Deep learning for high-dimensional uncertainty quantification

Kunal Kanawade[1] **Nick Dexter**[1]

[1]Department of Scientific Computing, Florida State University

Abstract

Most engineering and science problems require us to solve complex partial differential equations (PDE). Optimizing for a particular parameter for a physical system can be computationally expensive due to repetitive computations. Deep learning techniques have worked well with data with large dimensions[1]. Hence, we propose to use deep neural network architecture to learn the mapping between the parameter and function spaces. In this research work, we would like to focus on formulating Deep neural learning high-dimensional functions with deep neural network architecture. This is ongoing research work, and more results are in progress.

Introduction

In this work, the primary objective is to demonstrate that deep neural networks can be an effective approach in learning the mapping from the parameter space to solution space. For example, consider the following parametric PDE:

$$-\nabla_x(a(x,y),\nabla_x u(x,y)) = F(x),\ x \in \Omega \qquad (1)$$
$$u(x,y) = 0,\ \text{on}\ \partial\Omega$$

The DNN architecture is employed in this work to approximate the parametric mapping to the solution space $y \mapsto u(\cdot, y)$.

Challenges

- The targeted function to be approximated is high-dimensional, i.e. 'curse of dimensionality'.
- It is computationally expensive to generate the data, requiring solution of a PDE to obtain each sample $u(\cdot, y_i)$
- The data can be corrupted with errors, e.g. discretization, measurement, and/or numerical errors from the solver
- The solution map $y \mapsto u(\cdot, y)$ takes up values typically in Hilbert or Banach spaces, i.e. infinite-dim. function spaces

Methodology

In this work we consider training a DNN as a surrogate for the parameter to solution map for an underlying parametric PDE model. Our goal is to learn a high dimensional function:

$$f : \mathcal{U} \mapsto \mathcal{V},\ y \mapsto u(\cdot, y)$$

with $\mathcal{U} \subset \mathbb{R}^d$, $d \in \mathbb{N}$ and $\mathcal{V}$ is the solution space of the parameterized PDE, from samples $\{(y_i, u(\cdot, y_i))\}_{i=1}^m$.

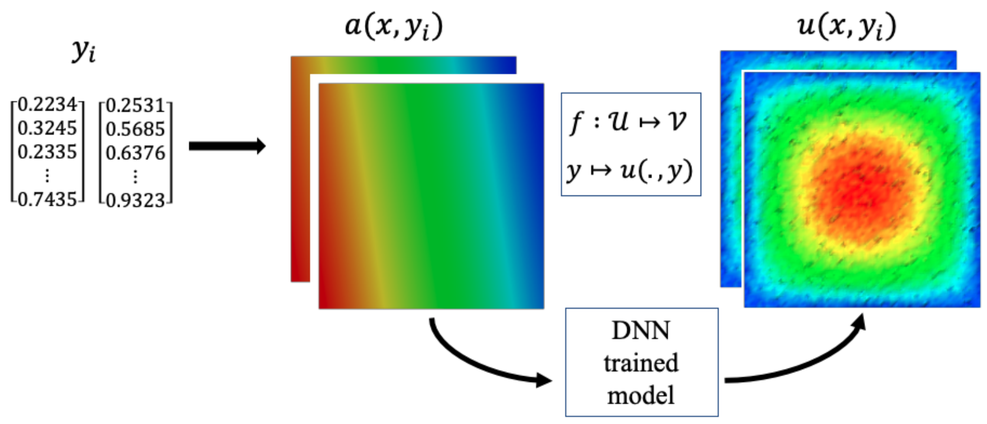

Figure 1: Schematic for the methodology used in the current work to map the parameter space to solution space

Results

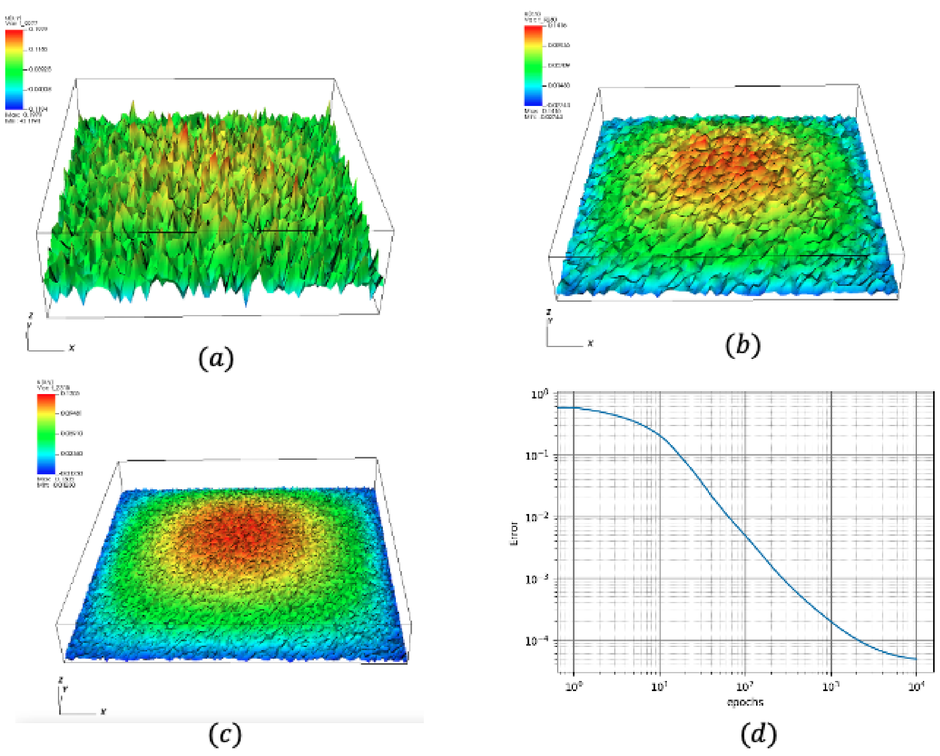

Figure 2: Comparison of the output of the DNN after training for (a) 1000 epochs, (b) 5000 epochs, (c) 10000 epochs, and (d) plot of the loss vs. epochs of training.

Discussion

It was shown in [2] that DNNs can approximate high-dimension smooth functions with error

$$\|f - \hat{f}\|_{L^2_\varrho(\mathcal{U};\mathcal{V})} \lesssim E_{\text{app}} + m^\theta(E_{\text{disc}} + E_{\text{samp}} + E_{\text{opt}})$$

where $\theta = 0$ if $\mathcal{V}$ is a Hilbert space, or $\theta = 1/4$ if $\mathcal{V}$ is a Banach space and

$$E_{\text{app}} \lesssim c_h \cdot (m/L)^{-\sigma(p)}, \qquad E_{\text{samp}} = \sqrt{\tfrac{1}{m}\textstyle\sum_{i=1}^m \|n_i\|^2_{\mathcal{V}}},$$
$$E_{\text{disc}} = \|f - \mathcal{P}_h(f)\|_{L^\infty_\varrho(\mathcal{U};\mathcal{V})},$$

where $\sigma(p) > 0$ is given by $\sigma(p) = \frac{1}{p} - \frac{1}{2}$ if $\mathcal{V}$ is a Hilbert space, or if $\mathcal{V}$ is a Banach space, $\sigma(p) = p - 1$ (known anisotropy) or $\sigma(p) = \frac{1}{2}(\frac{1}{p} - \frac{1}{2})$ (unknown anisotropy). Here

- E_{app} (approximation error): depends on m and smoothness of the parameter to solution mapping
- E_{disc} (discretization error): since we work in a finite-dimensional subspace $\mathcal{V}_h$
- E_{samp} (measurement error): quantifies the error in the point-wise evaluations
- E_{opt} (optimization error): depends on training, proportional to loss function error

References

[1] Dexter, N., Adcock, B., Brugiapaglia, S., Moraga, S. (2022, October). Effective deep neural network architectures for learning high-dimensional Banach-valued functions from limited data. In 2022 Fall Southeastern Sectional Meeting. AMS.

[2] Adcock B., Brugiapaglia S., Dexter N., Moraga S., "Near-optimal learning of Banach-valued, high-dimensional functions via deep neural networks." *arXiv preprint arXiv:2211.12633* (2022).