

Data-driven Inverse Modeling from Sparse Observations

Kailai Xu¹ and Eric Darve^{1, 2}

¹Institute for Computational and Mathematical Engineering

²Mechanical Engineering

Stanford University

Stanford, California 94305

Abstract

Deep neural networks (DNN) have been used to model nonlinear relations between physical quantities. Those DNNs are embedded in physical systems described by partial differential equations (PDE) and trained by minimizing a loss function that measures the discrepancy between predictions and observations in some chosen norm. This loss function often includes the PDE constraints as a penalty term when only sparse observations are available. As a result, the PDE is only satisfied approximately by the solution. However, the penalty term typically slows down the convergence of the optimizer for stiff problems. We present a new approach that trains the embedded DNNs while numerically satisfying the PDE constraints. We develop an algorithm that enables differentiating both explicit and implicit numerical solvers in reverse-mode automatic differentiation. This allows the gradients of the DNNs and the PDE solvers to be computed in a unified framework. We demonstrate that our approach enjoys faster convergence and better stability in relatively stiff problems compared to the penalty method. Our approach allows for the potential to solve and accelerate a wide range of data-driven inverse modeling, where the physical constraints are described by PDEs and need to be satisfied accurately.

Introduction: Data-driven Inverse Modeling with Neural Networks

Models involving partial differential equations (PDE) are usually used for describing physical phenomena in science and engineering. Unknown parameters in the models can be calibrated using observations, which are typically associated with the output of the models.

When the unknown is a function, an approach is to approximate the unknown using a neural network and plug it into the PDE. The neural network is trained by matching the predicted and the observed output of the PDE model. In the presence of full-field observations, in many cases we can approximate the derivatives in the PDE and reduce the inverse problem to a standard regression problem (see [1] for an example). However, in the context of sparse observations, i.e.,

only part of the outputs of the models are observable, we must couple the PDE and the neural network to obtain the prediction.

Specifically, we formulate the inverse problem as a PDE-constrained optimization problem

$$\begin{aligned} \min_{\theta \in \Theta} L(u) &= \sum_{i \in \mathcal{I}_{\text{obs}}} (u(\mathbf{x}_i) - u_i)^2 \\ \text{s.t. } F(\theta, u) &= 0 \end{aligned}$$

where L is called the *loss function*, which measures the discrepancy between estimated outputs u and observed outputs u_i at locations $\{\mathbf{x}_i\}$. \mathcal{I}_{obs} is the set of indices of locations where observations are available. F is the PDE model from which we can calculate the solution u . Θ is the space of all neural networks with a fixed architecture and θ can be viewed as weights and biases. Θ can also be physical parameter spaces when we solve a parametric inverse problem. One popular way to solve this problem is by minimizing the augmented loss function (penalty method) [2]

$$\min_{\theta, u} \tilde{L}(\theta, u) = L(u) + \lambda \|F(\theta, u)\|_2^2$$

However, this approach suffers from ill-conditioning and slow convergence partially due to the additional independent variable u besides θ .

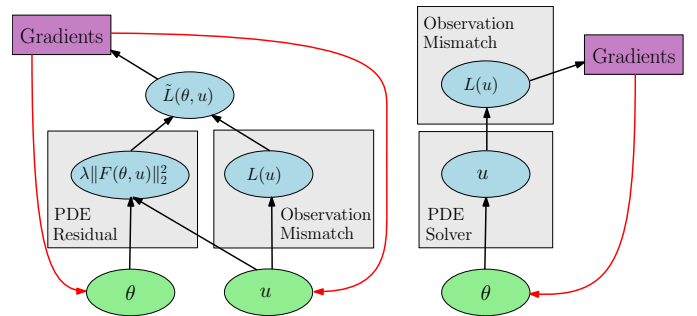


Figure 1: Comparison of the penalty method (left) and PCL (right).

In this work, we propose a new approach, physics constrained learning (PCL), that improves the conditioning and accelerates the convergence of inverse modeling. First, we

enforce the physical constraint $F(\theta, u) = 0$ by solving the PDE numerically. Our approach is compatible with common numerical schemes such as finite difference methods, finite volume methods, and finite element methods. Second, the gradient $\frac{\partial L(u(\theta))}{\partial \theta}$ needed for optimization is computed with reverse-mode automatic differentiation (AD) [3], and the required Jacobian is computed using forward-mode automatic differentiation. We use ADCME¹ for AD functionalities in this work.

Methods: Physics Constrained Learning

The main step in PCL is how to compute the gradients $\frac{\partial L(u(\theta))}{\partial \theta}$. PCL is based on the formula

$$\frac{\partial L(u(\theta))}{\partial \theta} = -\frac{\partial L(u)}{\partial u} \left(\frac{\partial F}{\partial u} \Big|_{u=u(\theta)} \right)^{-1} \frac{\partial F}{\partial \theta}(\theta, u(\theta))$$

The key for efficiency is to compute the gradients in the following three steps

1. The Jacobian $\frac{\partial F}{\partial u} \Big|_{u=u(\theta)}$ is computed with forward Jacobian propagation and will remain sparse as long as the numerical scheme we choose has local basis functions.
2. Solving the linear system

$$\left(\frac{\partial F}{\partial u} \Big|_{u=G(\theta)} \right)^T w = \left(\frac{\partial L(u)}{\partial u} \right)^T \quad (1)$$

3. Apply reverse mode automatic differentiation to compute

$$\frac{\partial L(u(\theta))}{\partial \theta} = w^T \frac{\partial F}{\partial \theta}(\theta, G(\theta)) \quad (2)$$

Here θ can be the neural network weights and biases and thus can be high dimensional. The challenge here is to compute the Jacobian matrix as well as the gradient Equation (2). The detailed algorithm and analysis is presented in [4].

Findings and Discussion: Enabling Faster and More Robust Convergence

The key finding from our work is that enforcing physical constraints leads to faster and more robust convergence compared to the penalty method for stiff problems. We conduct multiple numerical examples and show that in our benchmark problems,

1. PCL enjoys faster convergence with respect to the number of iterations to converge to a predetermined accuracy. Particularly, we observe a 10^4 times speed-up compared with the penalty method in the Helmholtz problem. We also prove a convergence result, which shows that for the chosen model problem, the condition number in the penalty method is much worse than that of PCL.
2. PCL exhibits mesh independent convergence, while the penalty method does not scale with respect to the number of iterations as well as PCL when we refine the mesh.

¹<https://github.com/kailaix/ADCME.jl>

3. PCL is more robust to noise and neural network architectures. The penalty method includes the solution u_h as independent variables to optimize, and the optimizer may converge to a nonphysical local minimum.

For theoretical analysis, we consider a model problem

$$\begin{aligned} \min_{\theta} \|u - u_0\|_2^2 \\ \text{s.t. } Au = \theta y \end{aligned}$$

where $u_0 = A^{-1}y$ so that the optimal $\theta = 1$; the corresponding penalty method solve a least square problem

$$\min_{\theta} \|\mathbf{A}\theta - \mathbf{y}\|_2^2 \quad \mathbf{A}_\lambda = \begin{bmatrix} I & 0 \\ \sqrt{\lambda}A & -\sqrt{\lambda}y \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} u_0 \\ 0 \end{bmatrix}$$

We have proved the following theorem

Theorem 0.1 *The condition number of \mathbf{A}_λ is*

$$\liminf_{\lambda \rightarrow \infty} \kappa(\mathbf{A}_\lambda) \geq \kappa(A)^2$$

and therefore the condition number of the unconstrained optimization problem from the penalty method is the square of that from PCL asymptotically.

Conclusions

We believe that enforcing physical constraints in ill-conditioned inverse problem is essential for developing robust and efficient algorithms. Particularly, when the unknowns are represented by neural networks, PCL demonstrates superior robustness and efficiency compared to the penalty method. Technically, the application of automatic differentiation gets rid of the challenging and time-consuming process of deriving and implementing gradients and Jacobians. Meanwhile, AD also allows for leveraging the computational graph optimization to improve the inverse modeling performance. One limitation of PCL is that the PDE must be solved for each gradient computation, which can be expensive in both memory and computational costs. This computational challenge can be alleviated by considering accelerating techniques such as reduced-order modeling.

References

- [1] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017.
- [2] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [4] Kailai Xu and Eric Darve. Physics constrained learning for data-driven inverse modeling from sparse observations. *arXiv preprint arXiv:2002.10521*, 2020.