Optimal deep learning of holomorphic operators between Banach spaces

Sebastian Moraga Scheuermann
smoragas@sfu.ca
sites.google.com/view/sebanthalas

November 11, 2024

Joint work with

Ben Adcock

SFU

Nick Dexter

FLORIDA STATE UNIVERSITY

Outline

Motivating problem

Noisy training data

We want to capture the dynamic behavior of holomorphic operators using surrogate models based on DNNs, i.e., to approximate

$$X \in \mathcal{X} \mapsto F(X) \in \mathcal{Y}$$

where $\mathcal{Y}$ is the PDE solutions space and X represents the data supplied to the PDE. Let μ be a probability measure on $\mathcal{X}$. Then the noisy training data is given by

$$\{(X_i, F(X_i) + E_i)\}_{i=1}^{m},$$

where $X_1, \ldots, X_m \sim_{\text{i.i.d.}} \mu$ and E_i is noise.

Keywords: uncertainty quantification, surrogate models, parametric PDEs, Deep Learning.

Motivating problem

We focus on learning holomorphic operators.

The typical operator learning methodology

Consists of three objects: an approximate encoder $\mathcal{E}_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}^{d_{\mathcal{X}}}$, an approximate decoder $\mathcal{D}_{\mathcal{Y}} : \mathbb{R}^{d_{\mathcal{Y}}} \to \mathcal{Y}$ and a DNN $\hat{N} : \mathbb{R}^{d_{\mathcal{X}}} \to \mathbb{R}^{d_{\mathcal{Y}}}$, which approximates F as

$$F \approx \hat{F} := \mathcal{D}_{\mathcal{Y}} \circ \hat{N} \circ \mathcal{E}_{\mathcal{X}}.$$

The encoder and decoder are either specified by the problem, learned separately from data, or learned concurrently with $\hat{N}$. The goal, as in all supervised learning problems, is to ensure good generalization via the learned operator $\hat{F}$ from as little training data m as possible.

Theorem [BA, ND, SM (2024)] (Upper bounds)

Let $0 < p, \epsilon < 1$, $m \geq 3$ and $\varepsilon > 0$. Then there is a class $\mathcal{N}$ of hyperbolic tangent DNN depending on m and ϵ only such that the following holds. Provided a technical assumption holds, with high probability, every approximate minimizer of the training problem above satisfies

$$|||F - \hat{F}|||_{L^2_\mu(\mathcal{X};\mathcal{Y})} \lesssim E_{\text{app},2} + E_{\mathcal{X},2} + E_{\mathcal{Y},2} + E_{\text{opt},2} + E_{\text{samp},2},$$

$$\|F - \hat{F}\|_{L^\infty_\mu(\mathcal{X};\mathcal{Y})} \lesssim E_{\text{app},\infty} + E_{\mathcal{X},\infty} + E_{\mathcal{Y},\infty} + E_{\text{opt},\infty} + E_{\text{samp},\infty},$$

where $\widetilde{m} = m/(\log(m) + \log\left(\epsilon^{-1}\right))$ and E_{opt} is the objective function error.

Here, for $q \in \{2, \infty\}$

- $E_{\text{app},q}$ is an *approximation error*, which decays algebraically in the amount of training data m.
- $E_{\mathcal{X},q}$, $E_{\mathcal{Y},q}$ are *encoding-decoding errors*, which depend on the accuracy of the learned encoders and decoders.
- $E_{\text{opt},q}$ is an *optimization error*, and $E_{\text{samp},q}$ is a *sampling error*, which depends on the noise E_i.

Theoretical contributions

The main theoretical contributions of this work are as follows

1. We consider operators taking values in general Banach spaces.
2. We consider standard feedforward DNN architectures (constant width, width exceeds depth) and training procedures (ℓ^2-loss minimization).
3. We construct a family of DNNs such that any approximate minimizer of the corresponding training problem satisfies a generalization bound that is explicit in the various error sources.
4. These DNN architectures are *problem agnostic*; they depend on m only. In particular, the architectures are completely independent on the regularity assumptions of target operator.
5. We show that training problems based on *any* family of fully-connected DNNs possess uncountably many minimizers that achieve the same generalization bounds.
6. We provide bounds in both the L^2_μ- and L^∞_μ-norms that hold in high probability, rather than just expectation.
7. We show that the generalization bound is optimal with respect to m: no learning procedure (not necessarily DL-based) can achieve better rates in m up to log terms.

Outline

Training data and design of experiments

We run several trials solving the problem

Given training data $\{(X_i, Y_i)\}_{i=1}^m \subset (\mathcal{X} \times \mathcal{Y})^m$, $X_i \sim_{\text{i.i.d.}} \mu$, $Y_i = F(X_i) + E_i \in \mathcal{Y}$, approximate $F \in L^2_\mu(\mathcal{X}; \mathcal{Y})$.

1. We generate the measurements Y_i using mixed variational formulations of the parametric elliptic PDEs discretized using `FEniCS` with input data X_i.
2. The noise $E_i \in \mathcal{Y}$ encompasses the discretization errors from numerical solution.
3. Each of our architectures is trained across a range of datasets with increasing sizes. This involves using a set of training data consisting of values $\{(X_i, Y_i))\}_{i=1}^m$, where $m \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500\}$.
4. After training we calculate the testing error for each trial and run statistics across all trials for each dataset.

Choice of architectures and initialization

We fix the number of nodes per layer N and depth L such that the ratio $\beta := L/N$ is $\beta = 0.5$. We initialize the weights and biases using the `HeUniform` initializer from `keras` setting the seed to the trial number. We consider the Rectified Linear Unit (ReLU)

$$\sigma_1(z) := \max\{0, z\},$$

hyperbolic tangent (tanh)

$$\sigma_2(z) := \frac{e^z - e^{-z}}{e^z + e^{-z}},$$

or Exponential Linear Unit (ELU)

$$\sigma_3(z) = \begin{cases} z & z > 0, \\ e^z - 1 & z \leq 0 \end{cases}$$

activation functions in our experiments.

Implementation

We use the open-source finite element library `FEniCS`, specifically version 2019.1.0, and Google's `TensorFlow` version 2.12.0.

Hardware

We train the DNN models in single precision on the Digital Research Alliance of Canada's Cedar compute cluster, using Intel Xenon Processor E5-2683 v4 CPUs with either 125GB or 250GB per node. Results were stored locally on the cluster and the estimated total space used to store the data for testing and training and results from computation is approximately 50 GB.

Experiments

For each experiment we consider training with 14 sets of points of size $m \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500\}$ and for 6 different architectures (4 x 40 and 10 x 100 with ReLU, ELU, and tanh activations) over two parametric dimensions ($d = 4$ and $d = 8$) and two coefficients giving 336 DNNs to be trained for each trial.

Outline

Parametric Boussinesq problem

Given $\boldsymbol{x} \in [-1,1]^d$, find the velocity $\boldsymbol{u} : [-1,1]^d \times \Omega \to \mathbb{R}^2$, pressure $p : [-1,1]^d \times \Omega \to \mathbb{R}$ and temperature $\varphi : [-1,1]^d \times \Omega \to \mathbb{R}$ of a fluid such that

$$\begin{aligned}
-\mathbf{div}(2a(\boldsymbol{x})\mu(\varphi(\boldsymbol{x}))\boldsymbol{e}(\boldsymbol{u}(\boldsymbol{x}))) + (\boldsymbol{u}(\boldsymbol{x}) \cdot \nabla)\boldsymbol{u}(\boldsymbol{x}) + \nabla p(\boldsymbol{x}) &= \varphi(\boldsymbol{x})\boldsymbol{g}, \quad \text{in } \Omega, \\
\mathrm{div}(\boldsymbol{u}(\boldsymbol{x})) &= 0, \quad \text{in } \Omega, \\
-\mathrm{div}(\mathbb{K}(\boldsymbol{x})\nabla\varphi(\boldsymbol{x})) + \boldsymbol{u}(\boldsymbol{x}) \cdot \nabla\varphi(\boldsymbol{x}) &= 0, \quad \text{in } \Omega, \\
\boldsymbol{u} &= \boldsymbol{u}_D, \ \text{on } \partial\Omega, \\
\varphi &= \varphi_D, \ \text{on } \partial\Omega, \\
\int_\Omega p(\boldsymbol{x}) &= 0.
\end{aligned}$$

We consider to approximate

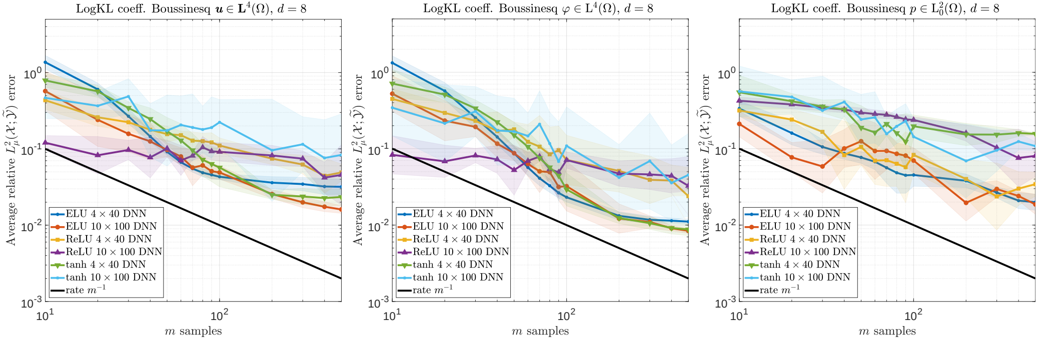
$$\boldsymbol{x} \in [-1,1]^d \mapsto (\boldsymbol{u}, p, \varphi)(\boldsymbol{x}) \in (\mathbf{L}^4(\Omega) \times \mathrm{L}^2_0(\Omega) \times \mathrm{L}^4(\Omega))$$

of a fully-mixed variational formulation in Banach spaces.

Colmenares, Gatica, Moraga (2020).

Parametric PDE approximation in Banach spaces

- Steady-state parametric Boussinesq equations with physical domain $(0,1)^3$ and $d=8$.
- Comparison of testing error in $L^2_\varrho([-1,1]^d;\mathbf{L}^4(\Omega))$, $L^2_\varrho([-1,1]^d;\mathrm{L}^4(\Omega))$ and $L^2_\varrho([-1,1]^d;\mathrm{L}^2(\Omega))$ for $(\boldsymbol{u},\varphi,p)$.



Outline

Conclusions

- We show sharp algebraic rates of convergence in m, confirming that certain classes of holomorphic operators involving PDEs can be learned efficiently and without the *curse of dimensionality*.

- The sizes of the various DNNs in our theorems also do not succumb to the so-called *curse of parametric complexity*, since the width and depth bounds are at most algebraic in m.

- We present a series of experiments demonstrating the efficacy of DL on challenging problems such as the parametric diffusion, Navier-Stokes-Brinkman and Boussinesq PDEs, the latter two of which involve operators whose codomains are Banach, as opposed to Hilbert, spaces.

References

B. Adcock; S. Brugiapaglia; N.Dexter; S. Moraga, Near-optimal learning of Banach-valued, high-dimensional functions via deep neural networks and deep learning. Neural Networks (in press), 2024

B. Adcock.; N.Dexter.; S. Moraga, Optimal approximation of infinite-dimensional holomorphic functions. Calcolo, 61(1):12, 2024.

B. Adcock.; S. Brugiapaglia.; N.Dexter.; S. Moraga, Learning smooth functions in high dimensions: from sparse polynomials to deep neural networks. In S. Mishra and A. Townsend, editors, Numerical Analysis Meets Machine Learning, volume 25 of Handbook of Numerical Analysis, pages 1–52. Elsevier, 2024.

B. Adcock.; N.Dexter.; S. Moraga, Optimal approximation of infinite-dimensional holo- morphic functions II: recovery from i.i.d. pointwise samples. arXiv:2310.16940, 2023.

B. Adcock.; S. Brugiapaglia.; N.Dexter.; S. Moraga, Deep Neural Networks Are Effective At Learning High-Dimensional Hilbert-Valued Functions From Limited Data. MSML, volume 145, pages 1–36. (2021)

smoragas@sfu.ca
sites.google.com/view/sebanthalas