

Решение обратных задач методами машинного обучения

Михаил Лытаев
mikelytaev@gmail.com

с.н.с. СПб ФИЦ РАН
доцент Университета ИТМО

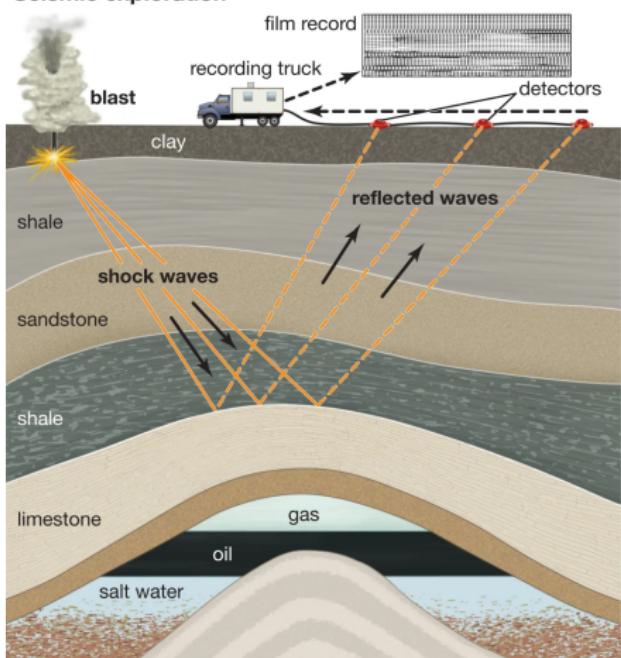
11 июня 2025

План

- ① Примеры обратных задач
- ② Проблемы
- ③ Математическая постановка обратных задач
- ④ Классические методы решения
- ⑤ Методы МЛ для решения обратных задач

Геофизика и сейсморазведка

Seismic exploration



- $c(x, y, z)$, $\rho(x, y, z)$, $\alpha(x, y, z)$
 - нужно определить трехмерные поля скорости звука, плотности и затухания
- записываем сейсмограмму (временной ряд)
- \mathbf{m} - неизвестные параметры модели (среды)
- \mathbf{d} - измерения
- $G(\mathbf{m}) = \mathbf{d}$ - прямая задача
- обратная задача:

$$\arg \min_{\mathbf{m}} \|G(\mathbf{m}) - \mathbf{d}\|$$

Медицинская томография (акустическая и электромагнитная)

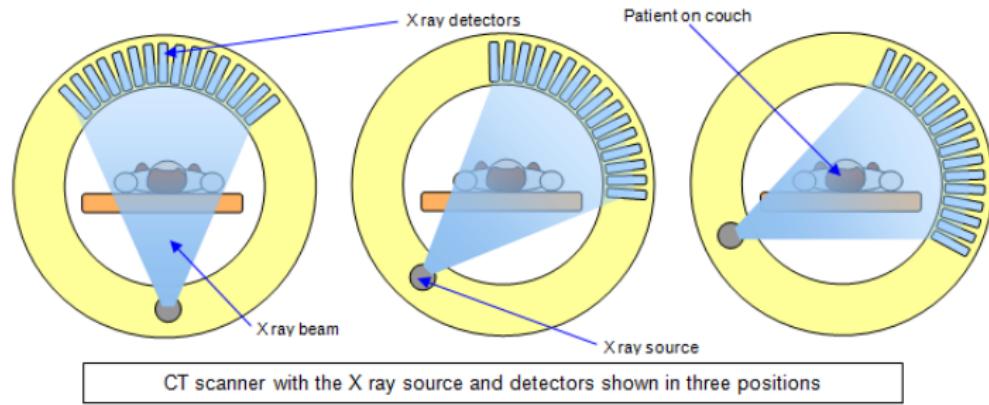


Рис.:

Океанология

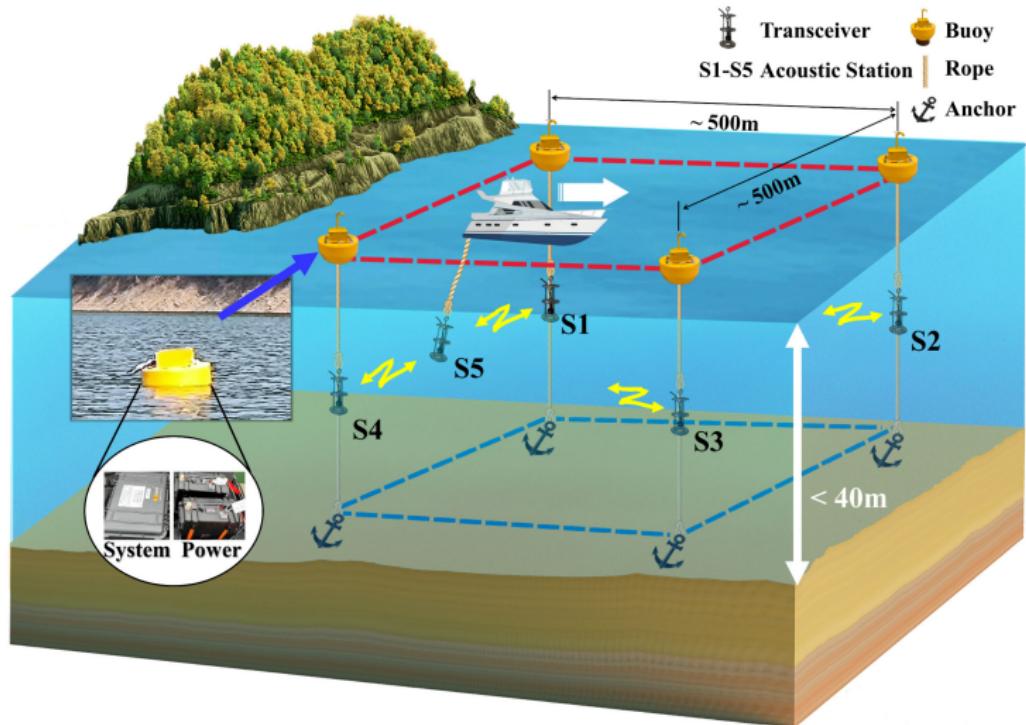


Рис.: Xu S. et al. Flow current field observation with underwater moving acoustic tomography //Frontiers in Marine Science. – 2023.

Зондирование атмосферы

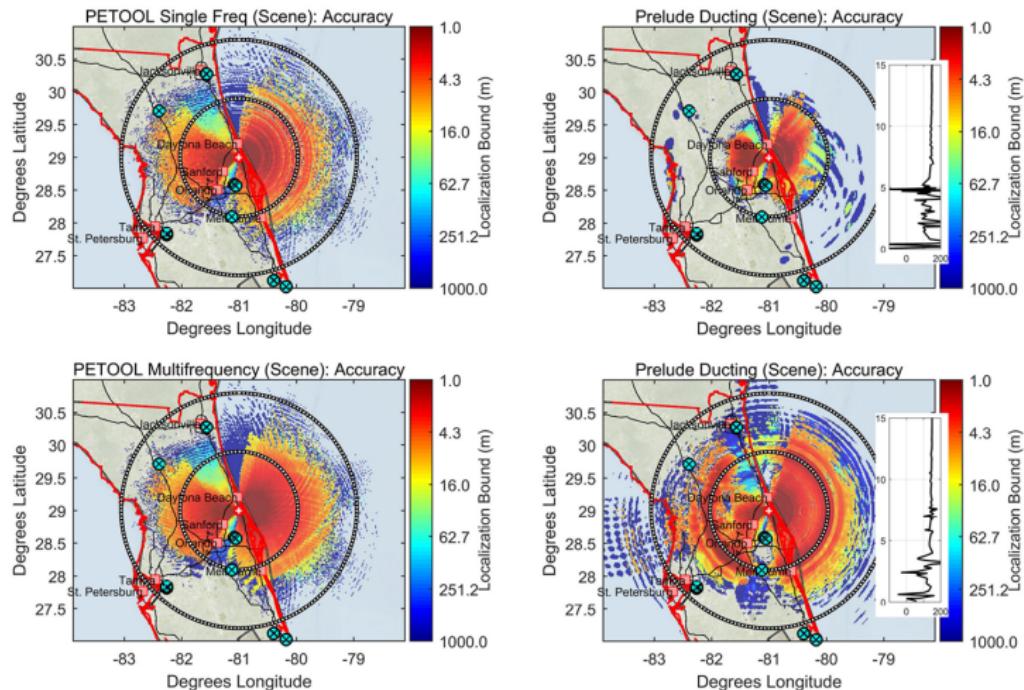
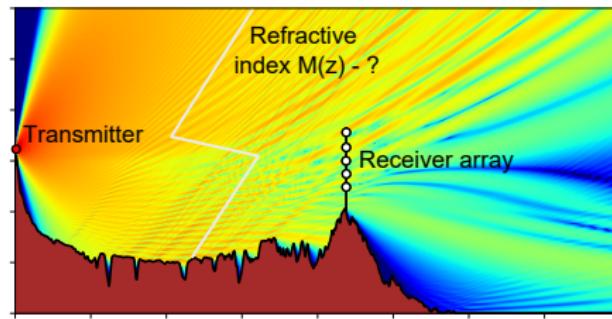
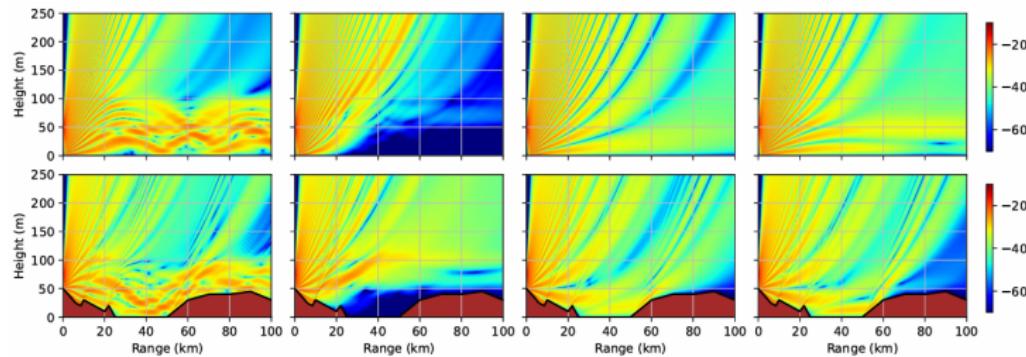


Рис.: Barott W. C., Scott K. M., Himed B. Effects of atmospheric refractivity and variability on passive radar performance prediction //2018 IEEE Radar Conference (RadarConf18). – IEEE, 2018.

Томография тропосферной рефракции



Другие применения

- Материаловедение и метаматериалы
- Квантовая физика и вычислительная химия
- Оптимальное управление
- Динамические системы

Проблемы

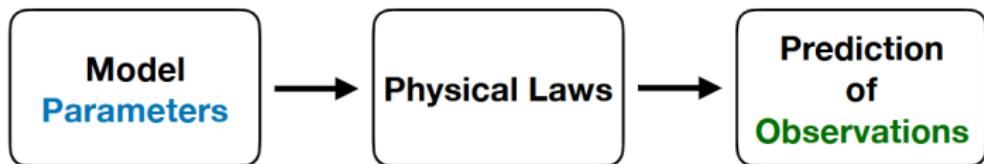
- Требуется слишком много дорогостоящих высокоточных приёмников
- Повышение точности (не только приближенные картинки но и количественные данные)
- Увеличение скорости вычислений и снижение вычислительных затрат
- 3D
- Мультимодальность
- Оценка погрешности

Понятие корректности задачи

- ① Решение существует
- ② Решение единственное
- ③ Решение непрерывно зависит от начальных данных

Прямые и обратные задачи

Forward Problem



Inverse Problem

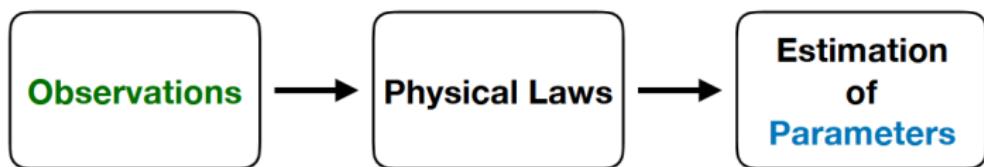


Рис.: Kailai Xu, and Eric Darve. Machine Learning for Inverse Problems in Computational Engineering

Постановка задачи

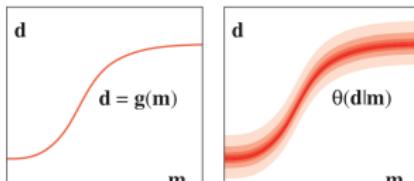
$$\arg \min_{\boldsymbol{m}} \|G(\boldsymbol{m}) - \boldsymbol{d}\|$$

- $G : \mathcal{M} \rightarrow \mathcal{D}$ - нелинейный невыпуклый оператор (для решения прямой задачи)
- \boldsymbol{d} - измерения
- \boldsymbol{m} - вектор параметров модели

Постановка задачи в вероятностной форме

$$\sigma_M(\mathbf{m}) = k \rho_M(\mathbf{m}) \int_{\mathfrak{D}} d\mathbf{d} \frac{\rho_D(\mathbf{d}) \theta(\mathbf{d}|\mathbf{m})}{\mu_D(\mathbf{d})}$$

- ρ_M, σ_M - априорное и апостериорное распределения для \mathbf{m}
- ρ_D - вероятностное распределение данных измерений



- $\theta(\mathbf{d}|\mathbf{m})$ прямая модель

Постановка задачи в вероятностной форме

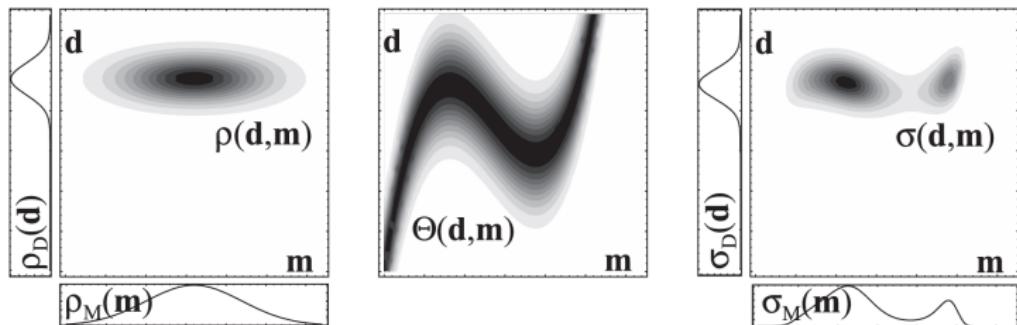


Рис.: Tarantola A. Inverse problem theory and methods for model parameter estimation. – SIAM, 2005.

Классическая теория некорректных задач

Классический пример некорректной задачи: интегральное уравнение Фредгольма первого рода

$$\int_a^b K(x, s) z(s) ds = u(x), \quad x \in [c, d]$$

- Метод квазирешений
- Метод регуляризации Тихонова

Тихонов А. Н., Леонов А. С., Ягола А. Г. Нелинейные некорректные задачи. М.: Наука, 1995.

Классические методы решения нелинейных задач

- Линеаризация
- Локальные итерационные методы
- Монте-Карло
- Генетические алгоритмы
- Метод имитации отжига

Обучение обратного оператора

- Оператор прямой задачи

$$G : \mathcal{M} \rightarrow \mathcal{D}$$

- Нужно найти обратный оператор

$$G_{\theta}^{-1} : \mathcal{D} \rightarrow \mathcal{M}$$

- \mathcal{M} - множество возможных параметров модели
 - ▶ Как его формально определить?

$$\mathbf{m}^{(i)} \sim \mu$$

- ▶ μ - вероятностное распределение на \mathcal{M} (можно обучить генеративную модель)

$$\arg \min_{\theta} \mathbb{E}_{m \sim \mu} \| G_{\theta}^{-1}(G(\mathbf{m})) - \mathbf{m} \|_{\mathcal{M}}^2 \approx$$

$$\arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \| G_{\theta}^{-1}\left(G\left(\mathbf{m}^{(i)}\right)\right) - \mathbf{m}^{(i)} \|_{\mathcal{M}}^2$$

- EOF

DeepONet

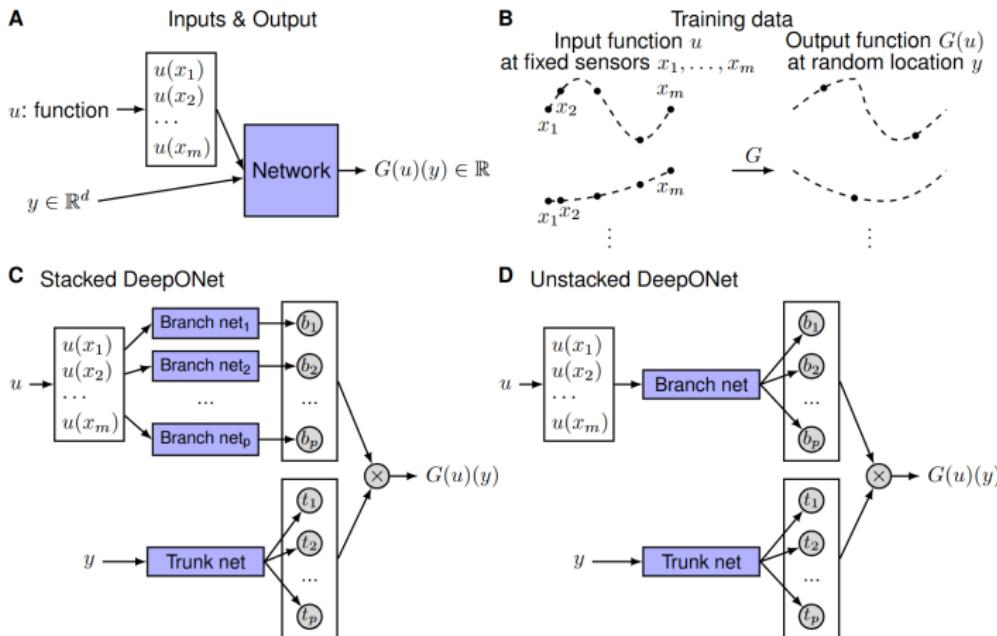


Рис.: Lu L. et al. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators // Nature machine intelligence. – 2021.

Схема обучения

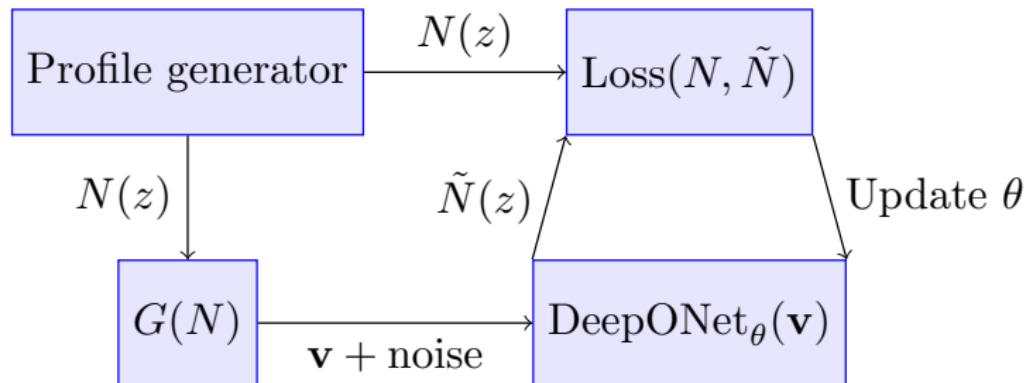


Рис.:

Универсальная теорема аппроксимации (для операторов)

Theorem 1 (Universal Approximation Theorem for Operator). Suppose that σ is a continuous non-polynomial function, X is a Banach Space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and \mathbb{R}^d , respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n , p , m , constants $c_i^k, \xi_{ij}^k, \theta_i^k, \zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \dots, n$, $k = 1, \dots, p$, $j = 1, \dots, m$, such that

$$\left| G(u)(y) - \underbrace{\sum_{k=1}^p \sum_{i=1}^n c_i^k \sigma \left(\underbrace{\sum_{j=1}^m \xi_{ij}^k u(x_j) + \theta_i^k}_{\text{branch}} \right)}_{\text{trunk}} \right| < \epsilon \quad (1)$$

holds for all $u \in V$ and $y \in K_2$.

Рис.: Lu L. et al. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators // Nature machine intelligence. – 2021.

Нейронные операторы

Входные и выходные данные нейронных сетей - вектора, а нужно чтобы были функции

$$\mathcal{N}(x) = \sigma(A_L(\dots\sigma(A_1x + b_1)\dots) + b_L)$$

A_i - матрицы; x, b_i - вектора

Заменим матричное умножение на линейный интегральный оператор

$$u_{i+1}(x) = \sigma \left(\int_{\Omega_i} K_\theta^{(i)}(x, y) u_i(y) dy + b_i(x) \right)$$

$K_\theta^{(i)}(x, y)$ - (обучаемое) ядро

Функция Грина

$$-\nabla^2 u = f, \quad x \in \Omega \subset \mathbb{R}^d, \quad u|_{\partial\Omega} = 0$$

$$\mathcal{A}(f) = \int_{\Omega} G(\cdot, y) f(y) dy = u$$

$G(\cdot, y)$ - функция Грина

Функция Грина

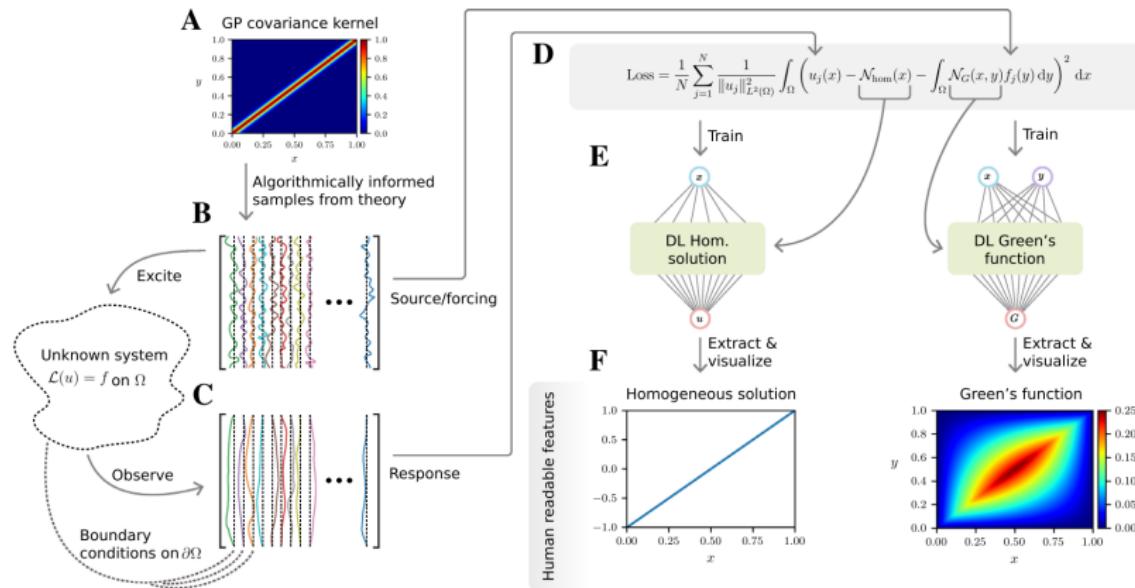


Рис.: Boullé N., Earls C. J., Townsend A. Data-driven discovery of Green's functions with human-understandable deep learning //Scientific reports. – 2022.

Оператор Фурье

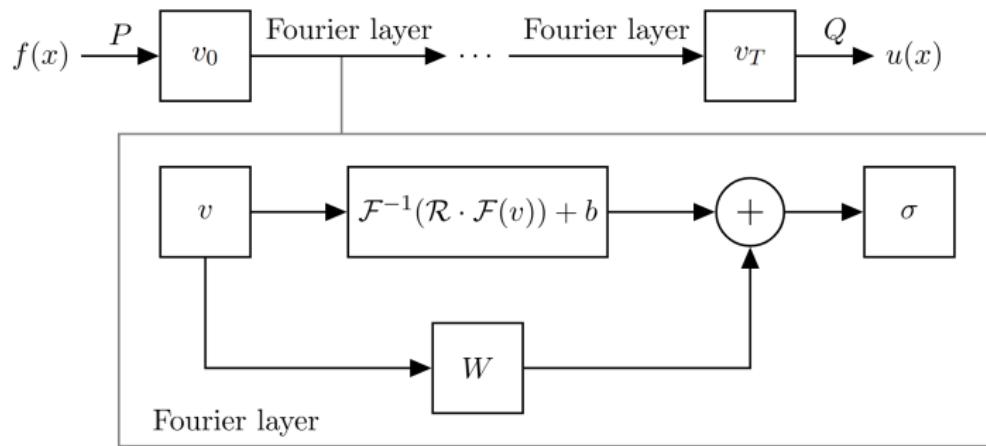


Рис.: Boullé N., Townsend A. A mathematical guide to operator learning //Handbook of Numerical Analysis. – Elsevier, 2024.

Physics-Informed Neural Networks (PINN)

Физические законы содержатся в (синтетических) данных, но не в самой модели

$G : \mathcal{M} \rightarrow \mathcal{D}$ - прямая задача, обычно является решением дифференциального уравнения

$$f(\mathbf{u}(x, y, z, t), \mathbf{m}(x, y, z, t)) = 0, \quad (x, y, z, t) \in \Omega$$

$$\mathcal{B}(\mathbf{u}, x, y, z, t) = 0, \quad (x, y, z, t) \in \partial\Omega$$



$$G\mathbf{m} = \mathbf{u}|_{(x, y, z, t) \subset \tau_{\text{measures}}} = \mathbf{d}$$

Physics-Informed Neural Networks (PINN)

$$\mathbf{u}(x, y, z, t) = \text{NN}_{\theta_u}^u(x, y, z, t)$$

$$\mathbf{m}(x, y, z, t) = \text{NN}_{\theta_m}^m(x, y, z, t)$$

$$\begin{aligned}\mathcal{L}(\theta_u, \theta_m) = & ||f\left(\text{NN}_{\theta_u}^u(x, y, z, t), \text{NN}_{\theta_m}^m(x, y, z, t)\right)||^2 + \\ & \lambda_{\text{boundary}} ||\mathcal{B}\left(\text{NN}_{\theta_u}^u\right)||^2 + \\ & \lambda_{\text{measure}} ||\text{NN}_{\theta_u}^u(x, y, z, t) - \mathbf{d}_{\text{measured}}||^2\end{aligned}$$

$$\arg \min_{\theta_u, \theta_m} \mathcal{L}(\theta_u, \theta_m)$$

- У обученного оператора фиксированное время инференса
- В PINN каждый раз нужно решать оптимизационную задачу

Пример

- Дано

- ▶ Уравнение теплопроводности

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(k(x) \frac{\partial u}{\partial x} \right) + f(x, t)$$

- ▶ Условия на границе
 - ▶ Измерения величины u на τ_{measures}

$$u|_{(x,t) \in \tau_{\text{measures}}} = d$$

- ▶ Функция источник $f(x, t)$
 - Требуется найти $k(x)$

$$u(x, t) = \text{NN}_{\theta_u}^u(x, t)$$

$$k(x) = \text{NN}_{\theta_k}^k(x)$$

Physics-Informed Neural Networks (PINN)

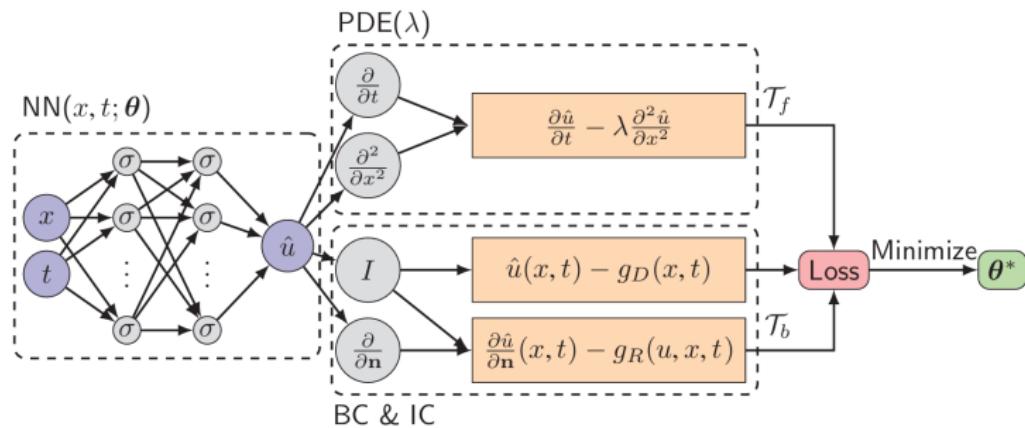


Рис.: Lu L. et al. DeepXDE: A deep learning library for solving differential equations //SIAM review. – 2021.

Physics-Informed Neural Networks (PINN)

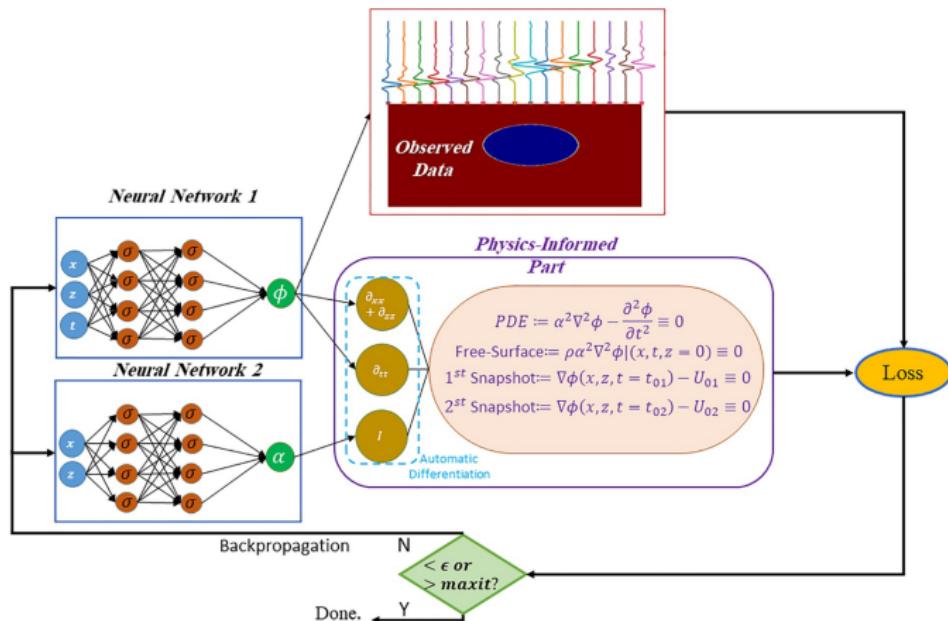
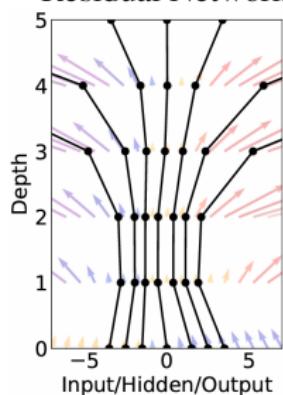


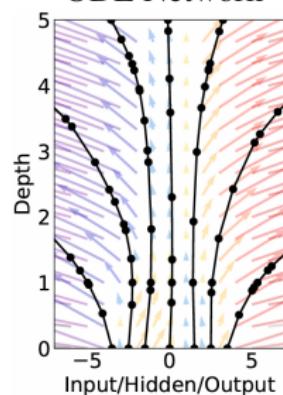
Рис.: Rasht-Behesht M. et al. Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions //Journal of Geophysical Research: Solid Earth. – 2022.

Нейронное ОДУ

Residual Network



ODE Network



Chen R. T. Q. et al. Neural ordinary differential equations //Advances in neural information processing systems. – 2018.

$$\frac{dy}{dt}(t) = f_{\theta}(t, y(t)) \quad (1)$$

$$y(0) = y_0$$

$$y_{n+1} = y_n + \Delta t f_{\theta}(n \Delta t, y(t))$$

Похоже на ResNet

Обучение нейронного ОДУ

$$L(y_\theta(T)) = L\left(y(0) + \int_0^T f_\theta(t, y_\theta(t)) dt\right) = L(\text{ODESolver}(y(0), f, \theta))$$

- Метод сопряженных уравнений (отдельное уравнение для градиента)
 - ▶ Для исходного уравнения
 - ▶ Громоздко
 - ▶ Лишние вычисления

Автоматически дифференцируемые численные схемы

- PINN «информирован» о физических законах, но не об особенностях их компьютерной реализации
- PINN одновременно ищет решение прямой задачи и параметры
- Автоматически дифференцируемые численные схемы

$$\begin{aligned}\mathcal{L}_{\text{PINN}} (\theta_u, \theta_m) = & \| f (\text{NN}_{\theta_u}^u (x, y, z, t), \text{NN}_{\theta_m}^m (x, y, z, t)) \|^2 + \\ & \lambda_{\text{boundary}} \| \mathcal{B} (\text{NN}_{\theta_u}^u) \|^2 + \\ & \lambda_{\text{measure}} \| \text{NN}_{\theta_u}^u (x, y, z, t) - \mathbf{d}_{\text{measured}} \|^2\end{aligned}$$

$$\mathcal{L}_{\text{NODE}} (\theta_m) = \| G_h (\text{NN}_{\theta_u}^u (x, y, z, t)) - \mathbf{d}_{\text{measured}} \|^2$$

- G_h - численная реализация солвера прямой задачи
 - ▶ Должна быть представима в виде графа вычислений

Автоматически дифференцируемые численные схемы

Differentiable step-by-step numerical scheme (implemented using JAX)
 $u(0, z) \rightarrow u(\Delta x, z) = \mathcal{P}(n_\theta) u(0, z) \rightarrow u(2\Delta x, z) = \mathcal{P}(n_\theta) u(\Delta x, z) \rightarrow \dots \rightarrow u(R, z)$

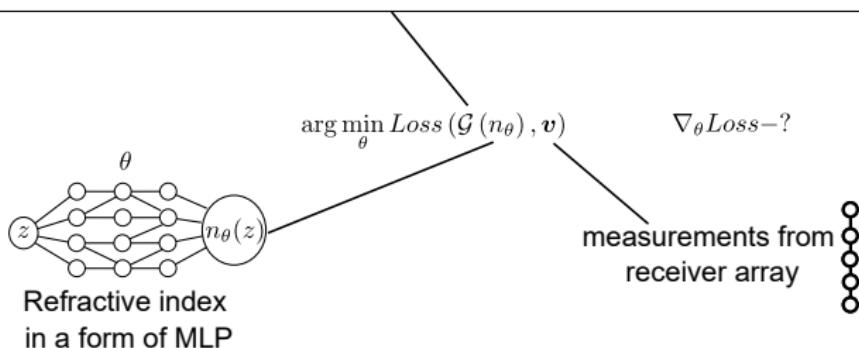
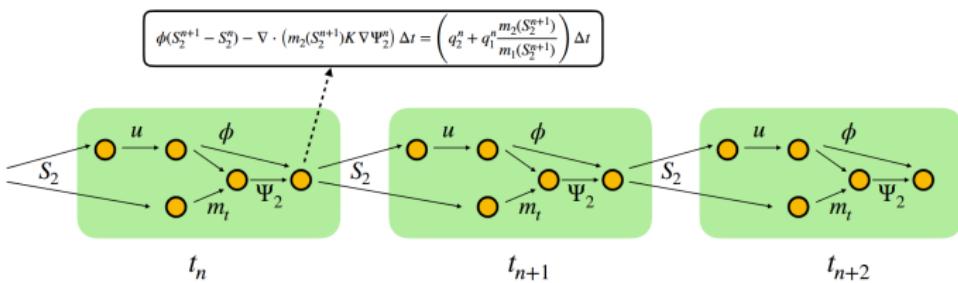


Рис.:



Система «хищник — жертва»

Модель Лотки-Вольтерр[а/ы]

$$\begin{cases} \frac{dx}{dt}(t) = \alpha x(t) - \beta x(t)y(t) \\ \frac{dy}{dt}(t) = -\gamma y(t) + \delta x(t)y(t) \end{cases}$$

$x(t)$, $y(t)$ - зависимость размера популяции жертвы и хищника от времени t .

Модель Лотки-Вольтерра

- Отсутствуют хищники

$$\frac{dx}{dt}(t) = \alpha x(t)$$

- ▶ α - скорость прироста популяции жертв

- Отсутствуют жертвы

$$\frac{dy}{dt}(t) = -\gamma y(t)$$

- ▶ γ - коэффициент убыли хищников

- Встреча хищников и жертв

- ▶ частота прямо пропорциональна произведению xy
- ▶ происходит убийство жертв с коэффициентом β
- ▶ сытые хищники способны к воспроизведству с коэффициентом δ .

Модель Лотки-Вольтерра

Population of Cheetahs and baboons

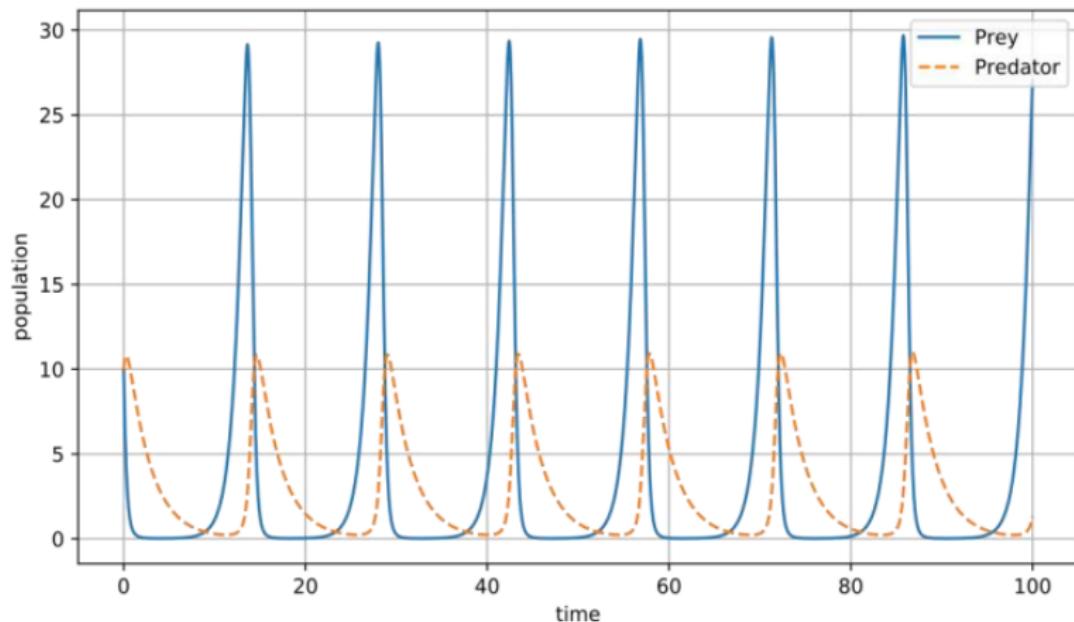


Рис.: моделирование популяции гепардов (хищники - Predator) и бабуинов (жертвы - prey).

Нейронная модель Лотки-Вольтерра

$$\begin{cases} \frac{dx}{dt}(t) = \alpha x(t) - \beta x(t)y(t) + f_\theta(x(t), y(t)) \\ \frac{dy}{dt}(t) = -\gamma y(t) + \delta x(t)y(t) + g_\theta(x(t), y(t)) \end{cases}$$

$f_\theta(x(t), y(t))$ - нейросеть

Обучение

- Датасет:
 - ▶ $x_i(t_j), y_i(t_j)$ - наблюдения размеров популяций
 - ▶ $i \in 1..N$ - независимые наблюдения
 - ▶ $j \in [1..M]$ - время наблюдения
- Обозначим $x_{x_0, y_0}(t), y_{x_0, y_0}(t)$ решение, полученное при начальном условии $x(0) = x_0$ и $y(0) = y_0$
- Функция потерь

$$L = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (x_{x_i(0), y_i(0)}(t_j) - x_i(t_j))^2 + (y_{x_i(0), y_i(0)}(t_j) - y_i(t_j))^2$$

Нейронная модель Лотки-Вольтерра

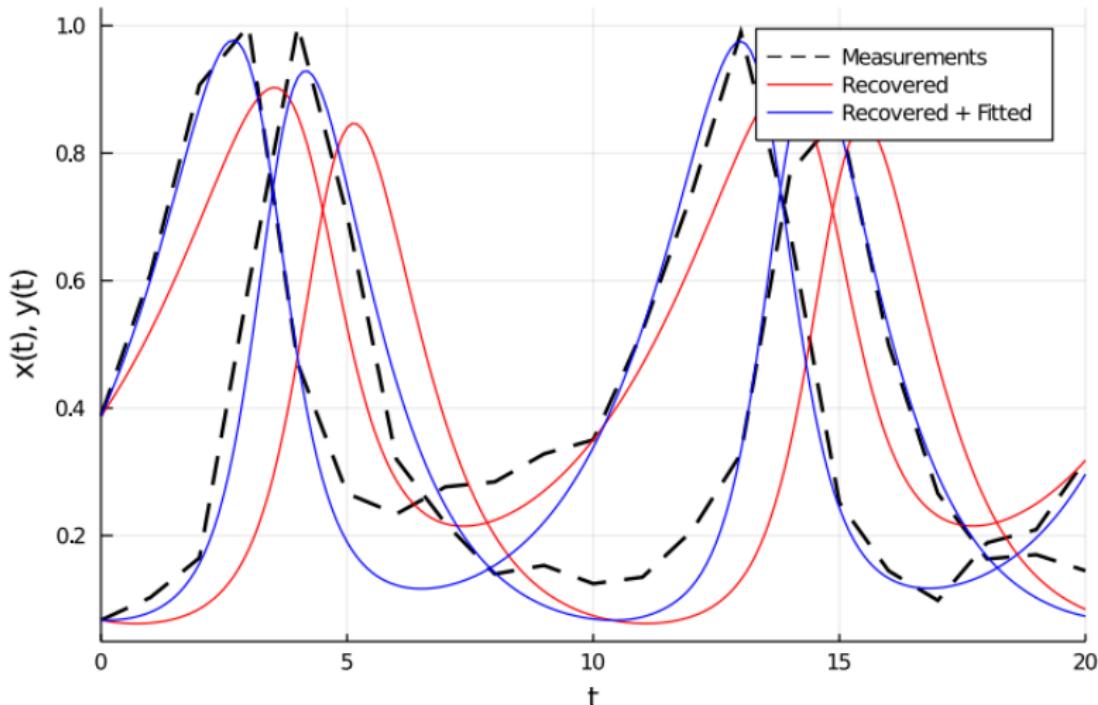


Рис.: Rackauckas C. et al. Universal differential equations for scientific machine learning //arXiv preprint arXiv:2001.04385. – 2020

Нейронная модель Лотки-Вольтерра

Recovered Model from Hudson Bay Data

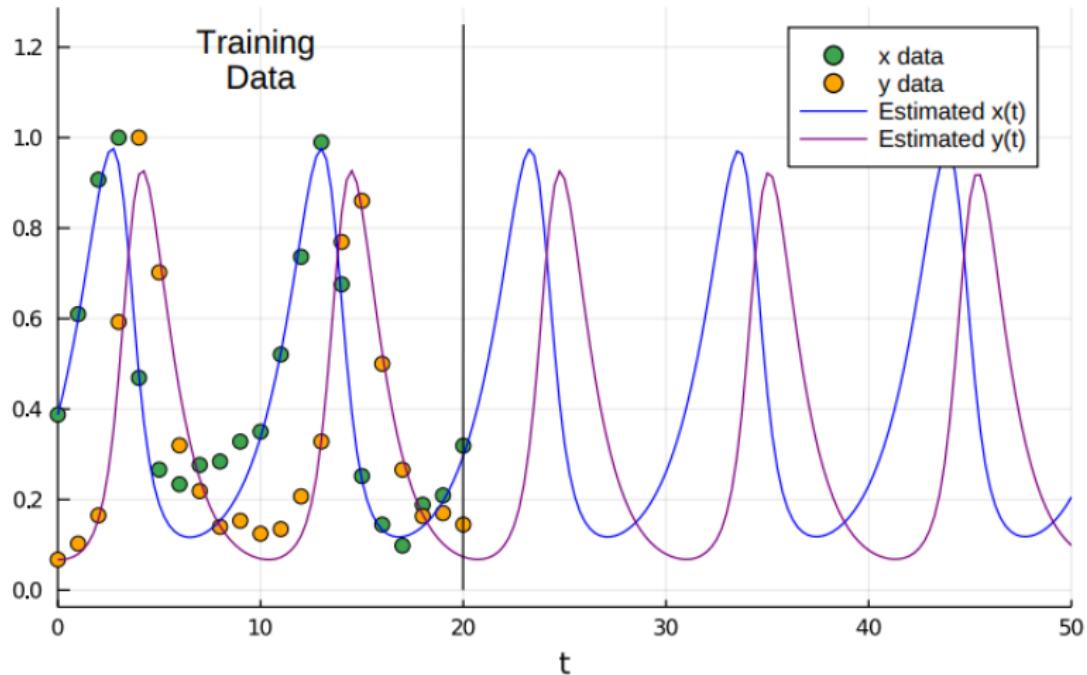


Рис.: Rackauckas C. et al. Universal differential equations for scientific machine learning //arXiv preprint arXiv:2001.04385. – 2020

Нейронные ОДУ и динамические системы

- Позволяет учитывать известную динамику модели
- Позволяет учитывать плохо формализуемые свойства модели, которые скрыты в данных
- МЛ модели без учета априорной информации о динамике (описываемой ДУ), будут, сложнее и менее точные

Библиотеки и фреймворки

- <https://github.com/lululxvi/deepxde>
 - ▶ PINN и DeepONet
- JAX
 - ▶ Автоматическое дифференцирование произвольного кода на Python (NumPy, Scipy)
- Neural ODE
 - ▶ <https://github.com/Zymrael/awesome-neural-ode>
 - ▶ tensorflow (tfp.math.ode)
 - ▶ PyTorch: pip install torchdiffeq
- <https://github.com/SciML>
 - ▶ Экосистема SciML на языке julia
- Вероятностные решения
 - ▶ PyMC
 - ▶ Blackjax
 - ▶ Numpyro

Нейронное параболическое уравнение теории дифракции

- Многослойный перцепtron в качестве неизвестного индекса преломления $\text{MLP}_\theta(z)$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial z^2} + k^2 \text{MLP}_\theta(z) \psi = 0$$

- Однонаправленное уравнение Гельмгольца

$$\frac{\partial \psi}{\partial x} = i \sqrt{\frac{\partial^2}{\partial z^2} + k^2 \text{MLP}_\theta(z)} \psi$$

$$u(x + \Delta x, z) = \mathcal{P}_\theta u(x, z)$$

- Параметризованный псевдодифференциальный оператор

$$\mathcal{P}_\theta u = \exp \left(ik \Delta x \left(\sqrt{\frac{1}{k^2} \frac{\partial^2}{\partial z^2} + \text{MLP}_\theta(z)} - 1 \right) \right) u$$

Автоматическое дифференцирование численной схемы

- Traditional ML optimization complexity
 - ▶ Neural architecture design
 - ▶ Large training datasets
- Our approach complexity
 - ▶ Numerical scheme in loss function

$$\text{Loss} = \underbrace{\|\mathcal{G}(n) - v\|^2}_{\text{Numerical scheme}} + \underbrace{\gamma \|\nabla n\|^2}_{\text{Regularization}}$$

- Differentiation through PDE solver
 - ▶ Represent numerical PDE solver as a computational graph
 - ▶ JAX - Autodiff of the numerical implementation of the direct problem
 - ▶ Adam optimizer

Numerical results

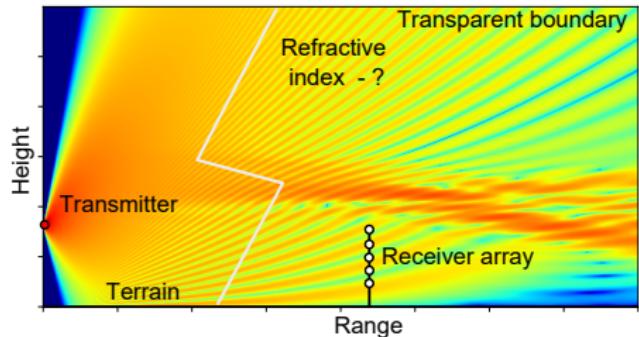


Рис.: Radio wave propagation setup

- 3 GHz source
- 17 receivers (5-170 m)
- 5 km range
- Signal-to-noise ratio 30 dB

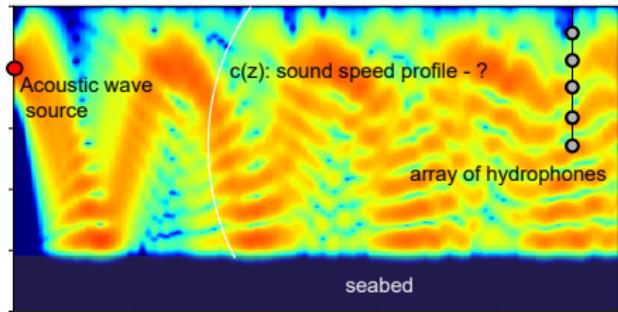


Рис.: Underwater acoustics setup

- 200 Hz source
- 15 hydrophones
- 3 km range
- Signal-to-noise ratio 30 dB

Numerical results

Tropospheric Inversion

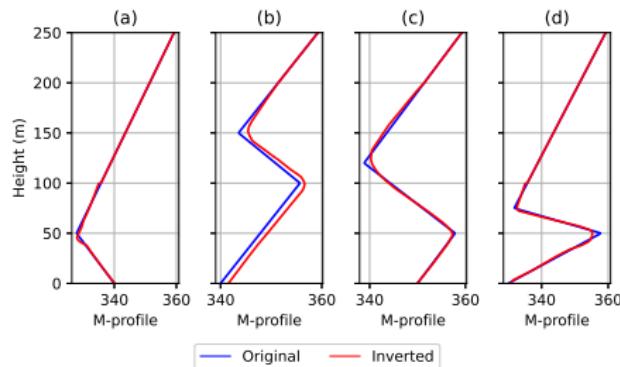


Рис.: True vs. inverted refractivity profiles

SSP Inversion

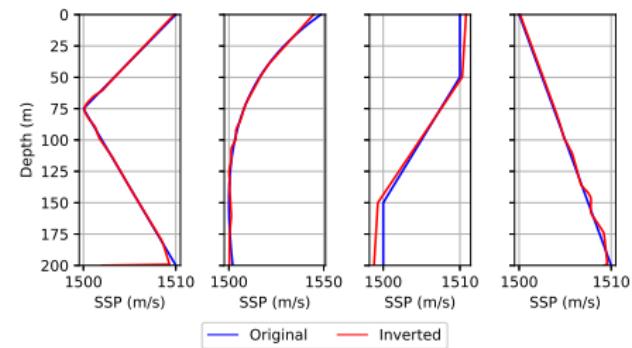


Рис.: Sound speed profile (SSP) inversion

- True and inverted refractivity functions coincide perfectly
- For the first time in these problems it was possible to invert the refraction on such a small volume of initial data
- A priori information about the distribution is not required

Обучение DeepONet для инверсии тропосферной рефракции

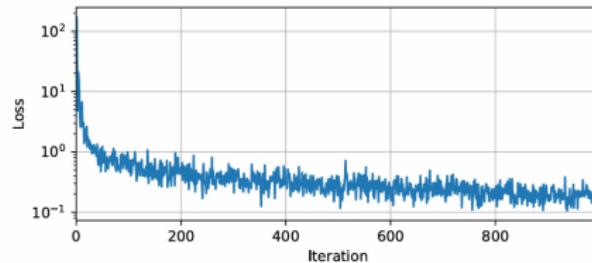


Figure 5. Loss vs iteration.

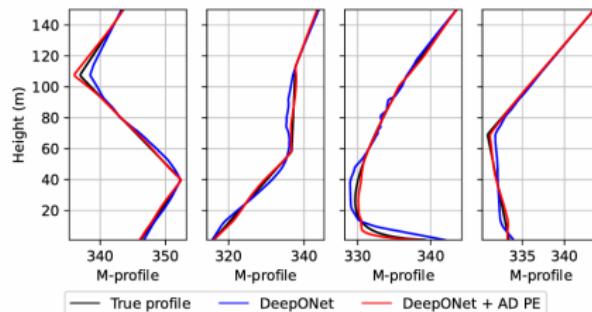


Figure 6. Comparison of true and inverted profiles. Marine surface, $R = 10$ km.

Дальнейшее развитие

- Теоретическое обоснование
- Применение к реальным задачам
- Оптимизация архитектур и процесса обучения (AutoML)