New deep neural networks solving non-linear inverse problems

or

How to use uniqueness results for inverse problems to design neural networks?

Matti Lassas
and
Christopher Wong
Maarten de Hoop

University of Helsinki

Slides available at
www.mv.helsinki.fi/home/lassas/

Inverse problem in a d-dimensional body

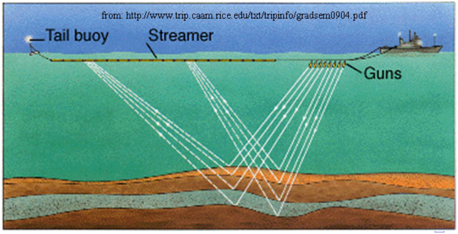
Let $u(x,t) = u^h(x,t)$ solve the wave equation

$$(\partial_t^2 - c(x)^2\Delta)u(x,t) = 0 \quad \text{on } (x,t) \in M \times \mathbb{R}_+,$$
$$\partial_\nu u(x,t)|_{\partial M \times \mathbb{R}_+} = h(x,t), \quad u|_{t=0} = 0, \quad \partial_t u|_{t=0} = 0,$$

where h is boundary source, $M \subset \mathbb{R}^d$. The Neumann-to-Dirichlet map is

$$Y_c h = u^h(x,t)\big|_{(x,t)\in\partial M \times \mathbb{R}_+}.$$

In the inverse problem we aim to find the unknown wave speeds $c(x)$ from boundary measurements Y_c (Traditionally, one denotes $Y_c = \Lambda_c$).
Next we consider this problem in the 1-dimensional case and solve it using neural networks.



Results on the hyperbolic inverse problem:

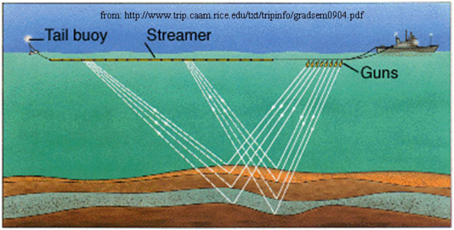
- 1-dimensional problems: Gelfand, Levitan, Marchenko 1950-1960.
- Inverse problem for $\Delta + q$: Nachman-Sylvester-Uhlmann 1988.
- Reconstruction of a Riemannian manifold with time-indepedent metric: Belishev-Kurylev 1992 and Tataru 1995.
- Solution by modified time reversal and focusing of waves: Bingham-Kurylev-L.-Siltanen 2008.
- Combining several measurements for together for the wave equation: Helin-L.-Oksanen 2012.
- Numerical methods for focusing of waves: de Hoop-Kepley-Oksanen 2018.
- Partial data: L.-Oksanen 2014, Mansouri-Milne 2017.
- Inverse problems for the connection Laplacian: Kurylev-Oksanen-Paternain 2018.
- Scattering control: Caday-de Hoop-Katsnelson-Uhlmann 2018.

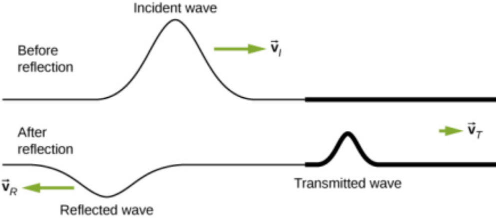
Overview of the talk

1. We consider the solution map $S : Y_c \to c$ that solves the inverse problem in the 1-dimensional case.
   For this, we use the boundary control method (Belishev 1987, Belishev-Kurylev 1992) and its regularized version (Bingham-Kurylev-L.-Siltanen 2008 and Korpela-L.-Oksanen 2018).
2. We propose an architecture of neural networks, where the input is a linear operator Y.
3. We show that the solution map S can be written as a neural network with the proposed architecture.
4. The performance of the trained neural network can be estimated using stability theorems for inverse problems.

Outline:

- Solution of the inverse problem in 1-dimensional space
- Standard neural networks
- Operator recurrent networks





Inverse problem in 1-dimensional space

Consider the wave equation in one-dimensional space, $x \in \mathbb{R}_+$.
This corresponds to subsurface imaging when the wave speed depends only on the depth.
Let $u(x,t)$ be the solution of the wave equation

$$(\frac{\partial^2}{\partial t^2} - c(x)^2 \frac{\partial^2}{\partial x^2}) u(x,t) = 0, \quad x \in \mathbb{R}_+, \ t \in \mathbb{R}_+$$
$$\frac{\partial}{\partial x} u|_{x=0} = h(t), \quad u|_{t=0} = 0, \quad \frac{\partial}{\partial t} u|_{t=0} = 0,$$

where the wave speed $c(x)$ is unknown. Denote $u(x,t) = u^h(x,t)$.
Let $T > 0$. Suppose we are given the Neumann-to-Dirichlet map, $Y = Y_c$,

$$Y_c h = u^h(x,t)\Big|_{x=0}, \quad t \in (0, 2T).$$

Y_c is a linear operator or "a matrix". Physically,

$$Y_c : \ \text{boundary source } h \ \rightarrow \text{the boundary value of the wave } u|_{x=0}.$$

Travel time function

The travel time for the wave from the boundary point 0 to the point x is

$$\tau(x) = \int_0^x \frac{1}{c(x')}\, dx'.$$

Assume that we can construct the function $\tau^{-1} : \mathbb{R}_+ \to \mathbb{R}_+$. Then we can determine the travel time function $\tau : \mathbb{R}_+ \to \mathbb{R}_+$ and the wave speed by

$$c(x) = \frac{1}{\frac{d}{dx}\tau(x)}.$$

Next, we study the inverse problem of finding the inverse travel time function τ^{-1} when Y_c is given.
We will consider the function $F : Y_c \to \tau^{-1}$ and construct a neural network that approximates F.

Neumann-to-Dirichlet map determines inner products of waves

Denote

$$\begin{aligned}\langle u^f(T), u^h(T)\rangle &= \int_{\mathbb{R}_+} u^f(x,T)u^h(x,T)dV(x), \quad dV = \frac{1}{c(x)^2}dx,\\ \|u^f(T)\|_{L^2(M)} &= \langle u^f(T), u^f(T)\rangle^{\frac{1}{2}}.\end{aligned}$$

By Blagovestchenskii formula,

$$\langle u^f(T), u^h(T)\rangle = \int_0^{2T} (K_Y f)(t)h(t)\,dt, \quad \langle u^f(T), 1\rangle = \int_0^T f(t)(T-t)\,dt$$

where $Y = Y_c$ is the Neumann-to-Dirichlet map,

$$\begin{aligned}&K_Y = JY - RYRJ,\\ &Rf(t) = f(2T-t) \quad \text{"time reversal operator"},\\ &Jf(t) = \tfrac{1}{2}1_{[0,T]}(t)\int_t^{2T-t} f(s)ds \quad \text{"low pass filter"}.\end{aligned}$$

An analytic solution algorithm for the inverse problem

By Bingham-Kurylev-L.-Siltanen 2008 and dH-L-W 2020, the inverse problem is solved as follows: Suppose we are given $Y = Y_c$.

Step 1: For the depth parameter $0 \le s \le T$, let $h_{\beta,s} \in L^2(0,2T)$ solve

$$\min_h \|u^h(T) - 1\|_{L^2}^2 + \beta\|Ah\|_{\ell^1} = \langle K_Y h, h\rangle - 2\langle h, b\rangle + C + \beta\|Ah\|_{\ell^1},$$

where $supp(h) \subset [T-s, T]$.
Here, $A : L^2(0,2T) \to \ell^2$ is an isometry and $K_Y = JY - RYRJ$. Then,

$$\lim_{\beta\to 0} u^{h_{\beta,s}}(x,T) = \begin{cases} 1, & \text{if } \tau(x) \le s \\ 0, & \text{otherwise.} \end{cases} .$$

We call $h_{\beta,s}$ the optimized sources.

Thus, when β is small,

$$u^{h_{\beta,s}}(x,T) \approx \begin{cases} 1, & \text{if } \tau(x) \le s \\ 0, & \text{otherwise.} \end{cases}$$

An analytic solution algorithm for the inverse problem

Step 2. Using the the optimized sources $h_{\beta,s}$, we compute

$$V(s) = \lim_{\beta\to 0} \int_0^T h_{\beta,s}(t)\,(T-t)dt = \lim_{\beta\to 0} \langle u^{h_{\beta,s}}(T), 1\rangle_{L^2(M)}$$

$$= \mathrm{vol}_c([0,\tau^{-1}(s)]) = \int_0^{\tau^{-1}(s)} \frac{1}{c(x)^2}\,dx,$$

$$w(s) = \frac{\partial}{\partial s} V(s).$$

Then

$$\tau^{-1}(s) = \int_0^s \frac{1}{w(t)}\,dt, \quad \text{and } c(\tau^{-1}(s)) = \frac{1}{w(s)}.$$

An analytic solution algorithm for the inverse problem

The above minimization problem can be solved using an iteration. Writing sources in a finite basis, the inverse problem is solved as follows:

Step 1: For $j = 1, \dots, K$ and $h^{(j)} = h_L^{(j)}$ be computed by doing L steps of the iterated soft thresholding,

$$h_{\ell+1}^{(j)} \quad := \quad \sigma_\beta\Big((I + P_j R Y R J - P_j J Y) h_\ell^{(j)} + P_j b\Big), \quad h_0^{(j)} = 0.$$

Here, $\beta > 0$ is the regularization parameter and

> R is the matrix of the time-reversal operator, P_j is a projector,
> J is the matrix of the low-pass filter, b is a constant vector,

$\sigma_\beta(x) = \text{relu}(x - \beta) - \text{relu}(-x - \beta)$ is soft thresholding, $\text{relu}(x) = \max(x, 0)$.

Step 2. Compute $\tau^{-1}(s_j) \approx G_j(h^{(1)}, \dots, h^{(K)})$, where $s_j = \frac{jT}{K}$ and G_j are explicit functions.

Summary on the analytic solution of the inverse problem

Consider the map $F : Y_c \to \tau^{-1}$ that determines the inverse of the travel time function τ^{-1} (and the wave speed $c(x)$) from the boundary measurements Y_c.

The discretized version of this map, $F : \mathbb{R}^{n\times n} \to \mathbb{R}^{2K}$ can be written as

$$F(Y_c) = G(f^{(1)}(Y_c), f^{(2)}(Y_c), \dots, f^{(K)}(Y_c))$$

where $f^{(j)} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n}$ map Y_c to the optimized sources,

$$f^{(j)}(Y_c) = h^{(j)}.$$

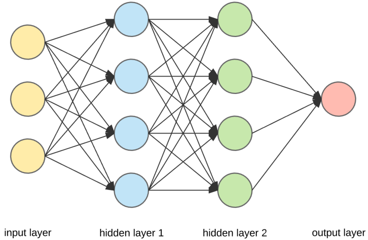
Next we define a family of neural networks (operator recurrent networks) than can approximate functions $f^{(j)} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n}$.

The explicit function G can be approximated by a standard neural network.

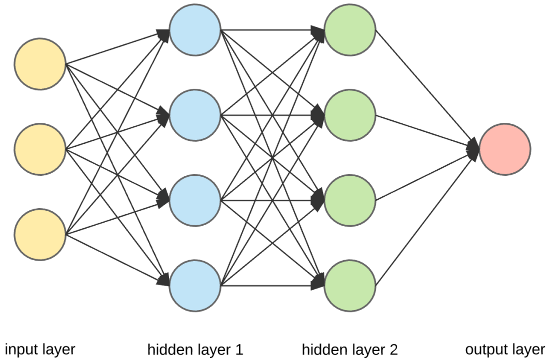
Then, we can approximate F by a neural network.

Outline:

- Solution of the inverse problem in 1-dimensional space
- Standard neural networks
- Operator recurrent networks



Standard neural network



- In every node in the hidden layers, one operates with a non-linear activation function ϕ. In this talk, ϕ is the Rectified Linear Unit,

$$\phi(x) = \mathrm{relu}(x) := \max(0, x) = \begin{cases} x, & x > 0, \\ 0, & x \leq 0, \end{cases} \qquad x \in \mathbb{R}.$$

Definition of the standard deep neural network

A standard neural network is a function $f_\theta : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ defined by

$$
\begin{aligned}
h_0 &= x, \\
h_{\ell+1} &= \phi\left(A_\theta^\ell h_\ell + b_\theta^\ell\right), \quad \ell = 0, \dots, L-1, \\
f_\theta(x) &= h_L.
\end{aligned}
$$

Architecture:

- ℓ: the layer index, max depth L.
- h_ℓ: intermediate output at layer ℓ.
- $b_\theta^\ell \in \mathbb{R}^{d_{\ell+1}}$, $A_\theta^\ell \in \mathbb{R}^{d_{\ell+1} \times d_\ell}$ are the biases and weight matrixes that depend on parameters $\theta = (\theta_1, \theta_2, \dots, \theta_m)$.
- ϕ is the activation function, the Rectified Linear Unit (relu)

$$
\phi : \mathbb{R}^d \to \mathbb{R}^d, \quad \phi(x_1, \dots, x_d) = (\max(0, x_1), \dots, \max(0, x_d))
$$

Applications of neural networks in inverse problems

- Modified gradient descent: Adler-Öktem 2017.
- Splines and Neural networks: Unser-Fageot-Ward (SIAM Rev. 2017), Jin-McCann-Froustey-Unser 2017.
- Data driven models: Arridge-Maass-Öktem-Schönlieb (Acta Numerica 2019)
- Generative adversarial networks: Bora-Jalal-Price-Dimakis 2017, Lunz-Öktem-Schönlieb 2018.
- Neumann Networks: Gilton-Ongie-Willett 2019.
- Diffusion problems: Arridge-Hauptmann 2019, Antholzer-Haltmeier-Schwab 2019, Agnelli-Col-L.-Murthy-Santacesaria-Siltanen 2020.
- Limited angle tomograpy: Bubba-Kutyniok-L.-Marz-Samek-Siltanen-Srinivasan 2019.
- Scattering problems: Uhlmann-Wang 2018, Khoo-Ying 2019, Li-Wang-Teixeira-Liu-Nehorai-Cui 2019, Wei-Chen 2019.

A modification of a neural network

Recall: A standard deep neural network is a function $f_\theta : \mathbb{R}^{d_0} \to \mathbb{R}^{d_L}$ that takes in a vector $x \in \mathbb{R}^{d_0}$ and computes following operations

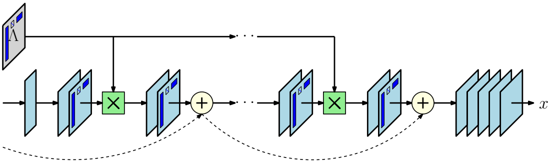
$$\begin{aligned} h_0 &= x, \\ h_{\ell+1} &= \phi\left(A_\theta^\ell h_\ell + b_\theta^\ell\right), \quad \ell = 0, \dots, L-1, \\ f_\theta(x) &= h_L. \end{aligned}$$

We will modify this: We define a function $f_\theta : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ that takes in a linear operator $Y \in \mathbb{R}^{n\times n}$ and computes following operations

$$\begin{aligned} h_0 &= b^0, \\ h_{\ell+1} &= \phi\left(A_\theta^\ell Y h_\ell + b_\theta^\ell\right), \quad \ell = 0, \dots, L-1, \\ f_\theta(Y) &= h_L. \end{aligned}$$

Outline:

- Solution of the inverse problem in 1-dimensional space
- Standard neural networks
- Operator recurrent networks

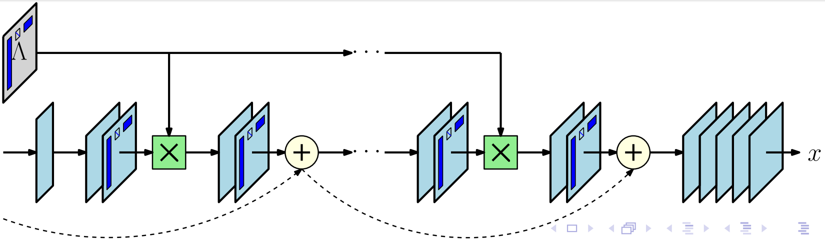


Definition

An *operator recurrent network* with depth L, width n and parameters $\theta \in [-1,1]^D \subset \mathbb{R}^D$ is a function $f_\theta : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ given by

$$
\begin{aligned}
h_0 &= b_\theta^{0,1},\\
h_\ell &= b_\theta^{\ell,1} + A_\theta^{\ell,1} h_{\ell-1} + A_\theta^{\ell,2} Y h_{\ell-1} + \phi\left[b_\theta^{\ell,2} + A_\theta^{\ell,3} h_{\ell-1} + A_\theta^{\ell,4} Y h_{\ell-1}\right],\\
f_\theta(Y) &= h_L,
\end{aligned}
$$

where the initial vector $h_0 = b_\theta^{0,1} \in \mathbb{R}^n$ is independent of the input $Y \in \mathbb{R}^{n\times n}$ and $A_\theta^{\ell,i} \in \mathbb{R}^{n\times n}$, $b_\theta^{\ell,i} \in \mathbb{R}^n$.
Activation functions ϕ are *relu* functions.



Iteration in the analytic algorithm is a neural network

Recall the earlier: The optimized sources were computed by doing L steps of the iterated soft thresholding,

$$h_{\ell+1}^{(j)} \;:=\; \sigma_\beta\Big((I + P_j R Y R J - P_j J Y) h_\ell^{(j)} + P_j b\Big), \quad h_0^{(j)} = 0.$$

Here, $\beta > 0$ and

R is the matrix of the time-reversal operator, P_j is a projector,
J is the matrix of the low-pass filter, b is a constant vector,

This iteration can be written as an operator recurrent network by using matrixes of operators.

Parametrization of the weight matrixes in the network

The weight matrixes $A_\theta^{\ell,i} \in \mathbb{R}^{n\times n}$ have the form

$$A_\theta^{\ell,i} = A^{\ell,i,(0)} + A_\theta^{\ell,i,(1)}, \quad A_\theta^{\ell,k,i,(1)} = \sum_{p=1}^{n} \theta_{2p-1}^{\ell,i}(\theta_{2p}^{\ell,i})^T,$$

where $A^{\ell,i,(0)}$ are fixed matrixes that do not depend on θ,
$A_\theta^{\ell,i,(1)}$ are sparse matrixes that are determined by parameters $\theta_p^{\ell,i} \in \mathbb{R}^n$.

The above iterated soft thresholding can be written as an operator recurrent network as follows:

- The compact operators in the analytic method (e.g. the low pass filter J) are replaced by sparse matrixes $A_\theta^{\ell,i,(1)}$. These matrixes are learned from the training data.
- Non-compact operators in the analytic method (e.g. the identity operator I or the time reversal R) determine the fixed matrixes $A^{\ell,i,(0)}$. The matrixes $A^{\ell,i,(0)}$ are not learned but determined by the analytic method.

Loss function and regularization

Next, we consider a general target function $f : \mathbb{R}^{n\times n} \to \mathbb{R}^n$.
We want to learn the parameters θ such that the neural network $f_\theta : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ approximate the function $f : \mathbb{R}^{n\times n} \to \mathbb{R}^n$.

Definition

The regularized loss function $\mathcal{L}$ with regularization parameter $\alpha > 0$ is given by

$$\mathcal{L}(\theta, Y) = \|f_\theta(Y) - f(Y)\|^2_{\mathbb{R}^n} + \alpha\mathcal{R}(\theta)$$

To make the weight matrixes $A_\theta^{\ell,i,(1)}$ sparse, we use the ℓ^1-norm

$$\mathcal{R}(\theta) = \|\theta\|_1 = \sum_{\ell,k,p} \|\theta_p^{\ell,i}\|_{\mathbb{R}^n}.$$

Training a neural network with sampled data

Assume that Y is random and has a priori distribution μ, that is, $Y \sim \mu$.

Let $Y_1, Y_2, \dots, Y_N$ be independent samples from a priori distribution μ. Suppose we are given the training set

$$S = \{(Y_1, f(Y_1)), \dots, (Y_N, f(Y_N))\}.$$

Training of the neural network means minimizing the the empirical loss function,

$$\theta(S) = \operatorname*{argmin}_{\theta} \mathcal{L}(\theta, S),$$

$$\mathcal{L}(\theta, S) = \frac{1}{N} \sum_{i=1}^{N} \|f_\theta(Y_i) - f(Y_i)\|^2_{\mathbb{R}^n} + \alpha \|\theta\|_1.$$

Definition of the optimal neural network

For a network f_θ with parameters θ, the expected loss is

$$\mathcal{L}(\theta, \mu) := \mathbb{E}_{Y \sim \mu} \left[\mathcal{L}(\theta, Y) \right].$$

The parameters θ^* of the optimal neural network $f_{\theta^*} : \mathbb{R}^{n \times n} \to \mathbb{R}^n$ are

$$\theta^* = \underset{\theta}{\text{argmin}}\ \mathcal{L}(\theta, \mu).$$

Neural network vs. analytic solution algorithm

Let $f_{\theta_0}(Y)$ be a deterministic approximation of an analytic solution algorithm (e.g. the analytic solution method for the inverse problem).

A trivial, but important result is that

$$\mathbb{E}_{Y\sim\mu}\left[\mathcal{L}(\theta^*, Y)\right] \leq \mathbb{E}_{Y\sim\mu}\left[\mathcal{L}(\theta_0, Y)\right].$$

This means that the optimal neural network $f_{\theta^*}(Y)$ has at least as good expected performance as $f_{\theta_0}(Y)$.

Approximation of the target function by a neural network

Definition

We say that the target function $f : \mathbb{R}^{n \times n} \to \mathbb{R}^n$ can be approximated with accuracy ε_0 by a neural network with a depth L and a sparsity bound R_0, if there is θ_0 such that

$$\|\theta_0\|_1 \leq R_0, \tag{1}$$

and the network f_{θ_0} satisfies

$$\sup_{\|Y\| \leq 1} \|f(Y) - f_{\theta_0}(Y)\|_{\mathbb{R}^n} \leq \varepsilon_0. \tag{2}$$

Stability results for the inverse problem for the 1-dimensional wave equation [Korpela-L.-Oksanen 2018], show that (1)-(2) are valid with $\varepsilon_0 > 0$, $L = C \log(1/\varepsilon_0)$, $n = C\varepsilon_0^{-175}$, and $R_0 = C\varepsilon_0^{-16}$.

How well a trained network works?

Next we estimate the expected performance gap between the trained neural network $f_{\theta(S)}$ and the optimal neural network f_{θ^*}, that is,

$$\mathcal{G}_{per}(S) = \left| \mathbb{E}_{Y \sim \mu} \mathcal{L}(\theta(S), Y) - \mathbb{E}_{Y \sim \mu} \mathcal{L}(\theta^*, Y) \right|$$

$\mathcal{G}_{per}(S)$ is the difference of the expected loss of $f_{\theta(S)}$ and f_{θ^*}.

Also, we estimate the expected generalization error that is the difference of the empirical loss function and the true loss function for the neural network $f_{\theta(S)}$,

$$\mathcal{G}_{gen}(S) = \left| \mathcal{L}(\theta(S), S) - \mathbb{E}_{Y \sim \mu} \mathcal{L}(\theta(S), Y) \right|.$$

$\mathcal{G}_{gen}(S)$ measures how well we can estimate the performance of $f_{\theta(S)}$ with a general input Y by using only the training data.

Theorem

Let $\alpha > 0$.
Let $S = \{(Y_1, f(Y_1)), \dots, (Y_N, f(Y_N))\}$ *be the training set that consists of* N *independent samples from the distribution* μ. *Then,*

$$\mathbb{P}_{S\sim\mu^N}\left[\mathcal{G}_{gen}(S) \leq \delta\right] \geq 1 - C_1 \left(\frac{1}{\delta}\right)^{C_2} \exp\left(-\frac{1}{50n^2\|f\|_\infty^4}\delta^2 \cdot N\right)$$

where

$$C_1 = \exp\left(8^{L+4} n^{\frac{3}{2}} (1 + \|f\|_\infty) \exp(5\|f\|_\infty^2 \alpha^{-1})\right),$$
$$C_2 = 8^{L+1} n \exp(4\|f\|_\infty^2 \alpha^{-1}),$$

Theorem

Suppose the target function f can be approximated with accuracy ε_0 by a neural network with the depth L and the sparsity bound R_0.
Let $\alpha \geq \varepsilon_0^2/R_0$.
Let $S = \{(Y_1, f(Y_1)), \dots, (Y_N, f(Y_N))\}$ be the training set that consists of N independent samples from the distribution μ. Then,

$$\mathbb{P}_{S\sim\mu^N}\left[\mathcal{G}_{gen}(S) \leq \delta\right] \geq 1 - C_1\left(\frac{1}{\delta}\right)^{C_2} \exp\left(-\frac{1}{50n^2\|f\|_\infty^4}\delta^2 \cdot N\right)$$

where

$$C_1 = \exp\left(8^{L+3} n^{\frac{3}{2}} (R_0 + L + \|f\|_\infty)\, e^{6R_0}\alpha^{-1/2}\right),$$
$$C_2 = 8^{L+1} n\, e^{6R_0}\alpha^{-1/2}.$$

Theorem

Suppose the target function f can be approximated with accuracy ε_0 by a neural network with the depth L and the sparsity bound R_0.
Let $\alpha \geq \varepsilon_0^2/R_0$.
Let $S = \{(Y_1, f(Y_1)), \dots, (Y_N, f(Y_N))\}$ be the training set that consists of N independent samples from the distribution μ. Then,

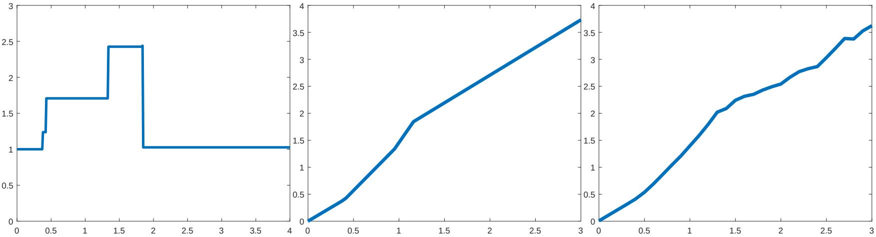
$$\mathbb{P}_{S\sim\mu^N}\left[\mathcal{G}_{per}(S) \leq 2\delta\right] \geq 1 - 2C_1\left(\frac{1}{\delta}\right)^{C_2} \exp\left(-\frac{1}{50n^2\|f\|_\infty^4}\delta^2 \cdot N\right)$$

where

$$C_1 = \exp\left(8^{L+3} n^{\frac{3}{2}} (R_0 + L + \|f\|_\infty)\, e^{6R_0}\alpha^{-1/2}\right),$$
$$C_2 = 8^{L+1} n\, e^{6R_0}\alpha^{-1/2}.$$

Learning travel depth in inverse problem for wave equation

Preliminary numerical tests on solving the inverse problem for a wave equation with a recurrent operator neural network (without sparsity):



(a) $c(x)$ (b) True depth (c) Predicted depth

Sample piecewise-constant wavespeed $c(x)$; True depth $\tau^{-1}(t)$ on how deep the waves propagate as a function of time t; Predicted depth as a function of time.

Numerical details: Training with piecewise-constant medium; 5000 data pairs, 20% withheld as testing data; Testing error: 6.3e-5; Networks with 16.5M parameters, sparsity regularization is not yet implemented.

Thank you for your attention!

References:

- M. de Hoop, M. Lassas, C. Wong: Deep learning architectures for nonlinear operator functions and nonlinear inverse problems. 2019. arXiv:1912.11090.
- J. Korpela, M. Lassas, L. Oksanen: Discrete regularization and convergence of the inverse problem for 1+1 dimensional wave equation. Inverse Problems and Imaging 13 (2019), 575-596.
- K. Bingham, Y. Kurylev, M. Lassas, S. Siltanen: Iterative time reversal control for inverse problems. Inverse Problems and Imaging 2 (2008), 63-81.
- A. Katchalov, Y. Kurylev, M. Lassas: Inverse Boundary Spectral Problems, CRC-press, 2001, xi+290 pp.
- D. Tataru: Unique continuation for solutions to PDE's; between Hörmander's theorem and Holmgren's theorem, Comm. PDE 20 (1995), 855-884.
- M. Belishev, Y. Kurylev: To the reconstruction of a Riemannian manifold via its spectral data (BC-method). Comm. PDE 17 (1992), 767–804.
- M. Belishev: On an approach to multidimensional inverse problems for the wave equation, Dokl. Akad. Nauk SSSR (in Russian). 297 (1987), 524-527.
- M. Belishev and Y. Kurylev: Non-stationary inverse problem for the multidimensional wave equation "in large", Zap. Nauk. Semin. LOMI (in Russian) 165 (1987), 21-30.
- I. Gelfand, B. Levitan: On the determination of a differential equation from its spectral function. Izvestiya Akad. Nauk SSSR. 15 (1951), 309–360.