An Inertial Newton Algorithm for Deep Learning

Camille Castera[†*], Jérôme Bolte[‡*], Cédric Févotte[†*], Edouard Pauwels[†§*]

† IRIT, CNRS * Univ. Toulouse, France
‡ Toulouse School of Economics § DEEL, IRT Saint Exupery

Contributions

- Building a **second-order** method with inertia for training deep networks.
- Physical interpretation of the **hyperparameters.**
- Proof of **convergence** in a very **general setting.**

Objective

Given a deep neural network f with parameters θ, a data set $(x_n, y_n)_{n=1\dots N}$,
Design algorithms to solve

$$\min_{\theta} \mathcal{J}(\theta) = \sum_{n=1}^{N} l(f(x_n,\theta), y_n)$$

Assumption

We focus on losses $\mathcal{J}$ that are **Continuous**, **locally Lipschitz**, and **Tame**. Hence, differentiable almost everywhere. **Covers most deep learning losses**.

From a Differential Equation to the Algorithm

1. Study the following second-order ODE (with $\alpha \geq 0, \beta > 0$),

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

2. Introduce an auxiliary variable to remove the explicit second-order derivatives:

$$\begin{cases} \dot{\theta}(t) + (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) + \beta\nabla\mathcal{J}(\theta(t)) & = 0 \\ \dot{\psi}(t) + (\alpha - \frac{1}{\beta})\theta(t) + \frac{1}{\beta}\psi(t) & = 0 \end{cases}$$

3. Discretize with an explicit Euler scheme at a time t_k with a step size $\gamma_k > 0$:

$$\dot{\theta}(t_k) \simeq \frac{\theta(t_k) - \theta(t_k - \gamma_k)}{\gamma_k}$$

The Algorithm: INNA

$$\begin{cases} \theta_{k+1} = \theta_k + \gamma_k\left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k - \beta\nabla\mathcal{J}(\theta_k)\right) \\ \psi_{k+1} = \psi_k + \gamma_k\left((\frac{1}{\beta} - \alpha)\theta_k - \frac{1}{\beta}\psi_k\right) \end{cases}$$

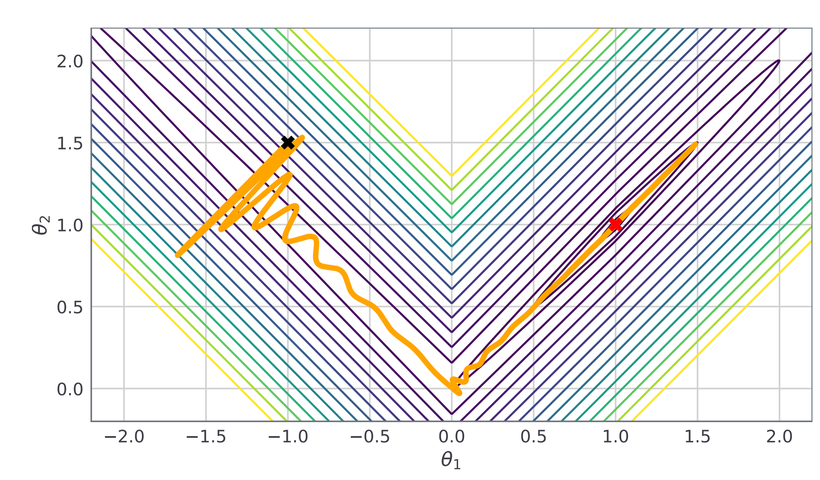
Mini-batch Subsampling

1. At each iteration, only consider a few data chosen randomly.
2. Produces a stochastic approximation of the gradient, up to a random noise ξ_k

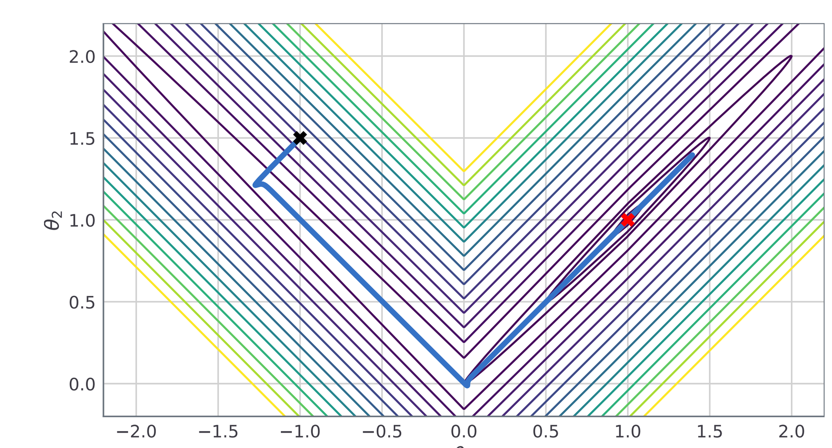
→ Overcomed by taking vanishing discretization step sizes γ_k.

Dynamical System Interpretation

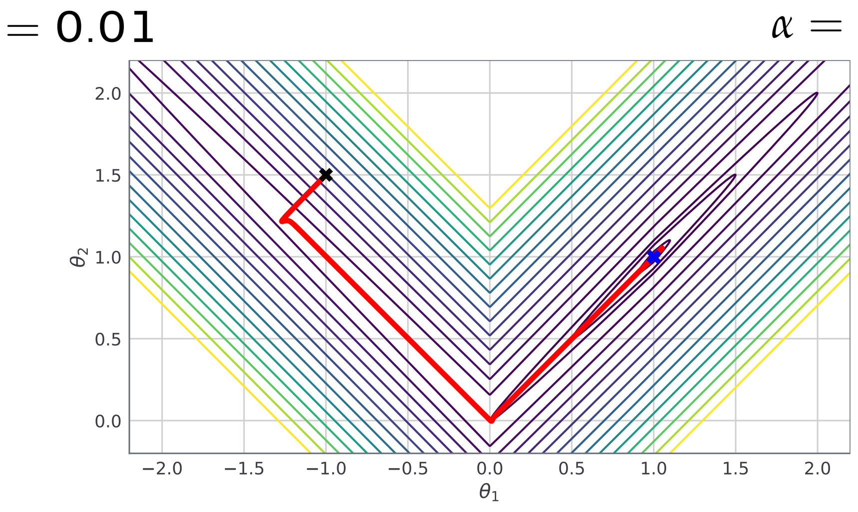
$$\underbrace{\ddot{\theta}(t)}_{\text{Inertia}} + \underbrace{\alpha\,\dot{\theta}(t)}_{\text{Friction}} + \underbrace{\beta\nabla^2\mathcal{J}(\theta(t))\dot{\theta}(t)}_{\text{Newtonian effects}} + \underbrace{\nabla\mathcal{J}(\theta(t))}_{\text{Gravity}} = 0$$



$\alpha = 0.5,\ \beta = 0.01$



$\alpha = 0.5,\ \beta = 0.1$



$\alpha = 1.3,\ \beta = 0.1$

Theoretical Guarantees

Theorem: INNA Converges

For any uniformly bounded sequence $(\theta_k, \psi_k)_k$ of INNA,

- Accumulation points $(\bar{\theta}, \bar{\psi})$ are such that $\nabla\mathcal{J}(\bar{\theta}) = 0$.
- The sequence of values $(\mathcal{J}(\theta_k))_{k\in\mathbb{N}}$ converges.

Proof Sketch

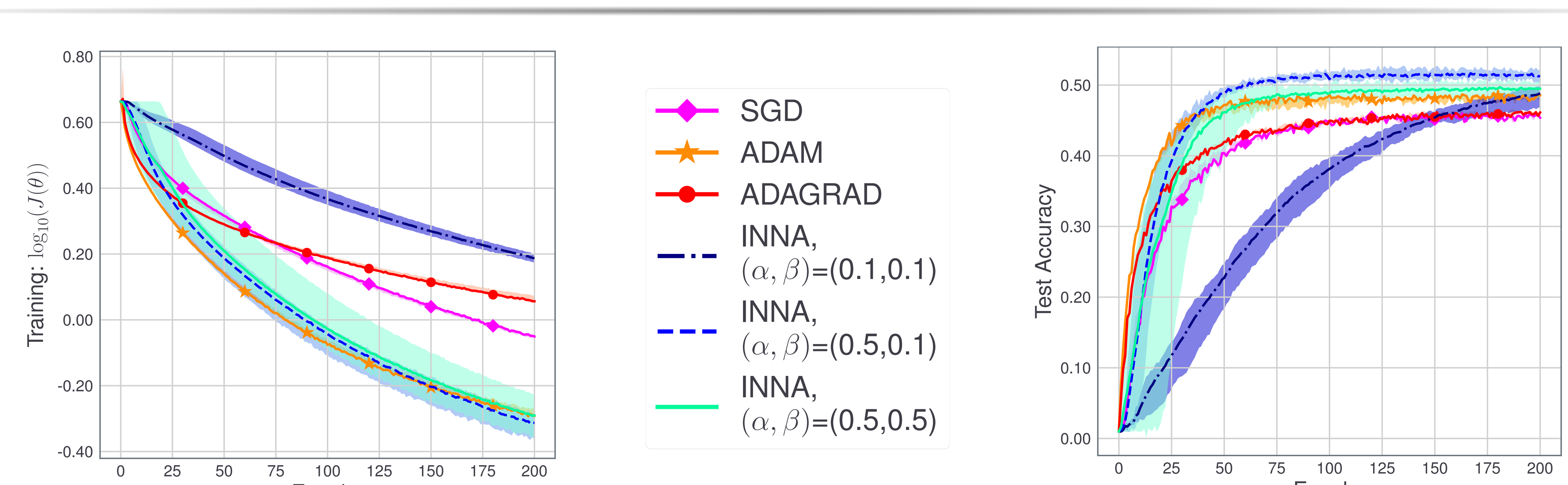
- Solutions of the continuous ODE converges to critical points. Control these solutions (Lyapunov Analysis).
- Control the noise ξ_k (vanishing step sizes).

→ INNA asymptotically behaves like the ODE.

Handling Nondifferentiable Losses

$\mathcal{J}$ Differentiable	$\mathcal{J}$ Nondifferentiable
Gradient $\nabla\mathcal{J}(\theta)$	Clarke Subgradient $\partial\mathcal{J}(\theta)$
Ordinary Differential Equation	Differential Inclusion
Chain rule for gradients $\frac{\partial\mathcal{J}}{\partial t}(\theta(t)) = \langle\nabla\mathcal{J}(\theta(t)), \dot{\theta}(t)\rangle$	Chain rule for subgradients $\frac{\partial\mathcal{J}}{\partial t}(\theta(t)) = \langle v_k, \dot{\theta}(t)\rangle$
Sum rule $\sum\nabla = \nabla\sum$	No sum rule $\sum\partial \neq \partial\sum$

Numerical Experiments



Training and test accuracy using *Network in Network* to classify images of the *CIFAR-100* data-set.

References

Castera C., Bolte J., Févotte C., Pauwels E.
An intertial Newton Algorithm for Deep Learning (2019)
https://arxiv.org/abs/1905.12278