LEARNING-BASED STRONG SOLUTIONS TO FORWARD AND INVERSE PROBLEMS IN PDES

Leah Bar & Nir Sochen
Department of Applied Mathematics
Tel Aviv University
Tel Aviv, 69978, Israel
barleah.libra@gmail.com
sochen@post.tau.ac.il

ABSTRACT

We introduce a novel neural network-based partial differential equations solver for forward and inverse problems. The solver is grid free, mesh free and shape free, and the solution is approximated by a neural network. We employ an unsupervised approach such that the input to the network is a points set in an arbitrary domain, and the output is the set of the corresponding function values. The network is trained to minimize deviations of the learned function from the *strong* PDE solution and satisfy the boundary conditions. The resulting solution in turn is an explicit smooth differentiable function with a known analytical form.

Unlike other numerical methods such as finite differences and finite elements, the derivatives of the desired function can be analytically calculated to any order. This framework therefore, enables the solution of high order non-linear PDEs. The proposed algorithm is a unified formulation of both forward and inverse problems where the optimized loss function consists of few elements: fidelity terms of L_2 and L_∞ norms that unlike previous methods promote a strong solution. Robust boundary conditions constraints and additional regularizers are included as well. This setting is flexible in the sense that regularizers can be tailored to specific problems. We demonstrate our method on several free shape 2D second order systems with application to Electrical Impedance Tomography (EIT).

1 INTRODUCTION

Partial differential equations are fundamental in science and mathematics with wide applications in medical imaging, signal processing, computer vision, remote sensing, electromagnetism, economics and more. Classical methods such as finite differences, finite volume and finite elements are numerical discretization-based methods where the domain is divided into a uniform grid or polygon mesh. The differential equation is then reduced to a system of algebraic equations. These methods may have some limitations: the solution is numeric and may suffer from high condition number, highly dependent on the discretization and even the second derivative is sensitive to noise. In the last few years, deep learning and neural network-based algorithms are extensively used in pattern recognition, image processing, computer vision and more.

Recently, the deep learning approach had been adopted to the field of PDEs as well by converting the problem into a machine learning one. In *Supervised learning*, the network maps an input to an output based on example input-output pairs. This strategy is used in inverse problems, where the input to the network is a set of observations/measurements (e.g. x-ray tomography, ultrasound) and the output is the set of parameters of interest (tissue density etc.) Feigin et al. (2018); Lucas et al. (2018); McCann et al. (2017); Seo et al. (2019). *Unsupervised learning* on the other hand is a self-learning mechanism where the natural structure present within a set of data points is inferred. Algorithms for forward and inverse problems in partial differential equations via unsupervised learning were recently introduced. The *indirect* approach utilizes a neural network as a component in the solution e.g. Li et al. (2018); Khoo & Ying (2018); Han et al. (2018). *Direct* algorithms solve the forward and inverse problem PDEs by directly approximating the solution with a deep neural network. The

network parameters are determined by the optimization of a cost function such that the optimal solution satisfies the PDE, boundary conditions and initial conditions Chiaramonte & Kiener (2017); Li et al. (2019); Sirignano & Spiliopoulos (2017); Raissi et al. (2017); Xu & Darve (2019).

In this work we address the forward and inverse PDE problems via a direct unsupervised method. Our key contributions are four fold: (1) inverse problems can be solved in the same framework as the forward problems. (2) In both forward and inverse problems we extend the standard L_2-based fidelity term in the cost function by adding L_∞-like norm. Moreover, (3) some regularization terms which impose a-priori knowledge on the solution can be easily incorporated. (4) Our construction exemplifies the ability to handle free-form domain in a mesh free manner. Point (1) is essential for full tomography-like solutions. The extension of the loss function by the L_∞-like norm is fundamental. The L_2 term, used in recent studies, aims only for weak solutions of the PDE since it minimizes the error in the average sense. The L_∞-like norm pushes towards a pointwise or strong solution since it handles possible outliers of the error. This is not a merely theoretical issue but strongly affects the quality of the result as we empirically demonstrate. In unsupervised learning of ill-posed problems regularization is crucial. Choosing the right regularizer and the ability to incorporate it in the formulation is of prime importance. Our formalism integrates such regularizations in a natural way. We demonstrate our algorithm by a second order elliptic equation, in particular the Electrical Impedance Tomography (EIT) application on a circular and three other arbitrary domains.

2 Mathematical Formulation

Let Ω be a bounded open and connected subset of $\mathbb{R}^d$ where d is the dimension. A differential operator $\mathcal{L}$ acting on a function $u(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\mathcal{L}u(\mathbf{x}) := \Big(a_n(\mathbf{x}) \cdot \mathcal{D}^n + a_{n-1}(\mathbf{x}) \cdot \mathcal{D}^{n-1} + \ldots + a_1(\mathbf{x}) \cdot \mathcal{D} + a_0(\mathbf{x})\Big)u(\mathbf{x}), \tag{1}$$

where $\mathcal{D}^n$ is the nth order d-dimensional derivative and $a_0(\mathbf{x}), \ldots, a_n(\mathbf{x})$ are the coefficients. Consider the partial differential problem with Dirichlet boundary conditions

$$\begin{aligned} \mathcal{L}u(\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d \\ u(\mathbf{x}) &= u_0(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega, \end{aligned} \tag{2}$$

where $f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is a given function. The *forward problem* solves $u(\mathbf{x})$ given the coefficients $\theta := \{a_0(\mathbf{x}), \ldots, a_n(\mathbf{x})\}$ while the *inverse problem* determines the coefficients set θ given $u(\mathbf{x})$.

The proposed algorithm approximates the solutions in both problems by neural networks $u(\mathbf{x}; w_u)$, $\{a_j(\mathbf{x}; w_{a_j})\}$ such that the networks are parameterized by $w_u, \{w_{a_j}\}$, and the input to the network is $\mathbf{x} \in \mathbb{R}^d$. The network consists of few fully connected layers with *tanh* activation and linear sum in the last layer. The network is trained to satisfy the PDE with boundary conditions by minimizing a cost function. In the forward problem

$$\mathcal{F}(u) = \lambda\|\mathcal{L}u - f\|_2^2 + \mu\|\mathcal{L}u - f\|_\infty + \|u - u_0\|_{1,\partial\Omega} + \mathcal{R}^F(u), \tag{3}$$

and in the inverse problem

$$\mathcal{I}(\theta) = \lambda\|\mathcal{L}u - f\|_2^2 + \mu\|\mathcal{L}u - f\|_\infty + \|\theta - \theta_0\|_{1,\partial\Omega} + \mathcal{R}^I(\theta). \tag{4}$$

The first two terms enforce the solution to satisfy the equation. The first term minimizes the error in the L_2 sense while the second term minimizes the maximal error.

Ideally, the solution to the PDE has to be satisfied at every point $\mathbf{x}$. This is considered as a *strong* solution. In the deep learning framework the optimization is carried out on mini batches of points. The first term minimizes the equation in the integral or average sense and is therefore insensitive to a point jump in the value of the integrand. It therefore yields a *weak* solution. The L_∞ term promotes a strong solution since it handles possible discontinuous points in $\mathcal{L}u - f$. Practically, we use a relaxed version of the L_∞ norm where we take the top-K errors. Then, at every mini batch the training procedure minimizes the equation at K different points having the largest errors. The third term imposes boundary conditions where u_0 and θ_0 are the boundary values of $u(\mathbf{x})$ and $\theta(\mathbf{x})$. We use the L_1 norm on the boundary so that the solution is robust to measurements noise. The last term is a regularizer which can be tailored to the application. There are few advantages

of this setting. First, the solution is a network and therefore an explicit function. It is a smooth analytic function and is therefore *analytically differentiable*. In addition, the proposed framework enables setting a prior physical knowledge on the solution by designing the regularizers $\mathcal{R}^F$ and $\mathcal{R}^I$. Moreover, the training procedure is mesh free. Unlike finite differences or finite elements methods, we use *random* points in the domain and its boundary in the course of the optimization of equation 3 and equation 4. This means that the solution does not depend upon a coordinate mesh and we can also define an arbitrary regular domain Ω.

3 Application to Electrical Impedance Tomography

Consider the following elliptic equation which is a special case of equation 1,

$$\begin{aligned} &\nabla \cdot \Big(\sigma(\mathbf{x})\nabla u(\mathbf{x})\Big) = 0, \qquad \mathbf{x} \in \Omega \subset \mathbb{R}^2 \\ &u(\mathbf{x}) = u_0(\mathbf{x}), \qquad\qquad\quad \mathbf{x} \in \partial\Omega, \end{aligned} \tag{5}$$

where $\nabla := (\partial/\partial_x, \partial/\partial_y)$. We assume that $0 < \sigma(\mathbf{x}) \in C^1(\Omega)$, which guarantees existence and uniqueness of a solution $u \in C^2(\Omega)$ Evans (2010).

The elliptical system equation 5 was addressed by Siltanen et al. (2000) in the context of Electrical Impedance Tomography (EIT) which is a reconstruction method for the inverse conductivity problem. The function σ indicates the electrical conductivity density, and u is the electrical potential. An electrical current

$$\psi_{n,\varphi} = \sigma \frac{\partial u_n}{\partial \nu}\Big|_{\partial\Omega} = \frac{1}{\sqrt{2\pi}} \cos(n\kappa + \varphi), \;\; n \in \mathbb{Z} \tag{6}$$

is applied on electrodes on the surface $\partial\Omega$, where κ is the angle in polar coordinate system along the domain boundary, n is the current frequency, φ is the phase and ν is the normal unit. The resulting voltage $u|_{\partial\Omega} = u_0$ is measured through the electrodes. The conductivity σ is determined from the knowledge of the Dirichlet-to-Neumann map or voltage-to-current map

$$\Lambda_\gamma : u|_{\partial\Omega} \to \sigma \frac{\partial u_n}{\partial \nu}\Big|_{\partial_\Omega},$$

Mueller & Siltanen (2012); Alsaker & Mueller (2018); Fan & Ying (2019).

We demonstrate our framework by solving the forward and inverse problem of equation 5 which is a first step towards a full tomography. Following Mueller & Siltanen (2012), we simulate the voltage measurement $u|_{\partial\Omega}$ by the Finite Elements Method (FEM) given three variants of a conductivity phantom $\sigma(\mathbf{x})$. We calculate the FEM solution with different triangle mesh densities and select as ground truth the one such that finer meshes do not improve the numerical solution. With our suggested method, the forward problem determines the electrical potential u in the whole domain Ω given σ, while the inverse problem uses the approximated u and calculates the conductivity σ given that $\sigma|_{\partial\Omega} = \sigma_0$.

3.1 Forward Problem

In the forward problem the conductivity $\sigma(x_i)$ and boundary conditions $u_0(x_b)$ are given for random points set $\{x_i\} \in \Omega \subset \mathbb{R}^2$, $\{x_b\} \in \partial\Omega \subset \mathbb{R}^2$ with sets size of N_s and N_b respectively. A neural network approximates $u(\mathbf{x})$. Let

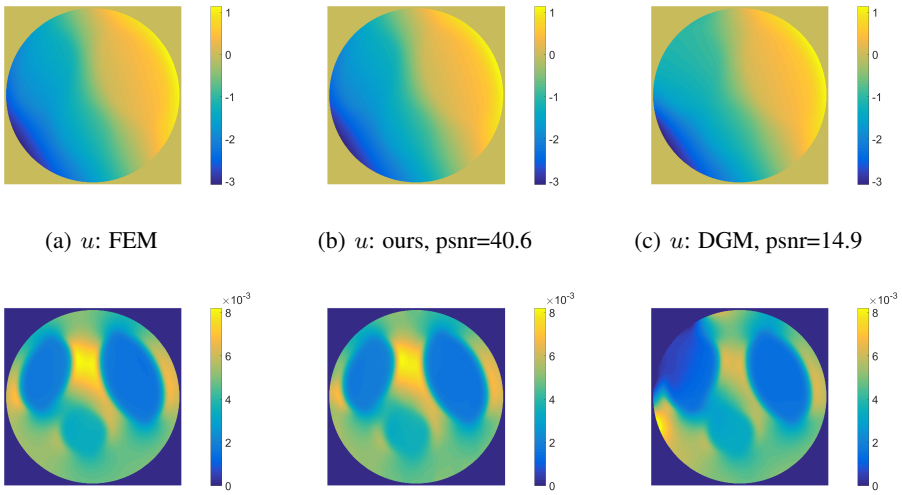
$$\mathcal{L}_i := \nabla \cdot \Big(\sigma(x_i)\nabla u(x_i)\Big). \tag{7}$$

The cost function equation 3 is then rewritten as

$$\begin{aligned} \mathcal{F}\Big(u(x; w_u)\Big) = &\frac{\lambda}{N_s} \sum_{i=1}^{N_s} |\mathcal{L}_i|^2 + \frac{\mu}{K} \sum_{k \in \text{top}_K(|\mathcal{L}_i|)} |\mathcal{L}_k| \\ &+ \frac{1}{N_b} \sum_{b=1}^{N_b} \Big|u(x_b) - u_0(x_b)\Big| + \alpha \|w_u\|_2^2. \end{aligned} \tag{8}$$

The first term is the L_2 norm of the differential operator, the second term is a relaxed version of the infinity norm where we take the mean value of the top-K values of $|\mathcal{L}_i|$. The third term imposes the boundary conditions and the last term serves as a regularizer of the network parameters.

Figure 1 shows the forward problem results for current ψ with $n = 1$ and $\varphi = \pi/8$. The left column is the FEM solution which is referred to as ground truth, where the top row indicates the solution $u(\mathbf{x})$ and the bottom row the derivative of $u(\mathbf{x})$ with respect to the first coordinate x calculated as the finite difference approximation of the FEM result. The middle column depicts the outcome of the proposed method where $\partial u/\partial x$ is an analytical derivative of our result. The right column shows the outcome of the DGM method Sirignano & Spiliopoulos (2017) which is a special case of equation 8 with $\lambda = 1$, $\mu = 0$, and $\alpha = 0$. Clearly, the proposed method outperforms the DGM method since the weighting parameters, the relaxed L_∞ norm and network weights regularization play a significant role in the loss function.



(a) u: FEM (b) u: ours, psnr=40.6 (c) u: DGM, psnr=14.9

(d) u_x: FEM (e) u_x: ours, psnr=40.55 (f) u_x: DGM, psnr=19.27

Figure 1: Forward problem results of $u(\mathbf{x})$ and $u_x := \partial u(\mathbf{x})/\partial x$ for current frequency $n = 1$ and phase $\varphi = \pi/8$. Left column: ground truth (FEM). Middle column: proposed method. Right column: DGM method.

3.2 Generalized Inverse Problem

In the inverse problem, the electrical potential $u(\mathbf{x})$ is known while $\sigma(\mathbf{x})$ is unknown. Since we have a network which approximates $u(\mathbf{x})$, we can evaluate it at any point $\mathbf{x}$. A physical a priori knowledge regarding σ is exploited in this application. The conductivity is assumed to have well defined sub-regions. We therefore design the regularization term to have sparse edges via total variation ($p = 1$ in the last term of equation 9). The objective function (equation 4) then takes the form

$$\begin{aligned}\mathcal{I}\Big(\sigma(\mathbf{x}; w_\sigma)\Big) =& \frac{\lambda}{N_s}\sum_{i=1}^{N_s}|\mathcal{L}_i|^2 + \frac{\mu}{K}\sum_{k\in \text{top}_K(|\mathcal{L}_i|)}|\mathcal{L}_k| \\ &+ \frac{1}{N_b}\sum_{b=1}^{N_b}\Big|\sigma(x_b) - \sigma_0(x_b)\Big| + \alpha\|w_\sigma\|_2^2 + \frac{\beta}{N_s}\sum_{i=1}^{N_s}|\nabla\sigma(x_i)|^p.\end{aligned} \tag{9}$$

Additional generalization exploits *multiple* voltage functions associated to a currents set $\{\psi_j\}$. The σ calculation thus, simultaneously relies on all $\{u_j\}$, resulting a more stable solution. Let

$$\mathcal{L}_{ij} := \nabla \cdot \Big(\sigma(x_i)\nabla u_j(x_i)\Big), \tag{10}$$

then equation 9 is generalized to

$$\mathcal{J}_j\big(\sigma(x;w_\sigma)\big) = \frac{\lambda}{N_s}\sum_{i=1}^{N_s}|\mathcal{L}_{ij}|^2 + \frac{\mu}{K}\sum_{k\in \text{top}_K(|\mathcal{L}_{ij}|)}|\mathcal{L}_{kj}|, \tag{11}$$

and

$$\begin{aligned}\mathcal{I}\Big(\sigma(x;w_\sigma\Big) = \sum_j \mathcal{J}_j\Big(\sigma(x;w_\sigma)\Big) + \frac{1}{N_b}\sum_{b=1}^{N_b}\Big|\sigma(x_b) - \sigma_0(x_b)\Big| \\ + \alpha\|w_\sigma\|_2^2 + \frac{\beta}{N_s}\sum_{i=1}^{N_s}|\nabla\sigma(x_i)|^p.\end{aligned} \tag{12}$$

The results are shown in Figure 2. In both phantoms we used four current combinations $\psi_{n,\varphi}$ where $\{n,\varphi\}$= $(1,0)$, $(1,\pi/4)$, $(2,0)$, $(2,\pi/4)$, and $\sigma_0 = 1$.

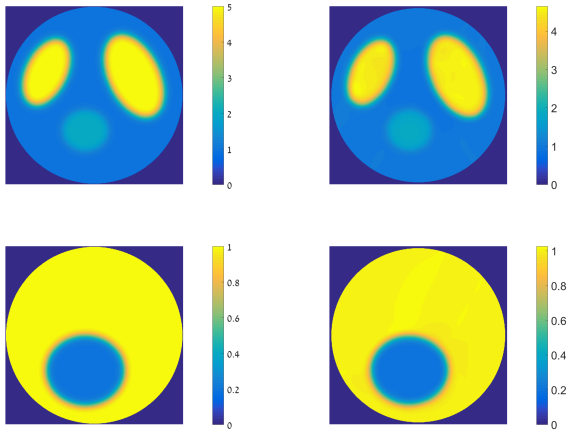


Figure 2: Generalized inverse problem results. Top left: ground truth of phantom 1. Top right: reconstructed $\sigma(\mathbf{x})$, PSNR=27.9. Bottom left: ground truth of phantom 2. Bottom right: reconstructed $\sigma(\mathbf{x})$, PSNR=40.44.

Next, we applied the proposed method to three arbitrary domains. The random sample points within the domain and along its boundary can be easily obtained. The outcome of the generalized inverse problem is shown in Figure 3

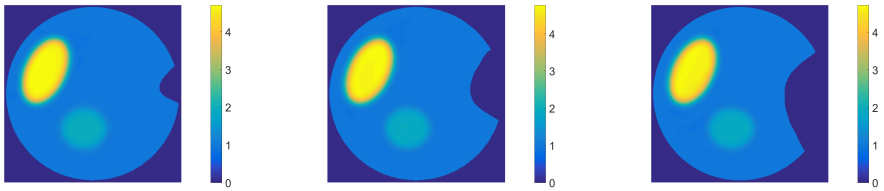


Figure 3: Reconstructed $\sigma(\mathbf{x})$ applied on different domains. PSNR=(33.67, 36.27, 33.51).

REFERENCES

M. Alsaker and J. L. Mueller. Use of an optimized spatial prior in D-bar reconstructions of EIT tank data. *Inverse Problems and Imaging*, 12:883–901, 2018.

M. M. Chiaramonte and M. Kiener. *Solving differential equations using neural networks*, 2017. `http://cs229.stanford.edu/proj2013`.

L.C. Evans. *Patial Differential Equations*. the American Mathematical Society, 2010.

Y. Fan and L. Ying. Solving electrical impedance tomography with deep learning. *arXiv 1906.03944v1*, 2019.

M. Feigin, D. Freedman, and B. W. Anthony. A deep learning framework for single-sided sound speed inversion in medical ultrasound. *arXiv 1810.00322v3*, 2018.

J. Han, A. Jentzen, and E. Weinan. Solving high-dimensional partial differential equations using deep learning. *arXiv 1707.02568*, 2018.

Y. Khoo and L. Ying. Switchnet: A neural network model for forward and inverse scattering problems. *arXiv 1810.09676v1*, 2018.

H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. NETT: Solving inverse problems with deep neural networks. *arXiv 1803.00092*, 2018.

Y. Li, J. Lu, and A. Mao. Variational training of neural network approximations of solution maps for physical models. *arXiv 1905.02789v1*, 2019.

A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging. *IEEE Signal Processing Magazine*, 2018.

M. T. McCann, K. H. Jin, and M. Unser. A review of convolutional neural networks for inverse problems in imaging. *arXiv 1710.04011*, 2017.

J. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, 2012.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv 1711.10561v1*, 2017.

J.K. Seo, K. C. Kim, A. Jargal, K. Lee, and B. Harrach. A learning-based method for solving ill-posed nonlinear inverse problems: A simulation study of lung EIT. *SIAM journal on Imaging Sciences*, 12:1275–1295, 2019.

S. Siltanen, J. Mueller, and D. Isaacson. An implementation of the reconstruction algorithm of a nachman for the 2d inverse conductivity problem. *Inverse Problems*, 16(3):681–699, 2000.

J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *arXiv 1708.07469v3*, 2017.

K. Xu and E. Darve. The neural network approach to inverse problems in differential equations. *arXiv 1901.07758v1*, 2019.