A Bridging Framework for Model Optimization and Deep Propagation

Risheng Liu, Shichao Cheng, Xiaokun Liu, Long Ma, Xin Fan, and Zhongxuan Luo

Dalian University of Technology

Abstract

Three existed Motivations:

- Optimizing task-related mathematical model is widely used in statistic and learning areas. However, generally designed schematic iterations may hard to investigate complex data distributions in real-world applications.
- In recent years, deep neural networks (DNNs) have been established and trained in end-to-end manner for different learning and vision problems. Though with relatively good performance on specific applications, it is challenging to reason about what a DNN model actually does due to its opaque or black-box nature.
- Embedding DNNs into the optimization process is recently popular and some preliminary works have been developed from different perspectives. Unfortunately, it is challenging to provide convergence analysis on these trained iterations.

Contributions:

We proposed a theoretically guaranteed paradigm, named **Propagation and Optimization based Deep Model (PODM)**, to incorporate knowledge-driven schematic iterations and data-dependent network architectures to address both model optimization and learning tasks.

- Provided a **model-inspired paradigm** to establish building-block modules for deep model design.
- Provide the proof of the **global convergence**.
- The relaxed PODM actually provides a **plug-and-play, collaborative, interpretable, and end-to-end** deep learning framework for real-world complex tasks.

Model Inspired Building-Blocks

Existing training based iterative model:

$$\min_{\mathbf{x}} \Phi(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}), \tag{1}$$

where f denotes the loss term and g is related to the regularization term. Suppose $\mathcal{A}$ is the given network architecture and denote its output as $\mathbf{x}_{\mathcal{A}} = \mathcal{A}(\mathbf{x}; \boldsymbol{\theta}_{\mathcal{A}})$. We would like to design our propagation module based on both $\mathcal{A}$ and Φ

$$\begin{aligned} &\min_{\mathbf{x}} \Phi(\mathbf{x}) + d(\mathbf{x}, \mathbf{x}_{\mathcal{A}}) - \langle \mathbf{x}, \boldsymbol{\epsilon} \rangle \\ &= \min_{\mathbf{x}} \overbrace{\underbrace{f(\mathbf{x})}_{\text{Fidelity}} + \underbrace{g(\mathbf{x})}_{\text{Designed prior}}}^{\text{Knowledge}} + \overbrace{\underbrace{d(\mathbf{x}, \mathbf{x}_{\mathcal{A}})}_{\text{Learned prior}} - \underbrace{\langle \mathbf{x}, \boldsymbol{\epsilon} \rangle}_{\text{Error}}}^{\text{Data}}. \end{aligned} \tag{2}$$

Here, $d(\mathbf{x}, \mathbf{x}_{\mathcal{A}})$ is the distance function which intents to introduce the output of network into the propagation module.

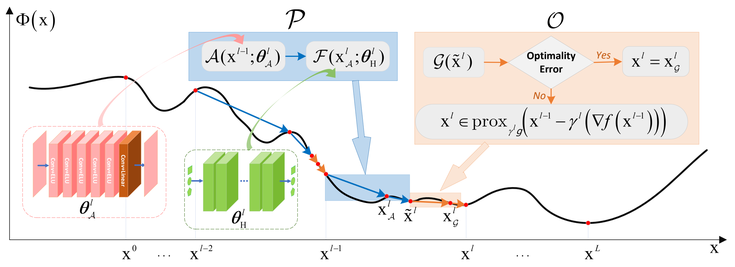


Figure 1: Illustrating the mechanism of PODM for nonconvex model optimization.

- **Propagation Module:**

We first investigate the following sub-model of Eq. (2) (i.e., only with fidelity and learned priors)

$$\mathbf{x}_{\mathcal{F}} = \mathcal{F}(\mathbf{x}_{\mathcal{A}}; \boldsymbol{\theta}_{\mathbf{H}}) := \arg\min_{\mathbf{x}} \{f(\mathbf{x}) + d(\mathbf{x}, \mathbf{x}_{\mathcal{A}})\}, \tag{3}$$

where $\boldsymbol{\theta}_{\mathbf{H}}$ denotes the parameter in distance $d(\mathbf{x}, \mathbf{x}_{\mathcal{A}})$. we can define our data-dependent propagation module $(\mathcal{P})$ as the cascade of $\mathcal{A}$ and $\mathcal{F}$ in the l-th stage, i.e.,

$$\tilde{\mathbf{x}}^l = \mathcal{P}(\mathbf{x}^{l-1}; \boldsymbol{\vartheta}^l) := \mathcal{F}\left(\mathcal{A}\left(\mathbf{x}^{l-1}; \boldsymbol{\theta}^l_{\mathcal{A}}\right); \boldsymbol{\theta}^l_{\mathbf{H}}\right),$$

where $\boldsymbol{\vartheta}^l = \{\boldsymbol{\theta}^l_{\mathcal{A}}, \boldsymbol{\theta}^l_{\mathbf{H}}\}$ is the set of trainable parameters.

- **Optimality Module:**

we recall the designed prior g and assume $\mathbf{x}^l_{\mathcal{G}}$ is one solution of Eq. (2) in the l-th stage, i.e.,

$$\mathbf{x}^l_{\mathcal{G}} \in \mathcal{G}(\tilde{\mathbf{x}}^l) := \arg\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) + d(\mathbf{x}, \mathbf{x}^l_{\mathcal{A}}) - \langle \mathbf{x}, \boldsymbol{\epsilon}^l \rangle. \tag{4}$$

The error $\boldsymbol{\epsilon}^l$ has a specific form as $\boldsymbol{\epsilon}^l = \nabla f(\mathbf{x}^l_{\mathcal{G}}) + \nabla d(\mathbf{x}^l_{\mathcal{G}}, \mathbf{x}^l_{\mathcal{A}}) + \mathbf{u}_{\mathbf{x}^l_{\mathcal{G}}}$ by considering the first-order optimality condition of Eq. (4). we can demonstrate that the convergence of our deep propagations can be successfully guaranteed by the following optimality error:

$$\|\psi(\boldsymbol{\epsilon}^l)\| \leq c^l \|\mathbf{x}^l_{\mathcal{G}} - \mathbf{x}^{l-1}\|. \tag{5}$$

Here $\psi(\boldsymbol{\epsilon}^l) = \boldsymbol{\epsilon}^l + \mu^l(\mathbf{x}^l_{\mathcal{G}} - \mathbf{x}^{l-1})/2 - \mathbf{H}(\mathbf{x}^{l-1} + \mathbf{x}^l_{\mathcal{G}} - 2\mathbf{x}^l_{\mathcal{A}})$ is the error function and c^l is a positive constant to reveal our tolerance of the inexactness at the l-th stage.

Theorem

Suppose f is proper and Lipschitz smooth, g is proper and lower semi-continuous, and Φ is a semi-algebraic function and coercive. Then the output of PODM (i.e., $\{\mathbf{x}^l\}_{l\in\mathbb{N}}$) satisfies:

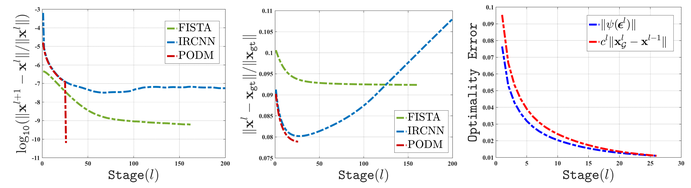
1. The limit points of $\{\mathbf{x}^l\}_{l\in\mathbb{N}}$ (denoted as Ω) is a compact set;
2. All elements of Ω are the critical points of Φ;
3. $\{\mathbf{x}^l\}_{l\in\mathbb{N}}$ converges to a critical point of Φ.

Relaxed PODM

We relax the theoretical constraint and develop a novel end-to-end learning framework to address real-world tasks. In particular, rather than only training the parameters $\boldsymbol{\theta}^l_{\mathcal{A}}$ in given network $\mathcal{A}$, we also introduce flexible networks to learn parameters $\boldsymbol{\theta}^l_{\mathbf{H}}$ in $\mathcal{F}$ at each layer. The forward propagation of RPODM at each stage can be summarized as $\mathbf{x}^l = \mathcal{G}\left(\mathcal{F}\left(\mathcal{A}\left(\mathbf{x}^{l-1}; \boldsymbol{\theta}^l_{\mathcal{A}}\right); \boldsymbol{\theta}^l_{\mathbf{H}}\right)\right)$.

Experimental Results

- **Convergence Analysis:**



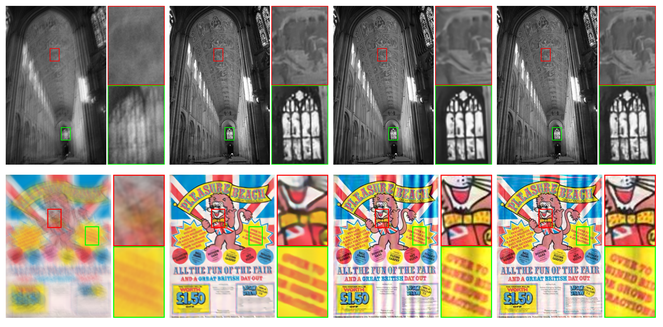
(a) (b) (c)

Figure 2: Convergence curves of FISTA, IRCNN, and our PODM.

- **Relaxed PODM for Image Restoration:**

Table 1: Averaged quantitative performance on image restoration.

	Metric	TV	HL	EPLL	CSF	RTF	IRCNN	Ours
Levin	PSNR	29.38	30.12	31.65	32.74	33.26	32.51	**34.06**
	SSIM	0.88	0.90	0.93	0.93	0.94	0.92	**0.97**
Sun	PSNR	30.67	31.03	32.44	31.55	32.45	32.61	**32.62**
	SSIM	0.85	0.85	0.88	0.87	**0.89**	**0.89**	**0.89**



Blurred Image RTF IRCNN Ours

Figure 3: Image restoration results on the real blurry image.

References

[a].Amir et al. A fast iterative shrinkage-thresholding algorithm for linear inverse problems.**SIAM JIS 2009.**

[b].Marcin et al. Learning to learn by gradient descent by gradient descent.**NIPS2016.**

[c].Liu et al. On the convergence of learning-based iterative methods for nonconvex inverse problems.**arXiv:1808.05331, 2018.**