the hyper-parameters α and β in this case. We proceed on a 2D visual nonsmooth ill-conditioned example à la Rosenbrock, see Figure 1. For this example, we aim at finding the minimum of the function $\mathcal{J}(\theta_1, \theta_2) = 100(\theta_2 - |\theta_1|)^2 + |1 - \theta_1|$. This function has a *V*-shaped valley, and a unique critical point at $(1, 1)$ which is also the global minimum. Starting from the point $(-1, 1.5)$ (the black cross), we apply INDIAN with constant steps $\gamma_k = 10^{-4}$. Figure 1 shows that when β is too small, the trajectory presents many transverse oscillations as well as longitudinal ones close to the critical point (subplot (a)). Then, increasing β significantly reduces transverse/parasite oscillations (subplot (b)). Finally, the longitudinal oscillations are reduced by choosing a higher α (subplot (c)). In addition, these behaviors are also reflected in the values of the objective function (subplot (d)).

The orange curve (first setting) presents large oscillations. Moreover, looking at the red curve, corresponding to plot (c), there is a short period between $20,000$ and $60,000$ iterations when the decrease is slower than for the other values of α and β, but still it presents fewer oscillations. In the longer term, the third choice ($\alpha = 1.3$, $\beta = 0.1$) provides remarkably good performance.

5.2 Training a DNN with INDIAN

Before using INDIAN to train a deep neural network, we first describe the methodology we followed.

5.2.1 Methodology

We now compare INDIAN with popular algorithms used in deep learning (SGD, ADAM, ADAGRAD). Here is the detailed methodology.

- We train a DNN for classification using the three most common image data sets (MNIST, CIFAR10, CIFAR100) (LeCun et al., 1998; Krizhevsky, 2009). These data sets are composed of $60,000$ small images associated with a label (numbers, objects, animals, etc.). We split the data sets into $50,000$ images for the training part and $10,000$ for the test.

- Regarding the network, we use a slightly modified version of the popular Network in Network (NiN) (Lin et al., 2013). It is a reasonably large convolutional network with $P \sim 10^6$ parameters to optimize. We use ReLU activation functions.

- We take the cross-entropy as dissimilarity measure ℓ, and use the corresponding loss $\mathcal{J}$ of the form (2) to quantify the training performance. We assess the accuracy of the trained DNNs using the test data set that contains $10,000$ images. Here measuring accuracy simply boils down to counting how many of the $10,000$ were correctly classified (in percentage).

7. This is easier to see when one rescales $\mathcal{J}$ by α.

– Based on Section 5.1 we run INDIAN for four different values of (α, β):

$$(\alpha, \beta) \in \{(0.1, 0.1), (0.5, 0.1), (0.5, 0.5), (0.5, 1)\}.$$

To compare INDIAN to other algorithms, we only present the best choice of parameters for each problem on Figure 2. For INDIAN though, we display the results for all choices of hyper-parameters in Figure 3 and we observe satisfying results in the four cases. Given the weights θ_0, we initialize ψ_0 such that the initial velocity is in the direction of $-\nabla\mathcal{J}(\theta_0)$. Hence we use $\psi_0 = (1 - \alpha\beta)\theta_0 - (\beta^2 - \beta)\nabla\mathcal{J}(\theta_0)$.

– We compare our algorithm to both the classical stochastic gradient descent algorithm and the very popular ADAGRAD (Duchi et al., 2011) and ADAM (Kingma and Ba, 2014) algorithms. At each iteration, we compute the approximation of $\partial\mathcal{J}(\theta)$ on a subset $\mathsf{B} \subset \{1, \ldots, 50000\}$ of size 32. To do a fair comparison, each algorithm is initialized with the same random weights (following a normal distribution). To obtain more relevant results, this process is done for five different random initializations θ_0.

– On step sizes. Both ADAGRAD and ADAM's steps follow an adaptive procedure based on past gradients, see Duchi et al. (2011); Kingma and Ba (2014). For the other two algorithms (INDIAN and SGD), we take the classical step size schedule $\gamma_k = \frac{\gamma_0}{\sqrt{k+1}}$, which meets Assumption 1. For all four algorithms, choosing the right initial step length γ_0 is often critical in terms of efficiency. We use a grid-search for each algorithm and chose the initial step size that most decreases the loss function over fifteen epochs (fifteen complete passes over the data). Remark that we could use more flexible step size schedules, we chose the classical decrease scheme for the sake of simplicity and investigate different exponents in further experiments.

For these experiments, we used `Keras 2.2.4` (Chollet, 2015) with `Tensorflow 1.13.1` (Abadi et al., 2016) as backend. The INDIAN algorithm is available in Pytorch, Keras and Tensorflow: `https://github.com/camcastera/Indian-for-DeepLearning/` (Castera, 2019).

5.2.2 Results

On each of the figures below, solid lines represent mean values and pale surfaces represent the best and worst runs in terms of training loss and validation accuracy over five random initializations.

We first compare INDIAN with the other algorithms in Figure 2. The scores reach the state of the art on NiN and represent what can be achieved with a moderately large network and a coarse grid-search tuning. In our comparison, ADAM and INDIAN outperform SGD and ADAGRAD for training. While ADAM seems to be faster in the early training phase, INDIAN achieves the best accuracy almost every time especially on CIFAR-100 (Figure 2(b)). Thus INDIAN appears to be competitive in comparison to the other algorithms with the advantage of having an elementary and transparent step size rule compared to ADAM and ADAGRAD.

Regarding the tuning of the hyper-parameters α and β, Figure 3 suggests that it is not too crucial to obtain satisfactory results both for training and testing. It mostly affects the training speed. Thus, INDIAN looks quite stable with respect to these hyper-parameters. For example $(\alpha, \beta) = (0.5, 0.1)$ might be a good default choice at least on each of these problems. Nevertheless, tuning these hyper-parameters appears necessary to get the most from INDIAN.

Finally let us point out that altough ADAM was faster in the experiments of Figure 2, INDIAN can outperform ADAM using Remark 4. Indeed, in the previous experiments we used a decreasing

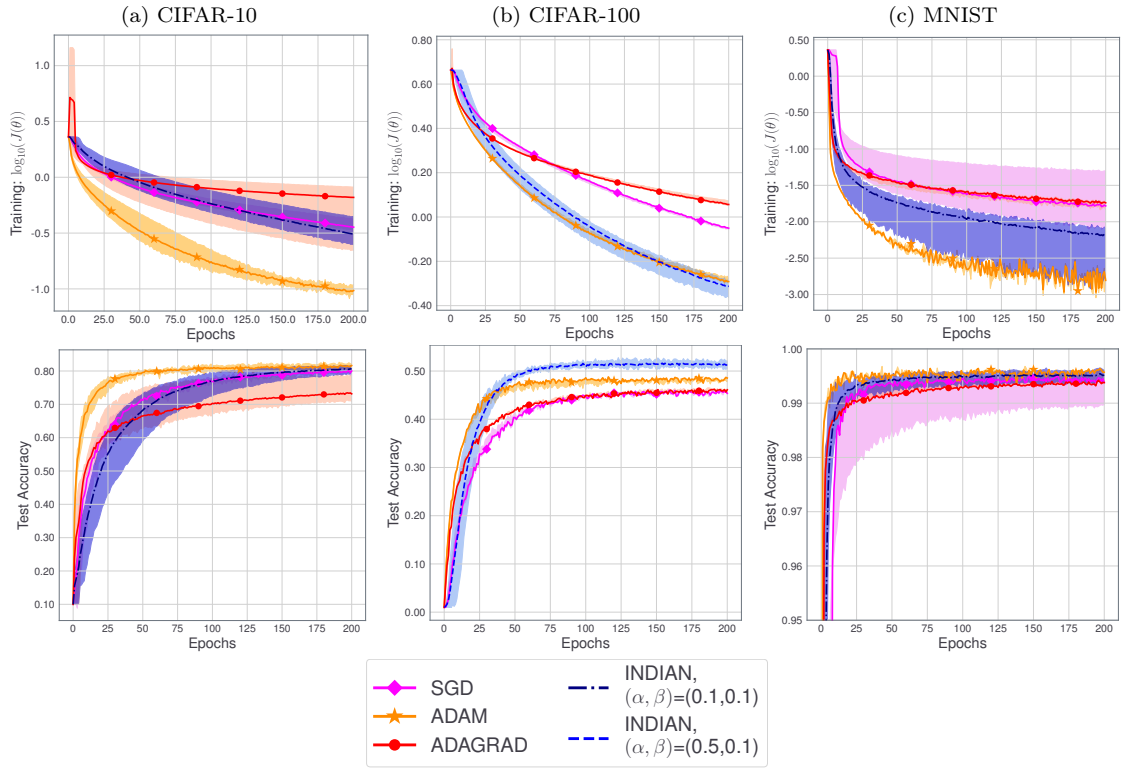


Figure 2: Train loss and test accuracy using NiN for three different image classification problems. The first row of figures shows the evolution of the logarithm of the loss functions $\mathcal{J}(\theta)$ during the training. The figures below are the percentage accuracy on the test set. On each figure we compare SGD, ADAM and ADAGRAD to INDIAN either with $(\alpha, \beta) = (0.1, 0.1)$ or $(\alpha, \beta) = (0.5, 0.1)$ depending on which one gave the best results (see the methodology in Section 5.2).

step size of the form $\gamma_0/\sqrt{k+1}$ since it is the most common choice, but Assumption 1 allows for step sizes decreasing much slower, thus we tried decays of the form $\gamma_0(k+1)^{-q}$ with $q \leq 1/2$. The results are displayed on top of Figure 4. It appears that a slower decay makes INDIAN a little bit faster than any of the other algorithms we tried. In particular, with a step size decay proportional to $k^{-1/4}$, INDIAN outperform ADAM (bottom of Figure 4). Let alone the flexibility that it offers once again, this suggests that an appropriate step size decay may be one of the key to accelerate the training process.

6. Conclusion

We introduced a novel stochastic optimization algorithm featuring inertial and Newtonian behaviour motivated by applications to Deep Learning. We provided a powerful algorithmic convergence

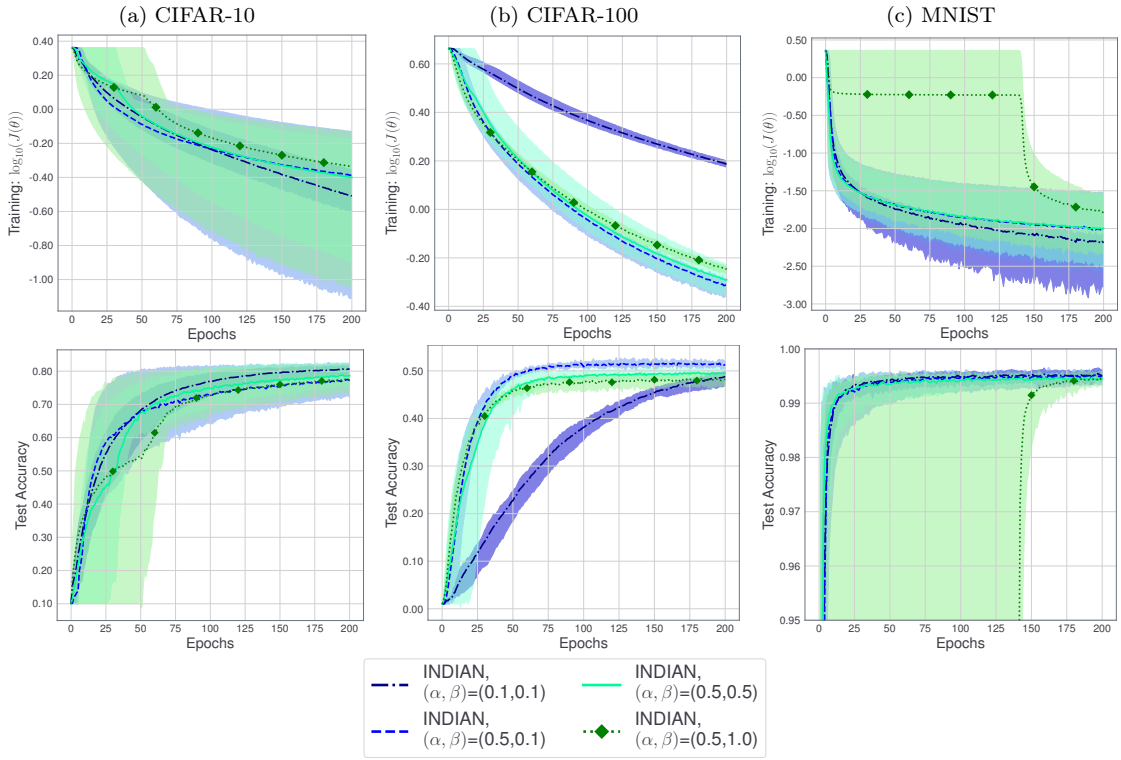


Figure 3: Analysis of the sensibility of INDIAN to the choice of α and β, the experiments were done in the same framework as that described in Figure 2. It represents training loss and testing accuracy using NiN for three different image classification problems. Top of the figure represents the training losses in $\log_{10}$ scale, and test accuracy is shown on the second row of figures.

analysis under weak hypotheses applicable to most Deep Learning problems. We also provided new general results to study differential inclusions on Clarke subdifferential and obtain convergence rates for the continuous time counterpart of our algorithm. We would like to point out that, apart from SGD (Davis et al., 2019), the convergence of concurrent methods in such a general setting is still an open question. Our result seems moreover to be the first one able to handle the mini-batch subsampling approach for ReLU DNNs via the introduction of the D-critical points. Our experiments show that INDIAN is very competitive with state-of-the-art algorithms for DL but also very malleable. We stress that these numerical manipulations were performed on substantial DL benchmarks with only minimal algorithm tuning (very classical step sizes with a simple grid search on a few epochs to set the initial step size). This facilitates reproducibility and allows to stay as close as possible to the reality of DL applications in machine learning.

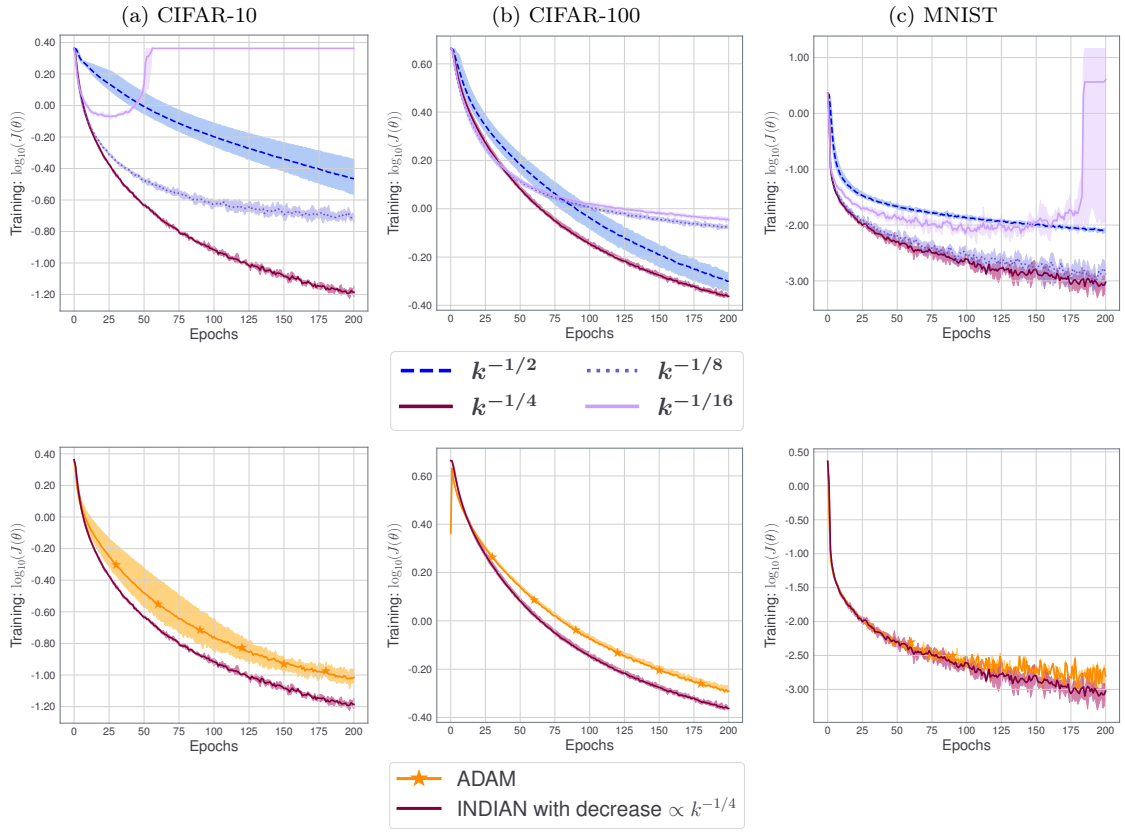


Figure 4: On top: Training loss of INDIAN on three image classification problems with various step size decays. In the legend, k^{-q} means a step size decay at iteration k of the form $\gamma_k = \gamma_0 k^{-q}$. The bottom row show the comparison between INDIAN with a well-chosen step size decay and ADAM.

Acknowledgments

This work has been supported by the European Research Council (ERC FACTORY-CoG-6681839) and ANR-3IA Artificial and Natural Intelligence Toulouse Institute. The authors acknowledge the support of Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA9550-18-1-0226, ANR CHESS under grant ANR-17-EURE-0010 (Investissements d'Avenir program) and MASDOL under grant ANR-PRC-CE23.

Part of the numerical experiments were done on the OSIRIM platform of IRIT, supported by the CNRS, the FEDER, the Occitanie region and the French government (`http://osirim.irit.fr/site/en`).

We thank Hedy Attouch and Sixin Zhang for useful discussions.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283, 2016.

Salim Adil. *Opérateurs monotones aléatoires et application à l'optimisation stochastique*. PhD Thesis, Paris Saclay, 2018.

Felipe Alvarez and JM Pérez. A dynamical system associated with Newton's method for parametric approximations of convex minimization problems. *Applied Mathematics and Optimization*, 38: 193–217, 1998.

Felipe Alvarez, Hedy Attouch, Jérôme Bolte, and P Redont. A second-order gradient-like dissipative dynamical system with Hessian-driven damping: Application to optimization and mechanics. *Journal de Mathématiques Pures et Appliquées*, 81(8):747–779, 2002.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

Jean-Pierre Aubin and Arrigo Cellina. *Differential inclusions: set-valued maps and viability theory*. Springer, 2012.

Anas Barakat and Pascal Bianchi. Convergence of the ADAM algorithm from a dynamical system viewpoint. *arXiv:1810.02263*, 2018.

Michel Benaïm. Dynamics of stochastic approximation algorithms. In *Séminaire de Probabilités XXXIII*, pages 1–68. Springer, 1999.

Michel Benaïm, Josef Hofbauer, and Sylvain Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

Albert S Berahas, Raghu Bollapragada, and Jorge Nocedal. An investigation of Newton-sketch and subsampled newton methods. *arXiv:1705.06211*, 2017.

Jérôme Bolte, Aris Daniilidis, Adrian Lewis, and Masahiro Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007.

Jérôme Bolte, Aris Daniilidis, Olivier Ley, and Laurent Mazet. Characterizations of łojasiewicz inequalities: subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

Vivek S Borkar. *Stochastic approximation: A dynamical systems viewpoint*. Springer, 2009.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 161–168, 2008.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Richard H Byrd, Gillian M Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3): 977–995, 2011.

Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

Camille Castera. INDIAN for deep learning. `https://github.com/camcastera/Indian-for-DeepLearning/`, 2019.

François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

Michel Coste. *An introduction to o-minimal geometry*. Istituti editoriali e poligrafici internazionali Pisa, 2000.

Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D Lee. Stochastic subgradient method converges on tame functions. *Foundations of Computational Mathematics*, 2019. (in press).

Lou van den Dries. *Tame topology and o-minimal structures*. Cambridge university press, 1998.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7):2121–2159, 2011.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 315–323, 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research, 2009.

Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pages 769–783, 1998.

Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*. Springer, 2003.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Lennart Ljung. Analysis of recursive stochastic algorithms. *IEEE Transactions on Automatic Control*, 22(4):551–575, 1977.

James Martens. Deep learning via Hessian-free optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 735–742, 2010.

Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 451–459, 2011.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(1):400–407, 1951.

David E Rumelhart and Geoffrey E Hinton. Learning representations by back-propagating errors. *Nature*, 323(9):533–536, 1986.

Arthur Sard. The measure of the critical values of differentiable maps. *Bulletin of the American Mathematical Society*, 48(12):883–890, 1942.

Masahiro Shiota. *Geometry of subanalytic and semialgebraic sets*, volume 150. Springer Science & Business Media, 2012.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4148–4158, 2017.

Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Second-order optimization for non-convex machine learning: An empirical study. *arXiv:1708.07827*, 2017.