SHINE: SHaring the INverse Estimate from the forward pass for bi-level optimization and implicit models

ICLR 2022 Spotlight

Zaccharie Ramzi
CEA & Inria Parietal

Joint work with: F. Mannel, S. Bai, P. Ciuciu, J.-L. Starck, T. Moreau

Deep Equilibrium networks

Deep Equilibrium networks (DEQs) (Bai, Kolter, et al., 2019) are a type of implicit model. The output is the solution to a fixed-point equation.

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{z}^\star, \text{ where } \boldsymbol{z}^\star = f_{\boldsymbol{\theta}}(\boldsymbol{z}^\star, \boldsymbol{x})$$

Deep Equilibrium networks

Deep Equilibrium networks (DEQs) (Bai, Kolter, et al., 2019) are a type of implicit model. The output is the solution to a fixed-point equation.

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{z}^\star, \text{ where } \boldsymbol{z}^\star = f_{\boldsymbol{\theta}}(\boldsymbol{z}^\star, \boldsymbol{x})$$

This approximates an infinite depth network:

$$\boldsymbol{z}_n = f_{\boldsymbol{\theta}}(\boldsymbol{z}_{n-1}), \quad \forall n \to \infty$$

In practice, we work with root finding algorithms using $g_{\boldsymbol{\theta}} = id - f_{\boldsymbol{\theta}}$.

quasi-Newton methods

For the forward pass' root finding problem

$$g(z^\star) = 0$$

Newton Methods: $z_{n+1} = z_n - J_g(z_n)^{-1} g(z_n)$

quasi-Newton methods

For the forward pass' root finding problem

$$g(z^\star) = 0$$

Newton Methods: $z_{n+1} = z_n - J_g(z_n)^{-1} g(z_n)$

Idea: replace the costly Jacobian inverse $J_g(z_n)^{-1}$ with a qN matrix $\boldsymbol{B}_n^{-1}$.

quasi-Newton Methods: $z_{n+1} = z_n - \boldsymbol{B}_n^{-1} g(z_n)$.

DEQ's backward pass

DEQs gradient computation using Implicit Function Theorem:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{z}^\star} = \nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}^\star)^\top \; J_{g_{\boldsymbol{\theta}}}(\boldsymbol{z}^\star)^{-1} \frac{\partial g_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{z}^\star},$$

we need to invert a huge matrix $J_{g_{\boldsymbol{\theta}}}(\boldsymbol{z}^\star)$ in a certain direction $\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}^\star)$.

In practice this is done using an iterative algorithm.

DEQs: Deep Equilibrium Networks

The limits of DEQs

DEQs achieve excellent results in NLP (Natural Language Processing) and CV (Computer Vision) tasks, but they are slow to train.

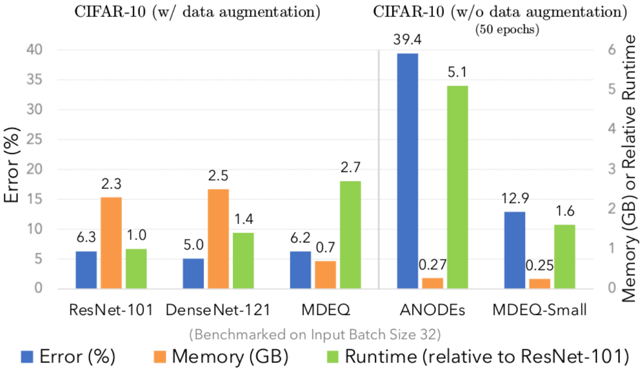

Figure: **Performance, memory and training speed of DEQs.** (Bai, Koltun, et al., 2020)

Can we avoid the Jacobian inversion?

We introduced **SHINE: SHaring the INverse Estimate**.

$$\boldsymbol{B}^{-1} \approx J_{\boldsymbol{g_\theta}}(\boldsymbol{z}^\star)^{-1}$$

Can we avoid the Jacobian inversion?

We introduced **SHINE: SHaring the INverse Estimate**.

$$\boxed{B^{-1}} \approx \underbrace{\boxed{J_{g_{\boldsymbol{\theta}}}(\boldsymbol{z}^\star)^{-1}}}_{\text{True Jacobian inverse}}$$

Can we avoid the Jacobian inversion?

We introduced **SHINE: SHaring the INverse Estimate**.

$$\boxed{\boldsymbol{B}^{-1}} \approx \boxed{J_{\boldsymbol{g_\theta}}(\boldsymbol{z}^\star)^{-1}}$$

quasi-Newton matrix

True Jacobian inverse

Can we avoid the Jacobian inversion?

We introduced **SHINE: SHaring the INverse Estimate**.

$$\boxed{\boldsymbol{B}^{-1}} \approx \boxed{J_{g_{\boldsymbol{\theta}}}(\boldsymbol{z}^\star)^{-1}}$$

quasi-Newton matrix

True Jacobian inverse

Properties of $\boxed{\boldsymbol{B}}$:

- It is computed when solving $\boldsymbol{z}^\star - f_{\boldsymbol{\theta}}(\boldsymbol{z}^\star, \boldsymbol{x}) = 0$ using a quasi-Newton method.

Can we avoid the Jacobian inversion?

We introduced **SHINE: SHaring the INverse Estimate**.

$$\underbrace{B^{-1}}_{\text{quasi-Newton matrix}} \approx \underbrace{J_{g_\theta}(z^\star)^{-1}}_{\text{True Jacobian inverse}}$$

Properties of B:

- It is computed when solving $z^\star - f_\theta(z^\star, x) = 0$ using a quasi-Newton method.
- It is easily invertible using the Sherman-Morrison formula, because low-rank.

SHINE direction convergence

Theorem (Convergence of SHINE to the Hypergradient using ULI)

Under the Uniform Linear Independence (ULI) assumption and some additional smoothness and convexity assumptions, for a given parameter $\boldsymbol{\theta}$, $(\boldsymbol{z}_n)$ *converges q-superlinearly to* $\boldsymbol{z}^\star$ *and*

$$\lim_{n\to\infty} \nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_n)^\top \boldsymbol{B}_n^{-1} \frac{\partial g_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{z}_n} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{z}^\star}.$$

Computer vision results

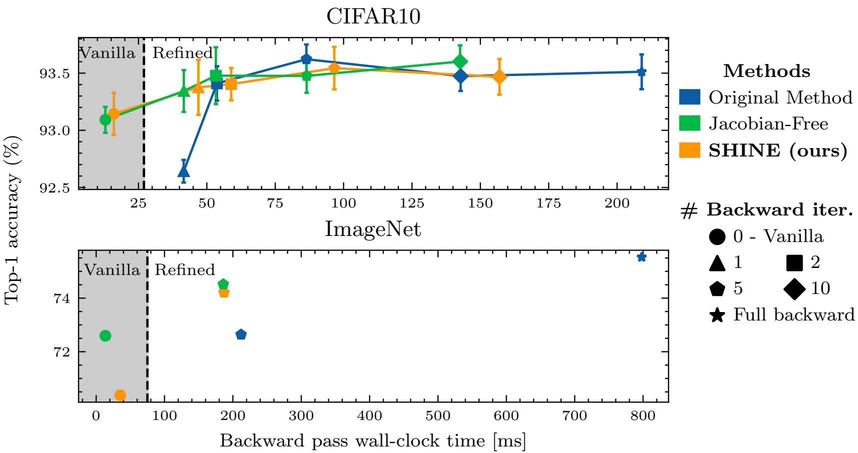

Figure: **MDEQs (Bai, Koltun, et al., 2020) with SHINE.**

Come check our poster #6363

SHINE accelerates the training of **Deep Equilibrium Networks**.

Come check our poster #6363

SHINE accelerates the training of **Deep Equilibrium Networks**.

Come chat with us at our poster to see:

- How we can obtain better theoretical guarantees by modifying the forward pass.
- Our application to Bi-Level optimzation.