New deep neural networks solving non-linear inverse problems

Matti Lassas
and
Christopher Wong
Maarten de Hoop

University of Helsinki

April 20, 2020

Inverse problem in a 3-dimensional body

In the inverse problem for the wave equation we aim to find the unknown wave speeds $c(x)$ from boundary measurements.

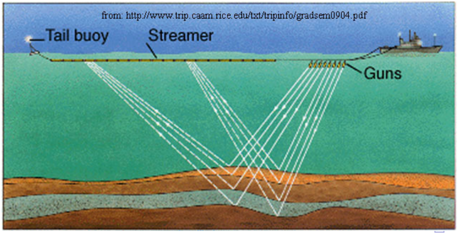
Let $u(x,t) = u^h(x,t)$ solve the wave equation

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

where h is boundary source, $M \subset \mathbb{R}^3$. The source-to-solution map is

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

Next we consider this problem in the 1-dimensional case and solve it using neural networks.

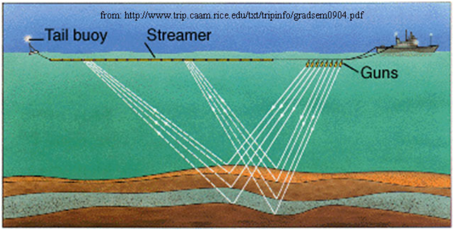


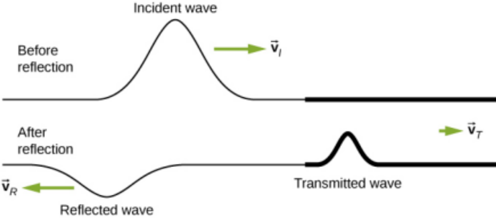
Overview of the talk

1. We consider the solution map $F : \Lambda_c \to c$ that solves the inverse problem in the 1-dimensional case.
2. We propose an architecture of neural networks, where the input is a linear operator Λ.
3. We show that the solution map F 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 source-to-solution map, $\Lambda = \Lambda_c$,

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

Λ_c is a linear operator or "a matrix". Physically the above means that

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

Travel time coordinate

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

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

The domain of influence corresponding to time s is the set

$$M(s) = \{x \in \mathbb{R}_+ :\ \tau(x) \leq s\}$$

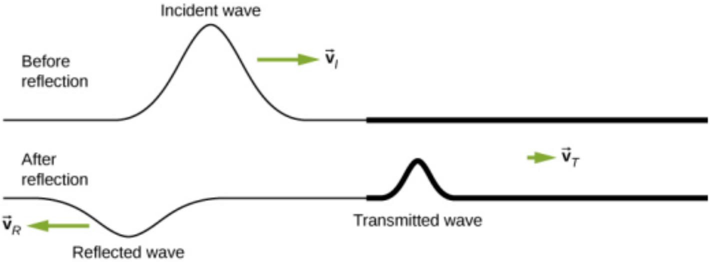
It consists of the points x for which the travel time $\tau(x)$ to the boundary is at most s.

Inverse problem in travel time coordinates

We study the inverse problem of finding $c(x)$ when Λ_c is given. Let

$$F : \Lambda_c \to (c \circ \tau^{-1}, \tau^{-1})$$

be the map that solves the inverse problem.
The map F is non-linear.
We aim to construct a neural network that approximates $F(\Lambda)$.



Source-to-solution 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,

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

where $\Lambda = \Lambda_c$ is the source-to-solution map,

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

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 Λ.

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

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

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

$$\lim_{\beta\to 0} u^{h_{\beta,s}}(x,T) = 1_{M(s)}(x).$$

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

Thus, when β is small,

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

An analytic solution algorithm for the inverse problem

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

$$V(s) = \lim_{\beta\to 0} \int_0^T h_{\beta,s}(t)\,(T-t)dt = \lim_{\beta\to 0} \langle u^{h_{\beta,s}}(T), 1\rangle_{L^2(M)}$$
$$= \int_0^{\tau^{-1}(s)} \frac{1}{c(x)^2}\,dx,$$
$$w(s) = \partial_s V(s).$$

Then

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

An analytic solution algorithm for the inverse problem

The above minimization problem can be solved using an iteration. Hence the inverse problem is solved as follows:

Step 1: For $j = 1, \ldots, K$, let $s_j = j\frac{T}{2K}$ and $h^{(j)} = h^{(j)}_L$ be computed by doing L steps of the iterated soft thresholding,

$$h^{(j)}_{\ell+1} \;:=\; \sigma_\beta(h^{(j)}_\ell - AP_j(J\Lambda - R\Lambda RJ)P_j A^* h^{(j)}_\ell + AP_j b), \quad h^{(j)}_0 = 0.$$

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

R is the time-reversal operator, J is a low-pass filter,
$A : L^2(0, 2T) \to \ell^2$ is an isometry, $P_j h = 1_{(T-s_j,T)} \cdot h$, $b = (T - t)_+$.

Also, $\sigma_\beta(x) = \text{relu}(x - \beta) - \text{relu}(-x - \beta)$ is the soft thresholding of x.

Step 2. Compute $c(\tau^{-1}(s_j)) \approx G_j(h^{(1)}, \ldots, h^{(K)})$, where G_j are explicit functions.

Summary on the analytic solution of the inverse problem

Consider the map $F : \Lambda_c \to (c \circ \tau^{-1}, \tau^{-1})$ that reconstructs the wave speed $c(x)$ from the boundary measurements Λ_c.
The discretized version of this map, $F : \mathbb{R}^{n\times n} \to \mathbb{R}^{2K}$ can be written as

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

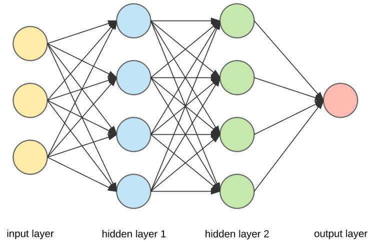
where $f^{(j)} : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ map Λ_c to the optimized sources,

$$f^{(j)}(\Lambda_c) = h^{(j)}.$$

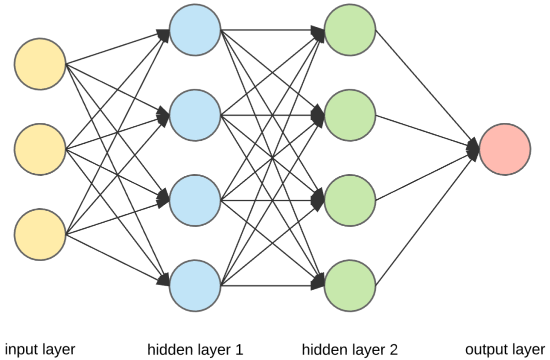
Next we define a family of neural networks than can approximate functions $f^{(j)} : \mathbb{R}^{n\times n} \to \mathbb{R}^n$.
The explicit function G can be approximated by a standard neural network.
Then, we can approximate F by a neural network.

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,

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

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

$$\begin{aligned} y_0 &= x, \\ y_{\ell+1} &= \phi\left(A_\theta^\ell y_\ell + b_\theta^\ell\right), \\ f_\theta(x) &= y_L. \end{aligned}$$

Architecture:

- ℓ: the layer index, max depth L.
- y_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 $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$.
- ϕ is the activation function, the Rectified Linear Unit (relu)

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

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

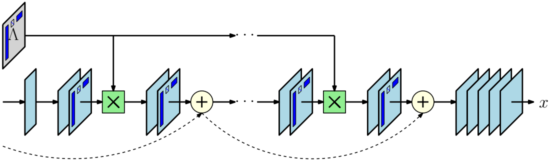
$$\begin{aligned} y_0 &= x, \\ y_{\ell+1} &= \phi\left(A_\theta^\ell y_\ell + b_\theta^\ell\right), \\ f_\theta(x) &= y_L. \end{aligned}$$

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 $\Lambda \in \mathbb{R}^{n\times n}$ and computes following operations

$$\begin{aligned} y_0 &= b^0, \\ y_{\ell+1} &= \phi\left(A_\theta^\ell \, \Lambda \, y_\ell + b_\theta^\ell\right), \\ f_\theta(\Lambda) &= y_L. \end{aligned}$$

Outline:

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

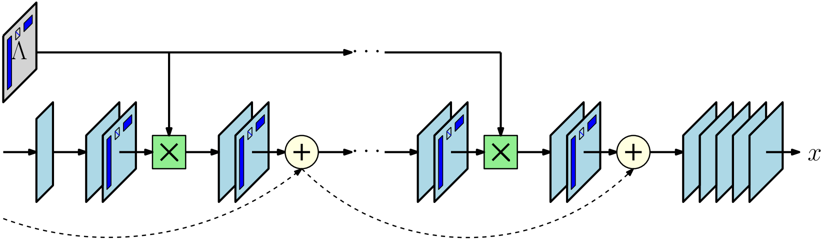


Definition

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

$$f_\theta(\Lambda) = h_L,$$
$$h_\ell = b_\theta^{\ell,1} + A_\theta^{\ell,1} h_{\ell-1} + A_\theta^{\ell,2} \Lambda h_{\ell-1} + \phi\left[b_\theta^{\ell,2} + A_\theta^{\ell,3} h_{\ell-1} + A_\theta^{\ell,4} \Lambda h_{\ell-1} \right],$$

where the initial vector $h_0 = b_\theta^{0,1} \in \mathbb{R}^n$ is independent of the input Λ 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.



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 $\theta_p^{\ell,i} \in \mathbb{R}^n$.

We replace the compact operators in the analytic method (e.g. the low pass filter J) by sparse matrixes and learn these matrixes $A_\theta^{\ell,i,(1)}$ from training data.

Non-compact operators in the analytic method (e.g. the identity operator I or the time reversal R) determine the fixed matrixes $A^{\ell,i,(0)}$.

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

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

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 Λ is random and has a priori distribution μ, that is, $\Lambda \sim \mu$.

Let $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$ be independent samples from a priori distribution μ. Suppose we are given the training set

$$S = \{(\Lambda_1, f(\Lambda_1)), \ldots, (\Lambda_N, f(\Lambda_N))\}.$$

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

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

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

Definition of the optimal neural network

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

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

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

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

Neural network vs. analytic solution algorithm

Let $f_{\theta_0}(\Lambda)$ 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

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

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

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

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

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,

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

$\mathcal{G}_{per}(S)$ 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 $f_{\theta(S)}$,

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

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

Theorem

Let $\alpha > 0$.
Let $S = \{(\Lambda_1, f(\Lambda_1)), \ldots, (\Lambda_N, f(\Lambda_N))\}$ *be the training set that consists of* N *independent samples from the distribution* μ. *Then,*

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

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 $S = \{(\Lambda_1, f(\Lambda_1)), \dots, (\Lambda_N, f(\Lambda_N))\}$ 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 $S = \{(\Lambda_1, f(\Lambda_1)), \dots, (\Lambda_N, f(\Lambda_N))\}$ be the training set that consists of N independent samples from the distribution μ. Then,

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

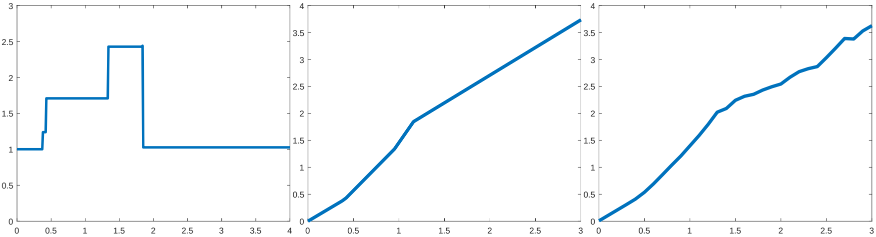
where

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

$$C_2 = 8^{L+1}n\; e^{6R_0}\alpha^{-1/2}.$$

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.
- M. Lassas: Inverse problems for linear and non-linear hyperbolic equations. Proceedings of International Congress of Mathematicians ICM 2018, Rio de Janeiro. 2018, Vol III, pp. 3739-3760.
- 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.
- M. Belishev, Y. Kurylev: To the reconstruction of a Riemannian manifold via its spectral data (BC-method). Comm. PDE 17 (1992), 767–804.