A Hands-on Introduction to Physics-informed Machine Learning

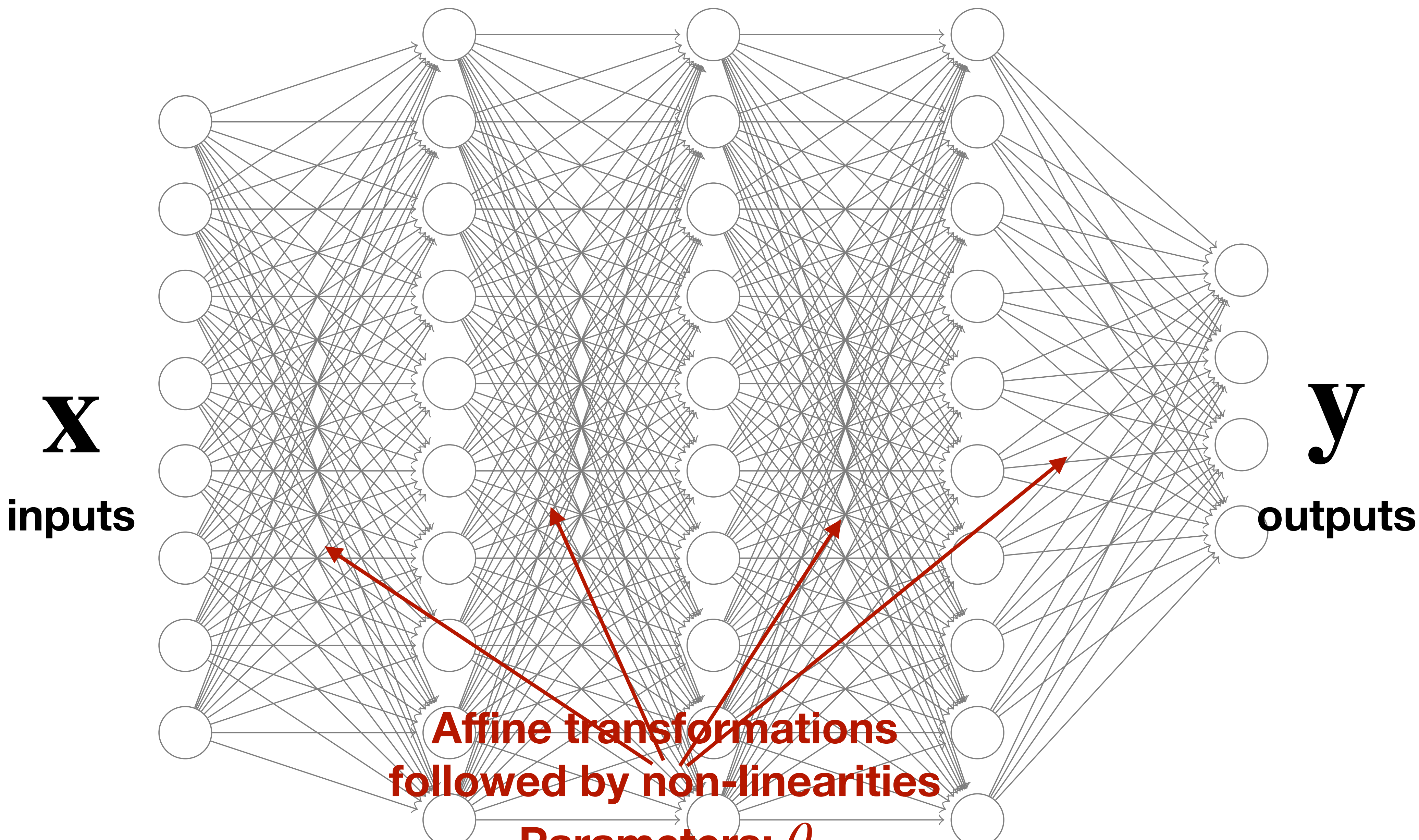
Ilias Bilionis, Atharva Hans,
Predictive Science Laboratory
School of Mechanical Engineering
Purdue University
May 26, 2021

PREDICTIVE
SCIENCE LABORATORY

Objective

Learn how physical information in the form of differential equations can be used to regularize neural networks.

Reminder - What are neural networks?



Parameters: θ

Function: $y = N(x;\theta)$

Reminder - How do we train neural networks?

- Depends on what the task is… Focusing on **regression**.
- You have some data consisting of **inputs** $x_{1:n} = (x_1, \ldots, x_n)$ and **outputs** $y_{1:n} = (y_1, \ldots, y_n)$.
- Say you want to find a neural net $N(x;\theta)$ that goes from input to output.
- Minimize a loss function:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - N(x_i;\theta)\right]^2$$

PREDICTIVE
SCIENCE LABORATORY

Reminder - How do we train neural networks?

Automatic differentiation for getting gradients (PyTorch, TensorFlow)

- Minimize a loss function:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - N(x_i;\theta)\right]^2$$

- We typically use a form of **stochastic gradient descent**

$$\theta_{t+1} = \theta_t - a_t \frac{1}{m}\sum_{j=1}^{m}\nabla_\theta\left[y_{i_j} - N(x_{i_j};\theta_t)\right]^2$$

Learning rate which has to satisfy certain constraints (Robbins-Monro, 1951)

Randomly sampled batch of inputs/outputs

PREDICTIVE SCIENCE LABORATORY

Illustrative Example 1: Solving an ODE

I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial Neural Networks for Solving Ordinary and Partial Differential Equations, 1997

From ODE to a loss function

- Consider the initial value problem:

$$\frac{d\Psi}{dx} = f(x, \Psi)$$

$$\Psi(0) = A$$

- Automatically satisfy the initial condition by parameterizing the solution as:

$$\hat{\Psi}(x;\theta) = A + xN(x;\theta)$$

- The idea is to find θ by minimizing the *integrated squared residual* of the ODE:

$$L(\theta) = \int_0^1 \left[\frac{d\hat{\Psi}(x;\theta)}{dx} - f(x, \hat{\Psi}(x;\theta))\right]^2 dx$$

Solving the problem with stochastic gradient descent

- The idea is to find θ by minimizing the *integrated squared residual* of the ODE:

$$L(\theta) = \int_0^1 \left[\frac{d\hat{\Psi}(x;\theta)}{dx} - f(x, \hat{\Psi}(x;\theta))\right]^2 dx$$

- The following algorithm converges (Robbins-Monro, 1951)

$$\theta_{t+1} = \theta_t - \frac{a_t}{n}\sum_{i=1}^{n} \nabla_\theta \left[\frac{d\hat{\Psi}(x_i;\theta_t)}{dx} - f(x_i, \hat{\Psi}(x_i;\theta_t))\right]^2$$

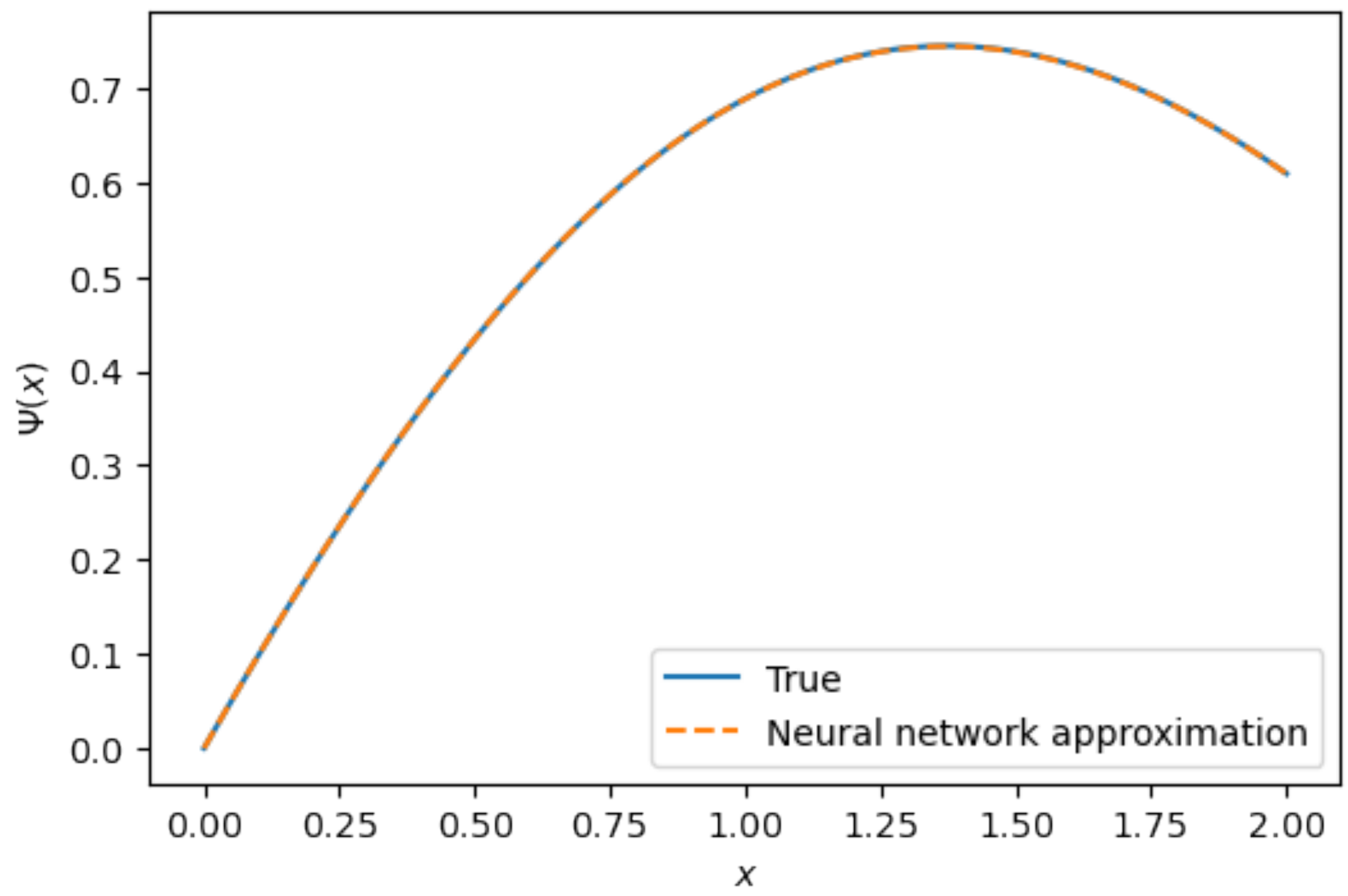
Spatial locations uniformly sampled in [0, 1] at each iteration.

PREDICTIVE
SCIENCE LABORATORY

Results (Part of Hands-on activity)

$$\frac{d\Psi(x)}{dt} = \exp\left\{-\frac{x}{5}\right\}\cos x - \frac{\Psi(x)}{5}$$

$$\Psi(0) = 0$$



PREDICTIVE
SCIENCE LABORATORY

Illustrative Example 2:
Solving an elliptic PDE

From PDEs to a loss function - Integrated squared approach

$$-\nabla \cdot \left[a(x)\nabla u(x)\right] + c(x)u(x) = f(x)$$

$$L(\theta) = \int \left\{ \nabla \cdot \left[a(x)\nabla \hat{u}(x;\theta)\right] + c(x)\hat{u}(x;\theta) + f(x) \right\}^2 dx$$

M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving non-linear partial differential equations, 2019

From PDEs to a loss function - Energy approach

$$-\nabla \cdot \left[a(x)\nabla u(x)\right] + c(x)u(x) = f(x)$$

Dirichlet principle

$$L(\theta) = \int \left\{ \frac{1}{2}a(x)\nabla \hat{u}(x;\theta) + c(x)\hat{u}^2(x;\theta) - f(x)\hat{u}(x;\theta) \right\} dx - \int_{\Gamma_N} g_N \hat{u}(x;\theta) d\Gamma_N$$

S. Karumuri, R. Tripathy, I. Bilionis, Simulator-free Solution of High-dimensional Elliptic Partial Differential Equations using Deep Neural Networks, 2020

PREDICTIVE
SCIENCE LABORATORY

I can already solve ODEs/ PDEs. Why is this useful?

Illustrative Example 3: Solving PDEs for all possible parameterizations

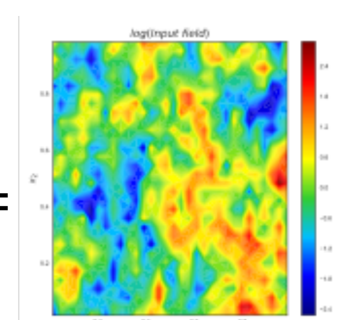
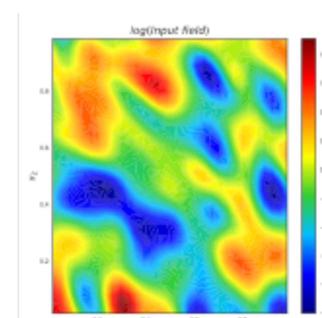
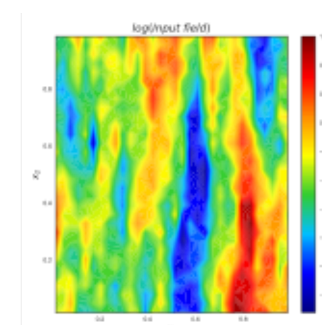
Random parameters

$$-\nabla \cdot \left[a(x;\xi)\nabla u(x;\xi)\right] + c(x)u(x;\xi) = f(x)$$

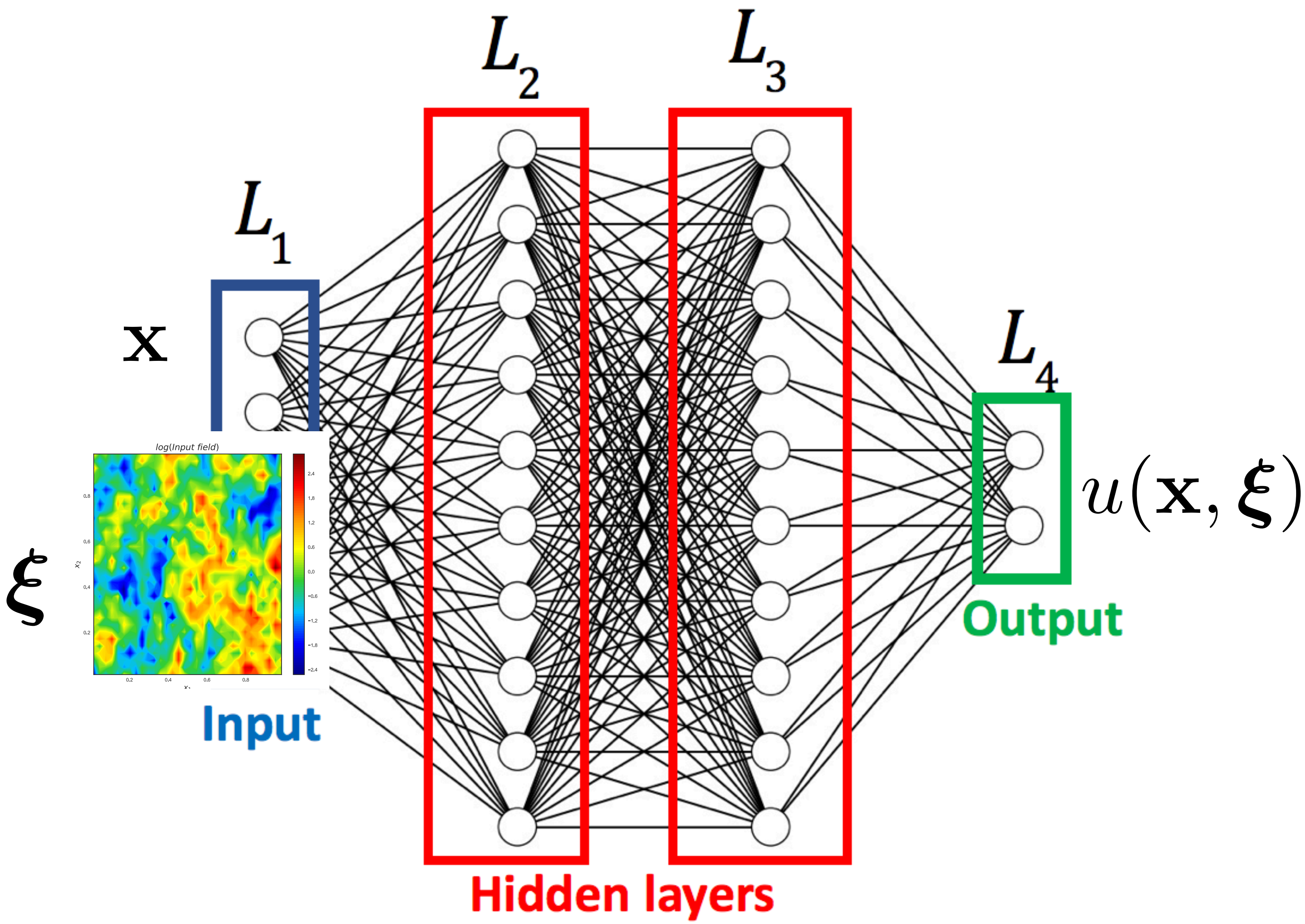
$$u = 0, \forall x_1 = 1,$$

$$u = 1, \forall x_1 = 0,$$

$$\frac{\partial u}{\partial n} = 0, \forall x_2 = 1.$$

$\log a(x,\xi) =$  or  or  ...

Representing the solution of the PDE with a DNN



PREDICTIVE
SCIENCE LABORATORY

From PDEs to a loss function - Energy approach

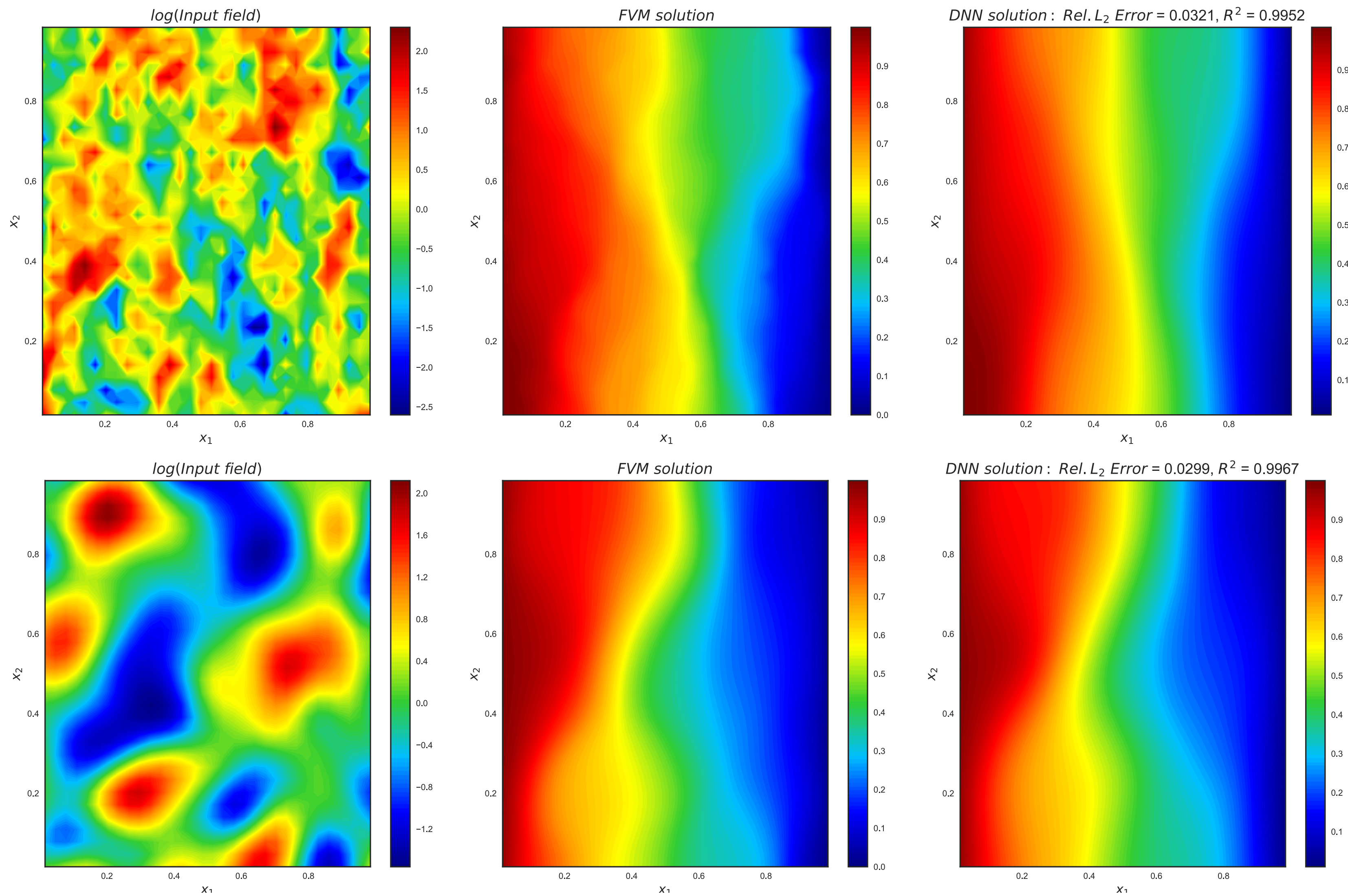
$$-\nabla \cdot \left[a(x;\xi)\,\nabla u(x;\xi)\right] + c(x)u(x;\xi) = f(x)$$

Dirichlet principle

$$L(\theta) = \mathbb{E}_{\xi}\left[\int \left\{ \frac{1}{2}a(x,\xi)\,\nabla\hat{u}(x,\xi;\theta) + c(x)\hat{u}^2(x,\xi;\theta) - f(x)\hat{u}(x,\xi;\theta) \right\} dx - \int_{\Gamma_N} g_N\hat{u}(x,\xi;\theta)d\Gamma_N]\right]$$

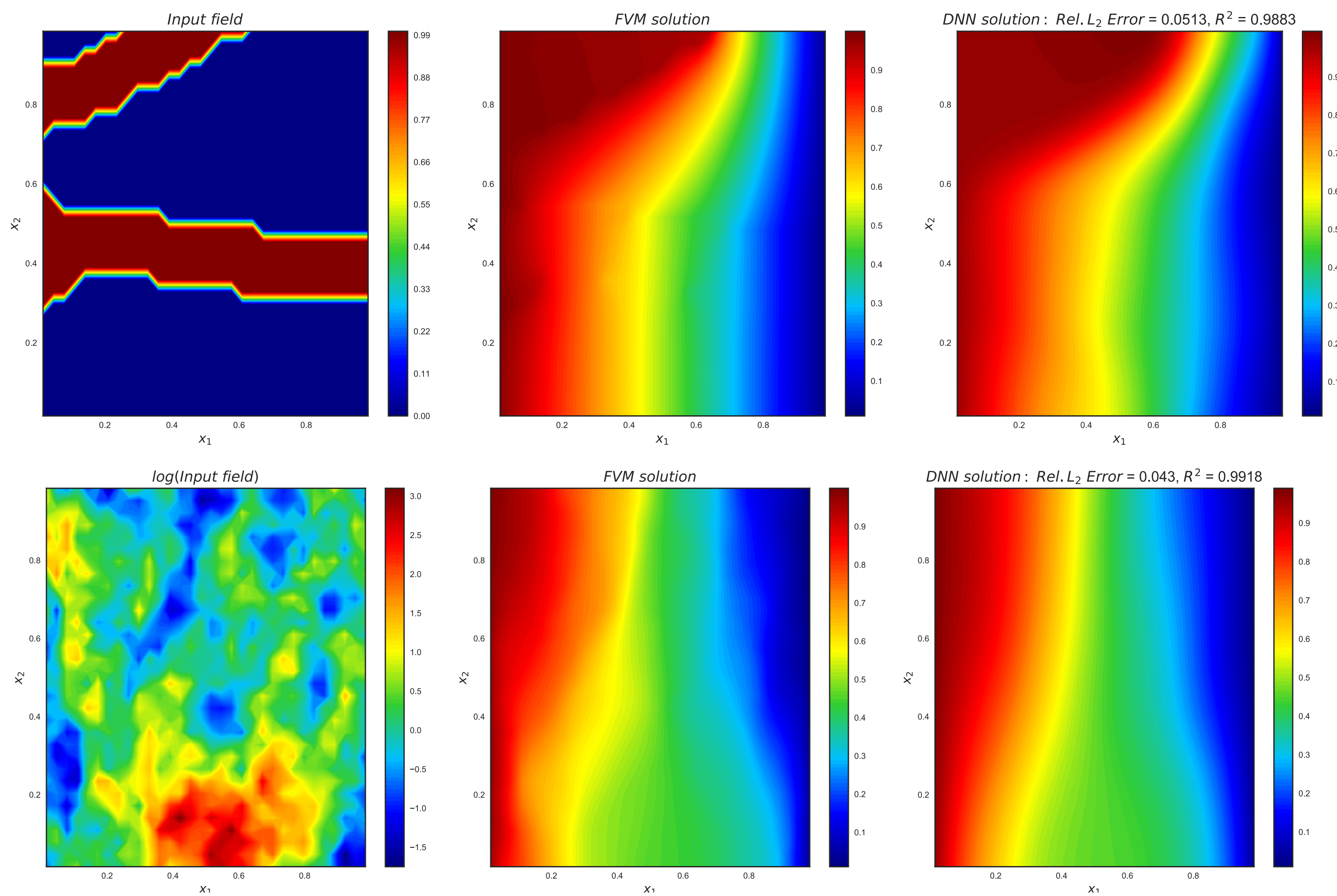
S. Karumuri, R. Tripathy, I. Bilionis, Simulator-free Solution of High-dimensional Elliptic Partial Differential Equations using Deep Neural Networks, 2020

One network for all kinds of random fields



Karumuri, S.; Tripathy, R.; Bilionis, I.; Panchal, J. Simulator-Free Solution of High-Dimensional Stochastic Elliptic Partial Differential Equations Using Deep Neural Networks. *Journal of Computational Physics* **2020**, *404*, 109120. https://doi.org/10.1016/j.jcp.2019.109120.

One network for all kinds of random fields



Karumuri, S.; Tripathy, R.; Bilionis, I.; Panchal, J. Simulator-Free Solution of High-Dimensional Stochastic Elliptic Partial Differential Equations Using Deep Neural Networks. *Journal of Computational Physics* **2020**, *404*, 109120. https://doi.org/10.1016/j.jcp.2019.109120.

What are the applications of this?

- High-dimensional uncertainty propagation through PDEs.
- Solving free boundary and Stefan problems (Wang, Perdikaris, 2021).
- PDE-constrained optimization (Hennigh et al., 2020).
- Inverse/model calibration problems (Raise, 2019).
- Data assimilation/filtering (no one yet).
- …

PREDICTIVE
SCIENCE LABORATORY

What is the catch?

- Not as easy as it looks in practice…
- Vanishing gradients…
- Spectral bias of deep nets…
- Fine solution features…
- We address some of these in the hands-on activity by solving the physical equations for a compressible neo-Hookean material.

Hands-on activity led by Atharva Hans

https://nanohub.org/tools/handsonpinns

PREDICTIVE
SCIENCE LABORATORY