On Training Implicit Models

Zhengyang Geng[1,2], Xin-Yu Zhang[2], Shaojie Bai[3], Yisen Wang[2], Zhouchen Lin[2,4]
[1]Zhejiang Lab, [2]Peking University, [3]Carnegie Mellon University, [4]Pazhou Lab

arXiv: https://arxiv.org/abs/2111.05177
Github: https://github.com/Gsunshine/phantom_grad
Email: zhengyanggeng@gamil.com

Background

- Implicitly-defined neural networks have achieved competitive performances compared with explicit models.
- Implicit models treat the evolution of hidden states as certain dynamics, *e.g.*, fixed-point equations or ordinary differential equations (ODEs);
- The forward passes are formulated as black-box solvers of the underlying dynamics, and the backward passes are performed via implicit differentiation.
- In this work, we argue that a carefully designed inexact gradient, named phantom gradient, is sufficient to efficiently and effectively train implicit models.

Implicit Differentiation

We adopt the formulation of DEQ models [1].

- Input projection module $\mathcal{M}$: $\boldsymbol{u} = \mathcal{M}(\boldsymbol{x})$, where $\boldsymbol{x}$ is the input data;
- Equilibrium module $\mathcal{F}$ and the equilibrium state $\boldsymbol{h}^*$ given by

$$\boldsymbol{h}^* = \mathcal{F}(\boldsymbol{h}^*, \boldsymbol{z}),$$

where $\boldsymbol{z}$ is a union of the module's input $\boldsymbol{u}$ and parameters $\boldsymbol{\theta}$;

- Post-processing module $\mathcal{G}$: $\hat{\boldsymbol{y}} = \mathcal{G}(\boldsymbol{h}^*)$, where $\hat{\boldsymbol{y}}$ is the predicted label of $\boldsymbol{x}$;
- Loss function $\mathcal{L}$ and the training objective, *i.e.*, the expected loss:

$$\mathcal{R}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{P}}[\mathcal{L}(\hat{\boldsymbol{y}}(\boldsymbol{x};\boldsymbol{\theta}), \boldsymbol{y})],$$

where $\boldsymbol{y}$ is the true label of $\boldsymbol{x}$.

- Using Implicit Differentiation, the gradient of $\boldsymbol{h}^*$ *w.r.t.* $\boldsymbol{z}$ is given by

$$\frac{\partial \boldsymbol{h}^*}{\partial \boldsymbol{z}} = \frac{\partial \mathcal{F}}{\partial \boldsymbol{z}}\bigg|_{\boldsymbol{h}^*}\left(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\bigg|_{\boldsymbol{h}^*}\right)^{-1}.$$

The gradient of $\mathcal{L}$ *w.r.t.* $\boldsymbol{z}$ is thus given by

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}} = \frac{\partial \boldsymbol{h}^*}{\partial \boldsymbol{z}}\frac{\partial \mathcal{L}}{\partial \boldsymbol{h}} = \frac{\partial \mathcal{F}}{\partial \boldsymbol{z}}\bigg|_{\boldsymbol{h}^*}\left(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\bigg|_{\boldsymbol{h}^*}\right)^{-1}\frac{\partial \mathcal{L}}{\partial \boldsymbol{h}^*}.$$

Future Perspectives

- The phantom gradient may come with a structured bias in comparison with the exact one; how to eliminate the structured bias?
- The UPG and its precision in the training process suggest developing an adaptive gradient solver.
- (Aggressive) The loss landscape and the training strategy are the two sides of the same coin; how to study their interaction in training implicit models?

Phantom Gradient

- **Definition**. The Jacobian $\partial \boldsymbol{h}^*/\partial \boldsymbol{\theta}$ is approximated by a matrix $\boldsymbol{A}$:

$$\frac{\widehat{\partial \mathcal{L}}}{\partial \boldsymbol{\theta}} \triangleq \boldsymbol{A}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}.$$

- **General Descent Condition.**

Theorem 1. Let $\sigma_{\max}$ and $\sigma_{\min}$ be the maximal and minimal singular value of $\partial\mathcal{F}/\partial\boldsymbol{\theta}$. If

$$\left\|\boldsymbol{A}\left(\boldsymbol{I} - \frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\right) - \frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\right\| \le \frac{\sigma_{\min}^2}{\sigma_{\max}},$$

then the phantom gradient provides an ascent direction of the function $\mathcal{F}$, *i.e.*,

$$\left\langle \frac{\widehat{\partial \mathcal{L}}}{\partial \boldsymbol{\theta}}, \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \right\rangle \ge 0.$$

- **Instantiations**.

a. **Unrolling-based Phantom Gradient** (**UPG**). Consider the damped fixed-point iteration:

$$\boldsymbol{h}_{t+1} = \lambda\mathcal{F}(\boldsymbol{h}_t, \boldsymbol{z}) + (1-\lambda)\boldsymbol{h}_t, \;\; t = 0,\, 1,\, \cdots,\, T-1.$$

Then, the matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A}_{k,\lambda}^{\text{unr}} = \lambda\sum_{t=0}^{k-1}\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{h}_t}\prod_{s=t+1}^{k-1}\left(\lambda\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\bigg|_{\boldsymbol{h}_s} + (1-\lambda)\boldsymbol{I}\right).$$

b. **Neumann-series-based Phantom Gradient** (**NPG**). The matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A}_{k,\lambda}^{\text{neu}} = \lambda\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{h}^*}\left(\boldsymbol{I} + \boldsymbol{B} + \boldsymbol{B}^2 + \cdots + \boldsymbol{B}^{k-1}\right), \;\text{ where } \boldsymbol{B} = \lambda\frac{\partial \mathcal{F}}{\partial \boldsymbol{h}}\bigg|_{\boldsymbol{h}^*} + (1-\lambda)\boldsymbol{I}.$$

- **Convergence Theory**.

Theorem 3. Suppose the loss function $\mathcal{R}$ is ℓ-smooth, lower-bounded, and has bounded gradient almost surely in the training process. Besides, assume the gradient $\partial\mathcal{L}/\partial\boldsymbol{\theta}$ is an unbiased estimator of $\nabla\mathcal{R}(\boldsymbol{\theta})$ with a bounded covariance. If the phantom gradient in is an ε-approximation to $\partial\mathcal{L}/\partial\boldsymbol{\theta}$, *i.e.*,

$$\left\|\frac{\widehat{\partial \mathcal{L}}}{\partial \boldsymbol{\theta}} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\right\| \le \varepsilon, \qquad \text{almost surely},$$

then using the phantom gradient as a stochastic first-order oracle with a step size of $\eta_n = O(1/\sqrt{n})$ to update $\boldsymbol{\theta}$ with gradient descent, it follows after N iterations that

$$\mathbb{E}\left[\frac{\sum_{n=1}^{N}\eta_n\|\nabla\mathcal{R}(\boldsymbol{\theta}_n)\|^2}{\sum_{n=1}^{N}\eta_n}\right] \le O\left(\varepsilon + \frac{\log N}{\sqrt{N}}\right).$$

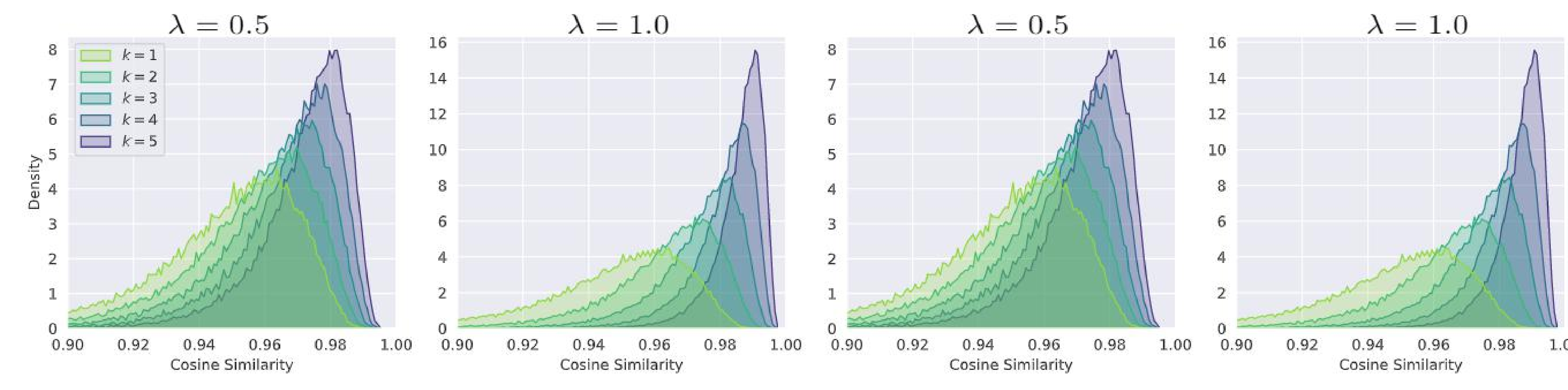
- **Complexity**.

Let **Mem** denote the memory cost, and K and k be the solver's steps and the unrolling/Neumann steps, respectively. Here, $K \gg k \approx 1$.

Method	Time	Mem	Peak Mem
Implicit	$\mathcal{O}(K)$	$\mathcal{O}(1)$	$\mathcal{O}(k)$
UPG	$\mathcal{O}(k)$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
NPG	$\mathcal{O}(k)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

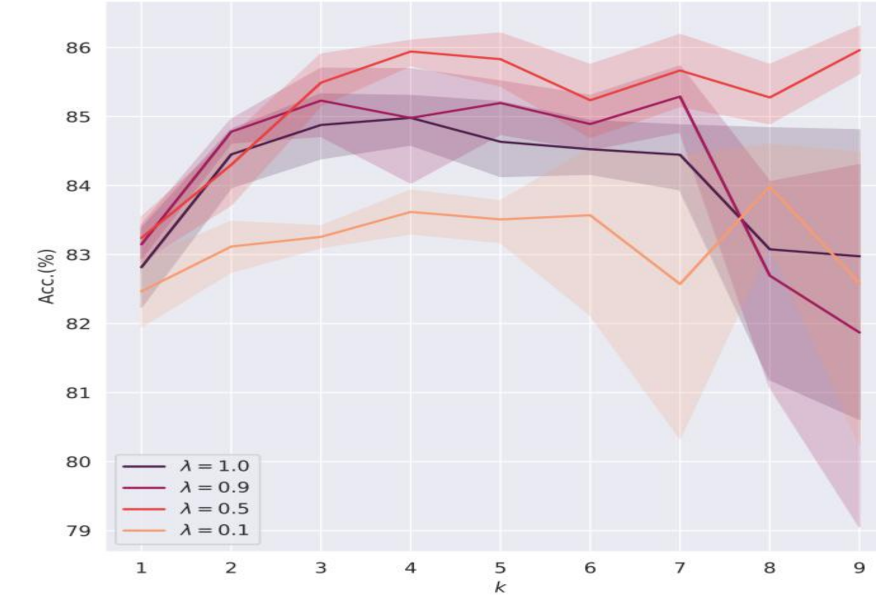
Experiments

- Cosine similarity between the phantom gradient and the exact gradient in the synthetic setting (see paper for details);



(a) Phantom gradient in the Neumann form (b) Phantom gradient in the unrolling form

- Impact of hyperparameters λ and k on the CIFAR-10 classification accuracy;



- Large-scale experiments;

Datasets	Model	Method	Params	Metrics	Speed
CIFAR-10	MDEQ	Implicit	10M	93.8 ± 0.17	1.0×
CIFAR-10	MDEQ	UPG $\boldsymbol{A}_{5,0.5}$	10M	95.0 ± 0.16	1.4×
ImageNet	MDEQ	Implicit	18M	75.3	1.0×
ImageNet	MDEQ	UPG $\boldsymbol{A}_{5,0.6}$	18M	75.7	1.7×
Wikitext-103	DEQ (PostLN)	Implicit	98M	24.0	1.0×
Wikitext-103	DEQ (PostLN)	UPG $\boldsymbol{A}_{5,0.8}$	98M	25.7	1.7×
Wikitext-103	DEQ (PreLN)	JR + Implicit	98M	24.5	1.7×
Wikitext-103	DEQ (PreLN)	JR + UPG $\boldsymbol{A}_{5,0.8}$	98M	24.4	2.2×
Wikitext-103	DEQ (PreLN)	JR + UPG $\boldsymbol{A}_{5,0.8}$	98M	$24.0^{\dagger}$	1.7×

- Implicit GNN [2] model on graph tasks.

Datasets	Model	Method	Params	Metrics (%)
COX2	IGNN	Implicit	38K	84.1 ± 2.9
COX2	IGNN	UPG $\boldsymbol{A}_{5,0.5}$	38K	83.9 ± 3.0
COX2	IGNN	UPG $\boldsymbol{A}_{5,0.8}$	38K	83.9 ± 2.7
COX2	IGNN	UPG $\boldsymbol{A}_{5,1.0}$	38K	83.0 ± 2.9
PROTEINS	IGNN	Implicit	34K	78.6 ± 4.1
PROTEINS	IGNN	UPG $\boldsymbol{A}_{5,0.5}$	34K	78.4 ± 4.2
PROTEINS	IGNN	UPG $\boldsymbol{A}_{5,0.8}$	34K	78.6 ± 4.2
PROTEINS	IGNN	UPG $\boldsymbol{A}_{5,1.0}$	34K	78.8 ± 4.2
PPI	IGNN	Implicit	4.7M	97.6
PPI	IGNN	UPG $\boldsymbol{A}_{5,0.5}$	4.7M	98.2
PPI	IGNN	UPG $\boldsymbol{A}_{5,0.8}$	4.7M	97.4
PPI	IGNN	UPG $\boldsymbol{A}_{5,1.0}$	4.7M	96.2

References

- [1] Shaojie Bai, J. Zico Kolter, Vladlen Koltun. Deep Equilibrium Models.
- [2] Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, Laurent El Ghaoui. Implicit Graph Neural Netowrks.