On Universal Approximation of set-valued maps and Deep operator network approximation of the controllability map

Pablo Pedregal

Universidad de Castilla- La Mancha
https://multisimo.com/fpe/
joint work with Carlos García-Cervera (UCSB, USA); Mathieu Kessler (UPCT, Spain) and Paco Periago (UPCT, Spain)

2th meeting of the network COPI

Almagro, December, 2024

Problem setup

Problem setup

Consider the control system

$$\begin{cases} y'(t) = Ay(t) + Bu(t), & t > 0 \\ y(0) = y^0 \in X, \end{cases} \tag{1}$$

where

- $A : D(A) \subset X \to X$ is a linear operator,
- $y = y(t) \in D(A)$ is the state variable,
- $u = u(t) \in Y$ is the control variable, and
- B is an admissible control operator

Problem setup

Consider the control system

$$\begin{cases} y'(t) = Ay(t) + Bu(t), & t > 0 \\ y(0) = y^0 \in X, \end{cases} \tag{1}$$

where

- $A : D(A) \subset X \to X$ is a linear operator,
- $y = y(t) \in D(A)$ is the state variable,
- $u = u(t) \in Y$ is the control variable, and
- B is an admissible control operator

Given a time $T > 0$ and a target state $y^T \in X$, the exact controllability problem for system (1) amounts to finding a control $u \in Y$ such that its associated state $y = y(u)$ satisfies

$$y(T) = y^T. \tag{2}$$

Problem setup

Consider the control system

$$\begin{cases} y'(t) = Ay(t) + Bu(t), & t > 0 \\ y(0) = y^0 \in X, \end{cases} \tag{1}$$

where

- $A : D(A) \subset X \to X$ is a linear operator,
- $y = y(t) \in D(A)$ is the state variable,
- $u = u(t) \in Y$ is the control variable, and
- B is an admissible control operator

Given a time $T > 0$ and a target state $y^T \in X$, the exact controllability problem for system (1) amounts to finding a control $u \in Y$ such that its associated state $y = y(u)$ satisfies

$$y(T) = y^T. \tag{2}$$

Goal: Learn, from a dataset, the controllability map associated with (1)-(2)

$$\mathcal{G} : X \to Y, \quad y^0 \mapsto \mathcal{G}(y^0) := u.$$

Problem setup

First difficulty: In general, $\mathcal{G}$ **is a set-valued map**, meaning that problem (1)-(2) may have many solutions.

Problem setup

First difficulty: In general, $\mathcal{G}$ **is a set-valued map**, meaning that problem (1)-(2) may have many solutions.

Important question: Is there a selection (single-valued map) G of $\mathcal{G}$ such that $G(y_j^0) = u_j(t; y_j^0)$? In such a case, how smooth is this selection?

Problem setup

First difficulty: In general, $\mathcal{G}$ **is a set-valued map**, meaning that problem (1)-(2) may have many solutions.
Important question: Is there a selection (single-valued map) G of $\mathcal{G}$ such that $G(y_j^0) = u_j(t; y_j^0)$? In such a case, how smooth is this selection?
Answer: YES, there is a Lipschitz continuous selection, and moreover:

Corollary

Under the same hypotheses as in the theorem below, let $K \subset X$, and $f : K \to Y$, a continuous single-valued map with $f(x) \in F(x)$ for every $x \in K$. Then f can be extended to a continuous selection defined on the whole space X. **In particular, if $K \subset X$ is finite, and $f : K \to X$ is an arbitrary map with $f(x) \in F(x)$ for every $x \in K$, then there is a continuous selection defined on the whole space X "interpolating" f.**

Problem setup

First difficulty: In general, $\mathcal{G}$ **is a set-valued map**, meaning that problem (1)-(2) may have many solutions.
Important question: Is there a selection (single-valued map) G of $\mathcal{G}$ such that $G(y_j^0) = u_j(t; y_j^0)$? In such a case, how smooth is this selection?
Answer: YES, there is a Lipschitz continuous selection, and moreover:

Corollary

Under the same hypotheses as in the theorem below, let $K \subset X$, and $f : K \to Y$, a continuous single-valued map with $f(x) \in F(x)$ for every $x \in K$. Then f can be extended to a continuous selection defined on the whole space X. **In particular, if $K \subset X$ is finite, and $f : K \to X$ is an arbitrary map with $f(x) \in F(x)$ for every $x \in K$, then there is a continuous selection defined on the whole space X "interpolating" f.**

Idea of the proof. We rely on the celebrated Michael's theorem

Theorem

Let F be a lower semicontinuous set-valued map with closed, convex values from a compact metric space X to a Banach space Y. It does have a continuous selection.

This is Th. 9.1.2 in

Single and set-valued maps

Idea of the proof (cont.). Consider the non-trivial closed, subspace

$$\mathbb{S} \equiv \mathcal{G}(0), \quad \mathbb{S} \subset L^2(0, T),$$

Idea of the proof (cont.). Consider the non-trivial closed, subspace

$$\mathbb{S} \equiv \mathcal{G}(0), \quad \mathbb{S} \subset L^2(0, T),$$

By linearity,

$$\mathcal{G}(y^0) = u^0 + \mathbb{S} \quad \text{for any } u^0 \in \mathcal{G}(y^0).$$

Hence, the values of $\mathcal{G}$ are closed and convex.

Single and set-valued maps

Idea of the proof (cont.). Consider the non-trivial closed, subspace

$$\mathbb{S} \equiv \mathcal{G}(0), \quad \mathbb{S} \subset L^2(0, T),$$

By linearity,

$$\mathcal{G}(y^0) = u^0 + \mathbb{S} \quad \text{for any } u^0 \in \mathcal{G}(y^0).$$

Hence, the values of $\mathcal{G}$ are closed and convex.
Let $\pi : L^2(0, T) \mapsto \mathbb{S}^\perp$ be the orthogonal projection onto the orthogonal complement of $\mathbb{S}$. Then,

$$\pi \circ \mathcal{G} : L^2(0,1) \mapsto L^2(0, T) \tag{3}$$

turns out to be single-valued and linear.

Idea of the proof (cont.). Consider the non-trivial closed, subspace

$$\mathbb{S} \equiv \mathcal{G}(0), \quad \mathbb{S} \subset L^2(0,T),$$

By linearity,

$$\mathcal{G}(y^0) = u^0 + \mathbb{S} \quad \text{for any } u^0 \in \mathcal{G}(y^0).$$

Hence, the values of $\mathcal{G}$ are closed and convex.
Let $\pi : L^2(0,T) \mapsto \mathbb{S}^\perp$ be the orthogonal projection onto the orthogonal complement of $\mathbb{S}$. Then,

$$\pi \circ \mathcal{G} : L^2(0,1) \mapsto L^2(0,T) \tag{3}$$

turns out to be single-valued and linear. In fact, since

$$\pi(y) = \text{argmin}_z \{\|z\| : z \in y + \mathbb{S}\},$$

the composition in (3) yields the control of minimal norm. Hence,

$$\|u\|_{L^2(0,T)} \leq C\|y_0\|_{L^2(0,1)},$$

which is equivalent to the continuity of such a projection π.

Idea of the proof (cont.). Consider the non-trivial closed, subspace

$$\mathbb{S} \equiv \mathcal{G}(0), \quad \mathbb{S} \subset L^2(0,T),$$

By linearity,

$$\mathcal{G}(y^0) = u^0 + \mathbb{S} \quad \text{for any } u^0 \in \mathcal{G}(y^0).$$

Hence, the values of $\mathcal{G}$ are closed and convex.
Let $\pi : L^2(0,T) \mapsto \mathbb{S}^\perp$ be the orthogonal projection onto the orthogonal complement of $\mathbb{S}$. Then,

$$\pi \circ \mathcal{G} : L^2(0,1) \mapsto L^2(0,T) \tag{3}$$

turns out to be single-valued and linear. In fact, since

$$\pi(y) = \operatorname{argmin}_z \{\|z\| : z \in y + \mathbb{S}\},$$

the composition in (3) yields the control of minimal norm. Hence,

$$\|u\|_{L^2(0,T)} \leq C\|y_0\|_{L^2(0,1)},$$

which is equivalent to the continuity of such a projection π.
The continuity of $\mathcal{G}$ is then a consequence of the identity

$$\mathcal{G}(y) = \pi \circ \mathcal{G}(y) + \mathbb{S}.$$

Numerical approximation of $\mathcal{G}$: Machine Learning setup

- **Dataset**
  We fix a set of **sensor points** $\{x_1, x_2, \cdots, x_m\} \subset [0,1]$. The information of each selected continuous initial datum y^0 is encoded in the vector

$$(y^0(x_1), y^0(x_2), \cdots, y^0(x_m)) \equiv y^{\text{initial}}$$

Numerical approximation of $\mathcal{G}$: Machine Learning setup

- **Dataset**

  We fix a set of **sensor points** $\{x_1, x_2, \cdots, x_m\} \subset [0, 1]$. The information of each selected continuous initial datum y^0 is encoded in the vector

$$(y^0(x_1), y^0(x_2), \cdots, y^0(x_m)) \equiv y^{\text{initial}}$$

  We also take $t_\ell \in [0, T]$. The corresponding **labels** are

$$\{u_\ell = u(y_\ell^{\text{initial}}; t_\ell), \quad 1 \le \ell \le N\},$$

  the control at time t_ℓ corresponding to the initial datum y_ℓ^{initial}.

Numerical approximation of $\mathcal{G}$: Machine Learning setup

- **Dataset**
  We fix a set of **sensor points** $\{x_1, x_2, \cdots, x_m\} \subset [0,1]$. The information of each selected continuous initial datum y^0 is encoded in the vector

$$(y^0(x_1), y^0(x_2), \cdots, y^0(x_m)) \equiv y^{\text{initial}}$$

  We also take $t_\ell \in [0, T]$. The corresponding **labels** are

$$\{u_\ell = u(y_\ell^{\text{initial}}; t_\ell), \quad 1 \le \ell \le N\},$$

  the control at time t_ℓ corresponding to the initial datum y_ℓ^{initial}.

- **Hypothesis space: the neural network**
  We will use the so-called **DeepONet**, which takes the form

$$\mathcal{N}(\boldsymbol{\theta}; (y^{\text{initial}}(x_j); t)) := \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k y^{\text{initial}}(x_j) + \theta_i^k \right) \cdot \sigma(w_k \cdot t + \eta_k)$$

  where $\boldsymbol{\theta} = (c_i^k, \xi_{ij}^k, \theta_i^k, w_k, \eta_k)$ is the set of parameters of the net, and σ, the activation function.

Numerical approximation of $\mathcal{G}$: Machine Learning setup

- **Dataset**
  We fix a set of **sensor points** $\{x_1, x_2, \cdots, x_m\} \subset [0,1]$. The information of each selected continuous initial datum y^0 is encoded in the vector

$$(y^0(x_1), y^0(x_2), \cdots, y^0(x_m)) \equiv y^{\text{initial}}$$

  We also take $t_\ell \in [0, T]$. The corresponding **labels** are

$$\{u_\ell = u(y_\ell^{\text{initial}}; t_\ell), \quad 1 \leq \ell \leq N\},$$

  the control at time t_ℓ corresponding to the initial datum y_ℓ^{initial}.

- **Hypothesis space: the neural network**
  We will use the so-called **DeepONet**, which takes the form

$$\mathcal{N}(\boldsymbol{\theta}; (y^{\text{initial}}(x_j); t)) := \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k y^{\text{initial}}(x_j) + \theta_i^k \right) \cdot \sigma(w_k \cdot t + \eta_k)$$

  where $\boldsymbol{\theta} = (c_i^k, \xi_{ij}^k, \theta_i^k, w_k, \eta_k)$ is the set of parameters of the net, and σ, the activation function.

- **Loss function (MSE)**

$$\text{loss}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{\ell=1}^{N} |\mathcal{N}(\boldsymbol{\theta}; (y_\ell^{\text{initial}}; t_\ell)) - u_\ell|^2$$

Numerical approximation of $\mathcal{G}$: Machine Learning setup

Deep Operator Network (DeepONet) as the prediction model

Numerical approximation of $\mathcal{G}$: Machine Learning setup

Deep Operator Network (DeepONet) as the prediction model

Branch net:

$$\beta_k(y^{\text{initial}}) = \sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{ij}^k y^{\text{initial}}(x_j) + \theta_i^k\right), \quad 1 \leq k \leq p$$

Numerical approximation of $\mathcal{G}$: Machine Learning setup

Deep Operator Network (DeepONet) as the prediction model

Branch net:

$$\beta_k(y^{\text{initial}}) = \sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k y^{\text{initial}}(x_j) + \theta_i^k \right), \quad 1 \leq k \leq p$$

Trunk net:

$$\tau_k(t) = \sigma(w_k \cdot t + \eta_k), \quad 0 \leq k \leq p$$

so that

$$\mathcal{N}(\boldsymbol{\theta}; (y^{\text{initial}}(x_j); t)) := \tau_0(t) + \sum_{k=1}^{p} \beta_k(y^{\text{initial}})\tau_k(t).$$

Numerical approximation of $\mathcal{G}$: Machine Learning setup

Deep Operator Network (DeepONet) as the prediction model

Branch net:

$$\beta_k(y^{\text{initial}}) = \sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{ij}^k y^{\text{initial}}(x_j) + \theta_i^k\right), \quad 1 \le k \le p$$

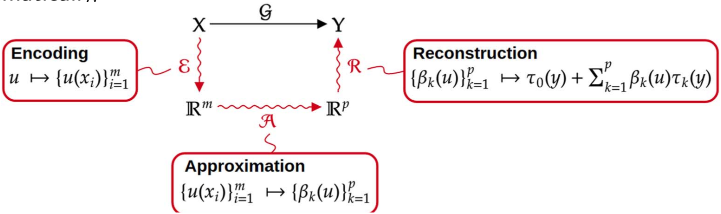
Trunk net:

$$\tau_k(t) = \sigma(w_k \cdot t + \eta_k), \quad 0 \le k \le p$$

so that

$$\mathcal{N}(\boldsymbol{\theta}; (y^{\text{initial}}(x_j); t)) := \tau_0(t) + \sum_{k=1}^{p} \beta_k(y^{\text{initial}})\tau_k(t).$$

Schematically,



Why DeepONet as the prediction model?

Why DeepONet as the prediction model?

Universal Approximation of functions, funcionals and operators

Why DeepONet as the prediction model?

Universal Approximation of functions, funcionals and operators

Why DeepONet as the prediction model?

Universal Approximation of functions, funcionals and operators

Theorem (UAT for functions (A. Pinkus))

Suppose that $K \in \mathbb{R}^d$ *is compact,* $U \subset C(K)$ *is compact, and* $\sigma(s) = \max\{s, 0\}$ *is the ReLU activation funcion. Then, for any* $\varepsilon > 0$ *there exist a positive integer* n*, real numbers* θ_i*,* $\omega_i \in \mathbb{R}^n$*, independent of* $f \in U$*, and constants* $c_i = c_i(f)$ *depending on* f*, such that*

$$|f(x) - \sum_{i=1}^{n} c_i \sigma(\omega_i \cdot x + \theta_i)| < \varepsilon$$

holds for all $x \in K$ *and* $f \in U$*. Moreover, each* $c_i(f)$ *is a continuous linear functional defined on* U*.*

Why DeepONet as the prediction model?

Universal Approximation of functions, funcionals and operators

Theorem (UAT for functions (A. Pinkus))

Suppose that $K \in \mathbb{R}^d$ is compact, $U \subset C(K)$ is compact, and $\sigma(s) = \max\{s, 0\}$ is the ReLU activation funcion. Then, for any $\varepsilon > 0$ there exist a positive integer n, real numbers θ_i, $\omega_i \in \mathbb{R}^n$, independent of $f \in U$, and constants $c_i = c_i(f)$ depending on f, such that

$$|f(x) - \sum_{i=1}^{n} c_i \sigma(\omega_i \cdot x + \theta_i)| < \varepsilon$$

holds for all $x \in K$ and $f \in U$. Moreover, each $c_i(f)$ is a continuous linear functional defined on U.

Theorem (UAT for functionals (Chen & Chen))

Suppose that $\sigma \in TW$, X is a Banach space, $K \subset X$ is a compact set, V is a compact set in $C(K)$, and $f : V \to \mathbb{R}$ is a continuous functional. Then for any $\varepsilon > 0$, there are a positive integer n, m sensor points $x_1, x_2, \cdots, x_m \in K$, and real constants c_i, θ_i, ξ_{ij}, $1 \leq i \leq n$, $1 \leq j \leq m$, such that

$$|f(u) - \sum_{i=1}^{n} c_i \sigma \left(\sum_{j=1}^{m} \xi_{ij} u(x_j) + \theta_i \right) | < \varepsilon, \quad \text{for all } u \in V.$$

Why DeepONet as the prediction model?

Universal Approximation of functions, funcionals and operators

Theorem (UAT for Borel single-valued measurable mappings (Lanthaler, Mishra, Karniadakis, 2022))

Let $\mu \in \mathcal{P}(C(D))$ *be a probability measure on* $C(D)$ *and let* $\mathcal{G} : C(D) \to L^2(U)$ *be a Borel measurable mapping, with* $\mathcal{G} \in L^2(\mu)$. *Then, for every* $\varepsilon > 0$, *there exists a* **DeepONet** $\mathcal{N} = \mathcal{R} \circ \mathcal{A} \circ \mathcal{E}$ *such that*

$$\|\mathcal{G} - \mathcal{N}\|_{L^2(\mu)} = \left(\int_X \|\mathcal{G}(u) - \mathcal{N}(u)\|^2_{L^2(U)} \, d\mu(u) \right)^{1/2} < \varepsilon$$

Why DeepONet as the prediction model?

Universal Approximation of functions, funcionals and operators

Theorem (UAT for Borel single-valued measurable mappings (Lanthaler, Mishra, Karniadakis, 2022))

Let $\mu \in \mathcal{P}(C(D))$ be a probability measure on $C(D)$ and let $\mathcal{G} : C(D) \to L^2(U)$ be a Borel measurable mapping, with $\mathcal{G} \in L^2(\mu)$. Then, for every $\varepsilon > 0$, there exists a **DeepONet** $\mathcal{N} = \mathcal{R} \circ \mathcal{A} \circ \mathcal{E}$ *such that*

$$\|\mathcal{G} - \mathcal{N}\|_{L^2(\mu)} = \left(\int_X \|\mathcal{G}(u) - \mathcal{N}(u)\|^2_{L^2(U)} \, d\mu(u) \right)^{1/2} < \varepsilon$$

Theorem (UAT for Borel set-valued measurable mappings)

Let $\mathcal{G} : X \mapsto Y$ be a lower semicontinuous set-valued map with closed, convex values. Then, for every $\varepsilon > 0$, there exists a **DeepONet** $\mathcal{N}$ *such that*

$$\left(\int_X \inf_{u \in \mathcal{G}(y)} \|u - \mathcal{N}(y)\|^2_{L^2(U)} \, d\mu(y) \right)^{1/2} < \varepsilon.$$

Why DeepONet as the prediction model?

Universal Approximation of functions, funcionals and operators

Theorem (UAT for Borel single-valued measurable mappings (Lanthaler, Mishra, Karniadakis, 2022))

Let $\mu \in \mathcal{P}(C(D))$ be a probability measure on $C(D)$ and let $\mathcal{G} : C(D) \to L^2(U)$ be a Borel measurable mapping, with $\mathcal{G} \in L^2(\mu)$. Then, for every $\varepsilon > 0$, there exists a **DeepONet** *$\mathcal{N} = \mathcal{R} \circ \mathcal{A} \circ \mathcal{E}$ such that*

$$\|\mathcal{G} - \mathcal{N}\|_{L^2(\mu)} = \left(\int_X \|\mathcal{G}(u) - \mathcal{N}(u)\|^2_{L^2(U)} \, d\mu(u) \right)^{1/2} < \varepsilon$$

Theorem (UAT for Borel set-valued measurable mappings)

Let $\mathcal{G} : X \mapsto Y$ be a lower semicontinuous set-valued map with closed, convex values. Then, for every $\varepsilon > 0$, there exists a **DeepONet** *$\mathcal{N}$ such that*

$$\left(\int_X \inf_{u \in \mathcal{G}(y)} \|u - \mathcal{N}(y)\|^2_{L^2(U)} \, d\mu(y) \right)^{1/2} < \varepsilon.$$

In practice, one would determine a DeepONet $\mathcal{N}$ such that $\mathcal{N}(y_i)$ is a good approximation to a suitable selection $G(y_i)$ of $\mathcal{G}(y_i)$ in the sense

$$\inf_{u \in \mathcal{G}(y_i)} \|u - \mathcal{N}(y_i)\|^2 \leq \|G(y_i) - \mathcal{N}(y_i)\|^2 < \varepsilon, \quad \text{for a finite, selected set } \{y_i\} \subset X.$$

Approximation error and the curse of dimensionality

Definition (Curse of dimensionality)

For a given $\varepsilon > 0$, let $\mathcal{N}_\varepsilon$ be a DeepONet providing error $< \varepsilon$, and

$$\text{size}\left(\mathcal{N}_\varepsilon\right) \sim \mathcal{O}\left(\varepsilon^{-\vartheta_\varepsilon}\right) \quad \text{for some } \vartheta_\varepsilon \geq 0.$$

Our DeepONet approximation is said to *incurr a curse of dimensionality* if $\lim_{\varepsilon \to 0} \vartheta_\varepsilon = +\infty$ and *breaks the curse of dimensionality* if $\lim_{\varepsilon \to 0} \vartheta_\varepsilon = \overline{\vartheta} < +\infty$.

[1] Size of a neural network is understood as the number of non-vanishing parameters of the net.

Approximation error and the curse of dimensionality

Definition (Curse of dimensionality)

For a given $\varepsilon > 0$, let $\mathcal{N}_\varepsilon$ be a DeepONet providing error $< \varepsilon$, and

$$\text{size}\left(\mathcal{N}_\varepsilon\right) \sim \mathcal{O}\left(\varepsilon^{-\vartheta_\varepsilon}\right) \quad \text{for some } \vartheta_\varepsilon \geq 0.$$

Our DeepONet approximation is said to *incurr a curse of dimensionality* if $\lim_{\varepsilon \to 0} \vartheta_\varepsilon = +\infty$ and *breaks the curse of dimensionality* if $\lim_{\varepsilon \to 0} \vartheta_\varepsilon = \overline{\vartheta} < +\infty$.

Yarotsky (2018) proved that the approximation of a general Lipschitz function to accuracy ε requires a ReLU network of size[1] $\varepsilon^{-m(\varepsilon)/2}$, with $m(\varepsilon) \to \infty$ as $\varepsilon \to 0$, and hence suffers from the **curse of dimensionality.** m is the number of sensors for the enconding $y \mapsto \mathcal{E}(y) = (y(x_1), \cdots, y(x_m))$.

[1] Size of a neural network is understood as the number of non-vanishing parameters of the net.

Approximation error and the curse of dimensionality

Definition (Curse of dimensionality)

For a given $\varepsilon > 0$, let $\mathcal{N}_\varepsilon$ be a DeepONet providing error $< \varepsilon$, and

$$\text{size}\left(\mathcal{N}_\varepsilon\right) \sim \mathcal{O}\left(\varepsilon^{-\vartheta_\varepsilon}\right) \quad \text{for some } \vartheta_\varepsilon \geq 0.$$

Our DeepONet approximation is said to *incurr a curse of dimensionality* if $\lim_{\varepsilon\to 0} \vartheta_\varepsilon = +\infty$ and *breaks the curse of dimensionality* if $\lim_{\varepsilon\to 0} \vartheta_\varepsilon = \overline{\vartheta} < +\infty$.

Yarotsky (2018) proved that the approximation of a general Lipschitz function to accuracy ε requires a ReLU network of size[1] $\varepsilon^{-m(\varepsilon)/2}$, with $m(\varepsilon) \to \infty$ as $\varepsilon \to 0$, and hence suffers from the **curse of dimensionality.** m is the number of sensors for the enconding $y \mapsto \mathcal{E}(y) = (y(x_1), \cdots, y(x_m))$.

However, if $\mathcal{G}$ is single-valued and linear, the curse of dimensionality can be broken. This is clearly so for the control of minimal L^2-norm in the case of the wave equation; but it is much more involved for the heat equation.

[1] Size of a neural network is understood as the number of non-vanishing parameters of the net.

Estimates for approximation and estimation errors

S. Lanthaler, S. Mishra, G. E. Karniadakis, *Error estimates for DeepONets: a deep learning framework in infinite dimensions*, Trans. Math. Appl. 6 (1) (2022) 1-144.

Estimates for approximation and estimation errors

S. Lanthaler, S. Mishra, G. E. Karniadakis, *Error estimates for DeepONets: a deep learning framework in infinite dimensions*, Trans. Math. Appl. 6 (1) (2022) 1-144.

Approximation error

$$\hat{\mathcal{E}}_{\text{approx}} := \left(\int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y) \right)^{1/2}.$$

Estimates for approximation and estimation errors

S. Lanthaler, S. Mishra, G. E. Karniadakis, *Error estimates for DeepONets: a deep learning framework in infinite dimensions*, Trans. Math. Appl. 6 (1) (2022) 1-144.

Approximation error

$$\hat{\mathcal{E}}_{\text{approx}} := \left(\int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y) \right)^{1/2}.$$

The DeepONet $\mathcal{N} : C(D) \to C(U)$ is decomposed into:

- An encoder

$$\mathcal{E} : C(D) \to \mathbb{R}^m, \quad y \mapsto (y(x_1), \cdots, y(x_m)), \quad x_j \in D$$

Estimates for approximation and estimation errors

S. Lanthaler, S. Mishra, G. E. Karniadakis, *Error estimates for DeepONets: a deep learning framework in infinite dimensions*, Trans. Math. Appl. 6 (1) (2022) 1-144.

Approximation error

$$\hat{\mathcal{E}}_{\text{approx}} := \left(\int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y) \right)^{1/2}.$$

The DeepONet $\mathcal{N} : C(D) \to C(U)$ is decomposed into:

- An encoder

$$\mathcal{E} : C(D) \to \mathbb{R}^m, \quad y \mapsto (y(x_1), \cdots, y(x_m)), \quad x_j \in D$$

- An approximation operator

$$\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^p, \quad \boldsymbol{y} = (y_1, \cdots, y_m) \mapsto (\beta_1(\boldsymbol{y}), \cdots, \beta_p(\boldsymbol{y})).$$

Estimates for approximation and estimation errors

S. Lanthaler, S. Mishra, G. E. Karniadakis, *Error estimates for DeepONets: a deep learning framework in infinite dimensions*, Trans. Math. Appl. 6 (1) (2022) 1-144.

Approximation error

$$\hat{\mathcal{E}}_{\text{approx}} := \left(\int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y) \right)^{1/2}.$$

The DeepONet $\mathcal{N} : C(D) \to C(U)$ is decomposed into:

- An encoder

$$\mathcal{E} : C(D) \to \mathbb{R}^m, \quad y \mapsto (y(x_1), \cdots, y(x_m)), \quad x_j \in D$$

- An approximation operator

$$\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^p, \quad \boldsymbol{y} = (y_1, \cdots, y_m) \mapsto (\beta_1(\boldsymbol{y}), \cdots, \beta_p(\boldsymbol{y})).$$

- An affine reconstruction operator

$$\mathcal{R} : \mathbb{R}^p \to C(U), \quad (\beta_1, \cdots, \beta_p) \mapsto \hat{\tau}_0(t) + \sum_{k=1}^{p} \beta_k \hat{\tau}_k(t).$$

Approximation error and the curse of dimensionality

$$\begin{array}{ccc} L^2(D) & \xrightarrow{\mathscr{G}} & L^2(U) \\ \mathscr{E} \downarrow \uparrow \mathscr{D} & & \mathscr{P} \downarrow \uparrow \mathscr{R} \\ \mathbb{R}^m & \xrightarrow{\mathscr{A}} & \mathbb{R}^p \end{array}$$

Approximation error and the curse of dimensionality

$$
\begin{array}{ccc}
L^2(D) & \xrightarrow{\mathscr{G}} & L^2(U) \\
\mathscr{E}\downarrow\uparrow\mathscr{D} & & \mathscr{P}\downarrow\uparrow\mathscr{R} \\
\mathbb{R}^m & \xrightarrow{\mathscr{A}} & \mathbb{R}^p
\end{array}
$$

- **Encoding error**

$$\hat{\mathcal{E}}_{\mathcal{E}} := \left(\int_X \|\mathcal{D} \circ \mathcal{E}(y) - y\|_X^2 \, d\mu(y) \right)^{1/2}$$

Approximation error and the curse of dimensionality

$$\begin{array}{ccc} L^2(D) & \xrightarrow{\mathscr{G}} & L^2(U) \\ \mathscr{E} \downarrow \uparrow \mathscr{D} & & \mathscr{P} \downarrow \uparrow \mathscr{R} \\ \mathbb{R}^m & \xrightarrow{\mathscr{A}} & \mathbb{R}^p \end{array}$$

- **Encoding error**

$$\hat{\mathcal{E}}_{\mathcal{E}} := \left(\int_X \|\mathcal{D} \circ \mathcal{E}(y) - y\|_X^2 \, d\mu(y) \right)^{1/2}$$

- **Approximation error due to the neural network**

$$\hat{\mathcal{E}}_{\mathcal{A}} := \left(\int_{\mathbb{R}^m} \|\mathcal{A}(\boldsymbol{y}) - \mathcal{P} \circ \mathcal{G} \circ \mathcal{D}(\boldsymbol{y})\|_{\ell^2(\mathbb{R}^p)}^2 \, d\left(\mathcal{E}_{\#}\mu\right)(\boldsymbol{y}) \right)^{1/2},$$

where

$\boldsymbol{u} = \mathcal{E}(y), \quad \mathcal{E}_{\#}(\mu)(B) = \mu\left(\mathcal{E}^{-1}(B)\right)$ is the push-forward measure

Approximation error and the curse of dimensionality

$$
\begin{array}{ccc}
L^2(D) & \xrightarrow{\mathscr{G}} & L^2(U) \\
\mathscr{E} \downarrow \uparrow \mathscr{D} & & \mathscr{P} \downarrow \uparrow \mathscr{R} \\
\mathbb{R}^m & \xrightarrow{\mathscr{A}} & \mathbb{R}^p
\end{array}
$$

- **Encoding error**

$$\hat{\mathcal{E}}_{\mathcal{E}} := \left(\int_X \|\mathcal{D} \circ \mathcal{E}(y) - y\|_X^2 \, d\mu(y) \right)^{1/2}$$

- **Approximation error due to the neural network**

$$\hat{\mathcal{E}}_{\mathcal{A}} := \left(\int_{\mathbb{R}^m} \|\mathcal{A}(\boldsymbol{y}) - \mathcal{P} \circ \mathcal{G} \circ \mathcal{D}(\boldsymbol{y})\|_{\ell^2(\mathbb{R}^p)}^2 \, d\left(\mathcal{E}_{\#}\mu\right)(\boldsymbol{y}) \right)^{1/2},$$

where

$\boldsymbol{u} = \mathcal{E}(y), \quad \mathcal{E}_{\#}(\mu)(B) = \mu\left(\mathcal{E}^{-1}(B)\right)$ is the push-forward measure

- **Reconstruction error**

$$\hat{\mathcal{E}}_{\mathcal{R}} := \left(\int_{L^2(U)} \|\mathcal{R} \circ \mathcal{P}(y) - y\|_{L^2(U)}^2 \, d\left(\mathcal{G}_{\#}\mu\right)(y) \right)^{1/2}$$

Approximation error and the curse of dimensionality

Our setup : (wave equation)

Approximation error and the curse of dimensionality

Our setup : (wave equation)

1. $\mu \in \mathcal{P}_2\left(C(D)\right)$ is a probability measure with mean 0, and with uniformly bounded eigenfunctions of the covariance operator Γ_μ

$$\Gamma_\mu = \sum_{j=1}^{\infty} \lambda_j e_j \otimes e_j$$

Approximation error and the curse of dimensionality

Our setup : (wave equation)

1. $\mu \in \mathcal{P}_2(C(D))$ is a probability measure with mean 0, and with uniformly bounded eigenfunctions of the covariance operator Γ_μ

$$\Gamma_\mu = \sum_{j=1}^{\infty} \lambda_j e_j \otimes e_j$$

2. The sensors $x_1, \cdots, x_m \sim \mathrm{Unif}(D)$ for the encoder are drawn iid random. The decoder is linear.

Approximation error and the curse of dimensionality

Our setup : (wave equation)

1. $\mu \in \mathcal{P}_2(C(D))$ is a probability measure with mean 0, and with uniformly bounded eigenfunctions of the covariance operator Γ_μ

$$\Gamma_\mu = \sum_{j=1}^{\infty} \lambda_j e_j \otimes e_j$$

2. The sensors $x_1, \cdots, x_m \sim \text{Unif}(D)$ for the encoder are drawn iid random. The decoder is linear.
3. For the reconstruction operator, $\hat{\tau}_0 = 0$ and $\hat{\tau}_k$ are the first p eigenfunctions of the covariance operator $\Gamma_{\mathcal{G}_{\#\mu}}$. The projector is

$$\mathcal{P}(u) := (< u, \hat{\tau}_1 >, \cdots, < u, \hat{\tau}_p >), \quad u \in L^2(U).$$

Breaking the curse of dimensionality

Theorem (Bound for approximation error)

Under the above conditions, let τ be the trunk net approximation of $\hat{\tau}$ such that the associated reconstruction $\mathcal{R}$ and projection $\mathcal{P}$ operators satisfy $Lip(\mathcal{R}), Lip(\mathcal{R} \circ \mathcal{P}) \leq 2$. Then, with probability 1 in the choice of the sensor points, there exists $C = C(|D|, \mu) > 0$ such that for any $m, p \in \mathbb{N}$ there exists a shallow ReLU approximator net $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^p$, with size $(\mathcal{A}) \leq 2(2+m)p$, depth $(\mathcal{A}) \leq 1$ such that the DeepONet $\mathcal{N}$ satisfies

$$\hat{\mathcal{E}}_{approx} \leq \sqrt{1 + \|\mathcal{G}\|^2} \left\{ \max_{k=1,\cdots,p} \|\hat{\tau}_k - \tau_k\|_{L^2(U)} + \sqrt{\sum_{j>p} \lambda_j} + \sqrt{\sum_{j > \frac{m}{C \log(m)}} \lambda_j} \right\}$$

Breaking the curse of dimensionality

Theorem (Bound for approximation error)

Under the above conditions, let τ be the trunk net approximation of $\hat{\tau}$ such that the associated reconstruction $\mathcal{R}$ and projection $\mathcal{P}$ operators satisfy $Lip(\mathcal{R}), Lip(\mathcal{R} \circ \mathcal{P}) \leq 2$. Then, with probability 1 in the choice of the sensor points, there exists $C = C(|D|, \mu) > 0$ such that for any $m, p \in \mathbb{N}$ there exists a shallow ReLU approximator net $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^p$, with size $(\mathcal{A}) \leq 2(2+m)p$, depth $(\mathcal{A}) \leq 1$ such that the DeepONet $\mathcal{N}$ satisfies

$$\hat{\mathcal{E}}_{approx} \leq \sqrt{1 + \|\mathcal{G}\|^2} \left\{ \max_{k=1,\cdots,p} \|\hat{\tau}_k - \tau_k\|_{L^2(U)} + \sqrt{\sum_{j>p} \lambda_j} + \sqrt{\sum_{j > \frac{m}{C \log(m)}} \lambda_j} \right\}$$

Idea of the proof.

Breaking the curse of dimensionality

Theorem (Bound for approximation error)

Under the above conditions, let τ be the trunk net approximation of $\hat{\tau}$ such that the associated reconstruction $\mathcal{R}$ and projection $\mathcal{P}$ operators satisfy $Lip(\mathcal{R}), Lip(\mathcal{R} \circ \mathcal{P}) \leq 2$. Then, with probability 1 in the choice of the sensor points, there exists $C = C(|D|, \mu) > 0$ such that for any $m, p \in \mathbb{N}$ there exists a shallow ReLU approximator net $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^p$, with size $(\mathcal{A}) \leq 2(2+m)p$, depth $(\mathcal{A}) \leq 1$ such that the DeepONet $\mathcal{N}$ satisfies

$$\hat{\mathcal{E}}_{approx} \leq \sqrt{1 + \|\mathcal{G}\|^2} \left\{ \max_{k=1,\cdots,p} \|\hat{\tau}_k - \tau_k\|_{L^2(U)} + \sqrt{\sum_{j>p} \lambda_j} + \sqrt{\sum_{j > \frac{m}{C \log(m)}} \lambda_j} \right\}$$

Idea of the proof. Crucially, since $\mathcal{G}$ is linear, and the decoder and projector are affine, there exists and exact affine approximator $\mathcal{A} = \mathcal{P} \circ \mathcal{G} \circ \mathcal{D}$, which can be represented by a shallow ReLU of the claimed size since

$$Ax + b = \sigma(Ax + b) - \sigma(-(Ax + b)).$$

Thus, $\hat{\mathcal{E}}_{\mathcal{A}} = 0$.

Breaking the curse of dimensionality

Theorem (Bound for approximation error)

Under the above conditions, let τ be the trunk net approximation of $\hat{\tau}$ such that the associated reconstruction $\mathcal{R}$ and projection $\mathcal{P}$ operators satisfy $Lip(\mathcal{R}), Lip(\mathcal{R} \circ \mathcal{P}) \leq 2$. Then, with probability 1 in the choice of the sensor points, there exists $C = C(|D|, \mu) > 0$ such that for any $m, p \in \mathbb{N}$ there exists a shallow ReLU approximator net $\mathcal{A} : \mathbb{R}^m \to \mathbb{R}^p$, with size $(\mathcal{A}) \leq 2(2+m)p$, depth $(\mathcal{A}) \leq 1$ such that the DeepONet $\mathcal{N}$ satisfies

$$\hat{\mathcal{E}}_{approx} \leq \sqrt{1 + \|\mathcal{G}\|^2} \left\{ \max_{k=1,\cdots,p} \|\hat{\tau}_k - \tau_k\|_{L^2(U)} + \sqrt{\sum_{j>p} \lambda_j} + \sqrt{\sum_{j>\frac{m}{C\log(m)}} \lambda_j} \right\}$$

Idea of the proof. Crucially, since $\mathcal{G}$ is linear, and the decoder and projector are affine, there exists and exact affine approximator $\mathcal{A} = \mathcal{P} \circ \mathcal{G} \circ \mathcal{D}$, which can be represented by a shallow ReLU of the claimed size since

$$Ax + b = \sigma(Ax + b) - \sigma(-(Ax + b)).$$

Thus, $\hat{\mathcal{E}}_{\mathcal{A}} = 0$.The estimate

$$\sum_{j>p} \lambda_j^{\mathcal{G}_{\#\mu}} \leq \|\mathcal{G}\|^2 \sum_{j>p} \lambda_j$$

must be used as well.

Breaking the curse of dimensionality

In practice, μ is chosen as the law of a Gaussian field

$$a(x,\omega) = \sum_{k\in\mathbb{Z}^d} \alpha_k \xi_k(\omega) e_k(x), \quad x \in \mathbb{T}^d := [0, 2\pi]^d,$$

where $|\alpha_k| \leq \exp(-\ell|k|)$, $\ell > 0$.

Breaking the curse of dimensionality

In practice, μ is chosen as the law of a Gaussian field

$$a(x,\omega) = \sum_{k\in\mathbb{Z}^d} \alpha_k \xi_k(\omega) e_k(x), \quad x \in \mathbb{T}^d := [0,2\pi]^d,$$

where $|\alpha_k| \leq \exp(-\ell|k|)$, $\ell > 0$.Then, for any $\varepsilon > 0$ and $\sigma > 0$, and taking

$$m \sim \log(\varepsilon^{-1})^{d(1+\sigma)} \quad \text{and} \quad p \sim \log(\varepsilon^{-1})^{d}$$

an overall approximation error $\hat{\mathcal{E}}_{\text{approx}} \lesssim \varepsilon$ may be achieved with a DeepONet $\mathcal{N}$ whose branch β and trunk τ nets satisfy

$$\text{size}(\beta) \lesssim \log(\varepsilon^{-1})^{2d+\sigma}, \quad \text{depth}(\beta) \lesssim 1$$

and

$$\text{size}(\tau) \lesssim \log(\varepsilon^{-1})^{d+2+\sigma}, \quad \text{depth}(\tau) \lesssim \log(\varepsilon^{-1})^{2+\sigma}.$$

Breaking the curse of dimensionality

In practice, μ is chosen as the law of a Gaussian field

$$a(x,\omega) = \sum_{k\in\mathbb{Z}^d} \alpha_k \xi_k(\omega) e_k(x), \quad x \in \mathbb{T}^d := [0,2\pi]^d,$$

where $|\alpha_k| \leq \exp(-\ell|k|)$, $\ell > 0$.Then, for any $\varepsilon > 0$ and $\sigma > 0$, and taking

$$m \sim \log(\varepsilon^{-1})^{d(1+\sigma)} \quad \text{and} \quad p \sim \log(\varepsilon^{-1})^d$$

an overall approximation error $\hat{\mathcal{E}}_{\text{approx}} \lesssim \varepsilon$ may be achieved with a DeepONet $\mathcal{N}$ whose branch β and trunk τ nets satisfy

$$\text{size}(\beta) \lesssim \log(\varepsilon^{-1})^{2d+\sigma}, \quad \text{depth}(\beta) \lesssim 1$$

and

$$\text{size}(\tau) \lesssim \log(\varepsilon^{-1})^{d+2+\sigma}, \quad \text{depth}(\tau) \lesssim \log(\varepsilon^{-1})^{2+\sigma}.$$

This means that **the proposed DeepONet-based approximation scheme for the controllability map breaks the curse of dimensionality** with respect to p and the number of sensor points m in the sense that the complexity of the DeepONet does not grow exponentially with these two parameters.

Estimation (or generalization) error

Consider the loss function

$$\hat{\mathcal{L}}(\mathcal{N}) := \int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y). \tag{4}$$

Estimation (or generalization) error

Consider the loss function

$$\hat{\mathcal{L}}(\mathcal{N}) := \int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y). \tag{4}$$

and the empirical loss

$$\hat{\mathcal{L}}_N(\mathcal{N}) = \frac{|U|}{N} \sum_{j=1}^{N} |\mathcal{G}(y_j)(t_j) - \mathcal{N}(y_j)(t_j)|^2 \tag{5}$$

where $(y_j, t_j) \sim \mu \otimes \text{Unif}(U)$.

Estimation (or generalization) error

Consider the loss function

$$\hat{\mathcal{L}}(\mathcal{N}) := \int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y). \tag{4}$$

and the empirical loss

$$\hat{\mathcal{L}}_N(\mathcal{N}) = \frac{|U|}{N} \sum_{j=1}^{N} |\mathcal{G}(y_j)(t_j) - \mathcal{N}(y_j)(t_j)|^2 \tag{5}$$

where $(y_j, t_j) \sim \mu \otimes \mathrm{Unif}(U)$.
Let $\hat{\mathcal{N}}$ be an optimizer of (4) and let $\hat{\mathcal{N}}_N$ be an optimizer of (5).

Estimation (or generalization) error

Consider the loss function

$$\hat{\mathcal{L}}(\mathcal{N}) := \int_{L^2(D)} \int_U |\mathcal{G}(y)(t) - \mathcal{N}(y)(t)|^2 dt d\mu(y). \tag{4}$$

and the empirical loss

$$\hat{\mathcal{L}}_N(\mathcal{N}) = \frac{|U|}{N} \sum_{j=1}^{N} |\mathcal{G}(y_j)(t_j) - \mathcal{N}(y_j)(t_j)|^2 \tag{5}$$

where $(y_j, t_j) \sim \mu \otimes \mathrm{Unif}(U)$.
Let $\hat{\mathcal{N}}$ be an optimizer of (4) and let $\hat{\mathcal{N}}_N$ be an optimizer of (5).

Estimation (Generalization) error

$$\hat{\mathcal{E}}_{\text{gen}} = \sqrt{\hat{\mathcal{L}}\left(\hat{\mathcal{N}}_N\right) - \hat{\mathcal{L}}\left(\hat{\mathcal{N}}\right)}$$

Estimation (or generalization) error

1) *Boundedness assumption:* there exists $\psi : L^2(D) \to [0, +\infty[$ such that

$$|\mathcal{G}(y)(t)| \leq \psi(y), \qquad \sup_{\theta \in [-B,B]^{d_\theta}} |\mathcal{N}_\theta(y)(t)| \leq \psi(y), \quad \forall y \in L^2(D), \forall t \in U,$$

and there exists $C, \kappa > 0$ such that $\psi(y) \leq C \left(1 + \|y\|_{L^2(D)}\right)^\kappa$.

Estimation (or generalization) error

1) *Boundedness assumption:* there exists $\psi : L^2(D) \to [0, +\infty[$ such that

$$|\mathcal{G}(y)(t)| \leq \psi(y), \qquad \sup_{\theta \in [-B,B]^{d_\theta}} |\mathcal{N}_\theta(y)(t)| \leq \psi(y), \quad \forall y \in L^2(D), \forall t \in U,$$

and there exists $C, \kappa > 0$ such that $\psi(y) \leq C \left(1 + \|y\|_{L^2(D)}\right)^\kappa$.

2) *Lipschitz continuity assumption:* there exists $\Phi : L^2(D) \to [0, +\infty[$ such that

$$|\mathcal{N}_\theta(y)(t) - \mathcal{N}_{\theta'}(y)(t)| \leq \Phi(y) \|\theta - \theta'\|_{\ell^\infty}, \quad \forall y \in L^2(D), \forall t \in U,$$

and there exists $C, \kappa > 0$ such that $\Phi(y) \leq C \left(1 + \|y\|_{L^2(D)}\right)^\kappa$.

Estimation (or generalization) error

1) *Boundedness assumption:* there exists $\psi : L^2(D) \to [0, +\infty[$ such that

$$|\mathcal{G}(y)(t)| \leq \psi(y), \qquad \sup_{\theta \in [-B,B]^{d_\theta}} |\mathcal{N}_\theta(y)(t)| \leq \psi(y), \quad \forall y \in L^2(D), \forall t \in U,$$

and there exists $C, \kappa > 0$ such that $\psi(y) \leq C \left(1 + \|y\|_{L^2(D)}\right)^\kappa$.

2) *Lipschitz continuity assumption:* there exists $\Phi : L^2(D) \to [0, +\infty[$ such that

$$|\mathcal{N}_\theta(y)(t) - \mathcal{N}_{\theta'}(y)(t)| \leq \Phi(y) \|\theta - \theta'\|_{\ell^\infty}, \quad \forall y \in L^2(D), \forall t \in U,$$

and there exists $C, \kappa > 0$ such that $\Phi(y) \leq C \left(1 + \|y\|_{L^2(D)}\right)^\kappa$.

Theorem (Bound for generalization error (Lanthaler-Mishra-Karniadakis))

Let $\hat{\mathcal{N}}$ be an optimizer of (4) and let $\hat{\mathcal{N}}_N$ be an optimizer of (5). If the above two assumptions hold, then there exists $C = C(\mu, \psi, \Phi)$, independent of B and d_θ, such that

$$\mathbb{E}\left[\hat{\mathcal{L}}\left(\hat{\mathcal{N}}_N\right) - \hat{\mathcal{L}}\left(\hat{\mathcal{N}}\right)\right] \leq \frac{C}{\sqrt{N}} \left(1 + C d_\theta \log\left(CB\sqrt{N}\right)\right)^{2\kappa + \frac{1}{2}} \qquad (6)$$

Estimation (or generalization) error

1) *Boundedness assumption:* there exists $\psi : L^2(D) \to [0, +\infty[$ such that

$$|\mathcal{G}(y)(t)| \leq \psi(y), \qquad \sup_{\theta \in [-B,B]^{d_\theta}} |\mathcal{N}_\theta(y)(t)| \leq \psi(y), \quad \forall y \in L^2(D), \forall t \in U,$$

and there exists $C, \kappa > 0$ such that $\psi(y) \leq C \left(1 + \|y\|_{L^2(D)}\right)^\kappa$.

2) *Lipschitz continuity assumption:* there exists $\Phi : L^2(D) \to [0, +\infty[$ such that

$$|\mathcal{N}_\theta(y)(t) - \mathcal{N}_{\theta'}(y)(t)| \leq \Phi(y) \|\theta - \theta'\|_{\ell^\infty}, \quad \forall y \in L^2(D), \forall t \in U,$$

and there exists $C, \kappa > 0$ such that $\Phi(y) \leq C \left(1 + \|y\|_{L^2(D)}\right)^\kappa$.

Theorem (Bound for generalization error (Lanthaler-Mishra-Karniadakis))

Let $\hat{\mathcal{N}}$ be an optimizer of (4) and let $\hat{\mathcal{N}}_N$ be an optimizer of (5). If the above two assumptions hold, then there exists $C = C(\mu, \psi, \Phi)$, independent of B and d_θ, such that

$$\mathbb{E}\left[\hat{\mathcal{L}}\left(\hat{\mathcal{N}}_N\right) - \hat{\mathcal{L}}\left(\hat{\mathcal{N}}\right)\right] \leq \frac{C}{\sqrt{N}} \left(1 + C d_\theta \log\left(CB\sqrt{N}\right)\right)^{2\kappa + \frac{1}{2}} \tag{6}$$

$|\mathcal{G}(y)(t)| \leq \psi(y)$ is like a uniform observability inequality .

Numerical experiments

Data generation. Initial conditions are computed by sampling a Gaussian ramdon field with the kernel

$$C(x, x') = \exp\left(-\frac{|x - x'|^2}{2\ell}\right), \quad x, x' \in (0, 1),$$

where ℓ is the correlation length. Thus, the associated input measure μ is

$$a(x, \omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} e_i(x) \xi_i(\omega),$$

where ξ_i are iid standard Gaussian variables, and $\{\lambda_i, e_i(x)\}_{i=1}^{\infty}$ are the eigenvalues and normalized eigenfuncions of the operator $\mathcal{C} : L^2(0, 1) \to L^2(0, 1)$

$$\mathcal{C}(\phi)(x) = \int_0^1 C(x, x') \phi(x') \, dx'.$$

Numerical experiments

Data generation. Initial conditions are computed by sampling a Gaussian ramdon field with the kernel

$$C(x, x') = \exp\left(-\frac{|x - x'|^2}{2\ell}\right), \quad x, x' \in (0, 1),$$

where ℓ is the correlation length. Thus, the associated input measure μ is

$$a(x, \omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} e_i(x) \xi_i(\omega),$$

where ξ_i are iid standard Gaussian variables, and $\{\lambda_i, e_i(x)\}_{i=1}^{\infty}$ are the eigenvalues and normalized eigenfuncions of the operator $\mathcal{C} : L^2(0, 1) \to L^2(0, 1)$

$$\mathcal{C}(\phi)(x) = \int_0^1 C(x, x')\phi(x')\, dx'.$$

DeepONet size. We use the ReLU activation function, trunk depth/width $=$ 2/40 and branch depth/width $=$ 2/40.

Numerical experiments

Data generation. Initial conditions are computed by sampling a Gaussian ramdon field with the kernel

$$C(x, x') = \exp\left(-\frac{|x - x'|^2}{2\ell}\right), \quad x, x' \in (0, 1),$$

where ℓ is the correlation length. Thus, the associated input measure μ is

$$a(x, \omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} e_i(x) \xi_i(\omega),$$

where ξ_i are iid standard Gaussian variables, and $\{\lambda_i, e_i(x)\}_{i=1}^{\infty}$ are the eigenvalues and normalized eigenfuncions of the operator $\mathcal{C} : L^2(0, 1) \to L^2(0, 1)$

$$\mathcal{C}(\phi)(x) = \int_0^1 C(x, x')\phi(x')\, dx'.$$

DeepONet size. We use the ReLU activation function, trunk depth/width $=$ 2/40 and branch depth/width $= 2/40$.
Training process. Adam optimizer with learning rate 0.01. Initialization of the parameters is carried out with Glorot uniform.
Implementation. DeepONet: learning operators [Lu, Li, Pang, Zhang, Karniadakis: Nat. Mach. Intell., 2021]
`https://github.com/lululxvi/deepxde`

Numerical experiments: the wave equation

An academic example with an explicit solution

$$\begin{cases} y_{tt} - y_{xx} = 0, & \text{in } (0,1) \times (0,2) \\ y(x,0) = y^0(x), & \text{on } (0,1) \\ y_t(x,0) = y^1(x) & \text{on } (0,1) \\ y(0,t) = 0, & \text{on } (0,2) \\ y(1,t) = u(t) & \text{on } (0,2) \\ y(x,2) = y_t(x,2) = 0, & \text{on } (0,1). \end{cases}$$

Numerical experiments: the wave equation

An academic example with an explicit solution

$$\begin{cases} y_{tt} - y_{xx} = 0, & \text{in } (0,1)\times(0,2) \\ y(x,0) = y^0(x), & \text{on } (0,1) \\ y_t(x,0) = y^1(x) & \text{on } (0,1) \\ y(0,t) = 0, & \text{on } (0,2) \\ y(1,t) = u(t) & \text{on } (0,2) \\ y(x,2) = y_t(x,2) = 0, & \text{on } (0,1). \end{cases}$$

The operator to be approximated is

$$\begin{aligned} \mathcal{G} : \quad L^2(0,1)\times H^{-1}(0,1) & \to L^2(0,2) \\ (y^0, y^1) & \mapsto \mathcal{G}(y^0, y^1) := u \end{aligned}$$

An academic example with an explicit solution

$$\begin{cases} y_{tt} - y_{xx} = 0, & \text{in } (0,1)\times(0,2) \\ y(x,0) = y^0(x), & \text{on } (0,1) \\ y_t(x,0) = y^1(x) & \text{on } (0,1) \\ y(0,t) = 0, & \text{on } (0,2) \\ y(1,t) = u(t) & \text{on } (0,2) \\ y(x,2) = y_t(x,2) = 0, & \text{on } (0,1). \end{cases}$$

The operator to be approximated is

$$\mathcal{G}: \quad \begin{array}{cl} L^2(0,1)\times H^{-1}(0,1) & \to L^2(0,2) \\ (y^0, y^1) & \mapsto \mathcal{G}(y^0, y^1) := u \end{array}$$

where

$$u(t) = \begin{cases} \frac{1}{2} y^0(1-t) + \frac{1}{2}\int_{1-t}^{1} y^1(s)\, ds, & 0 \le t \le 1 \\ \\ -\frac{1}{2} y^0(t-1) + \frac{1}{2}\int_{t-1}^{1} y^1(s)\, ds, & 1 < t \le 2 \end{cases}$$

is the **unique** control of minimal L^2-norm.

Numerical experiments: the wave equation

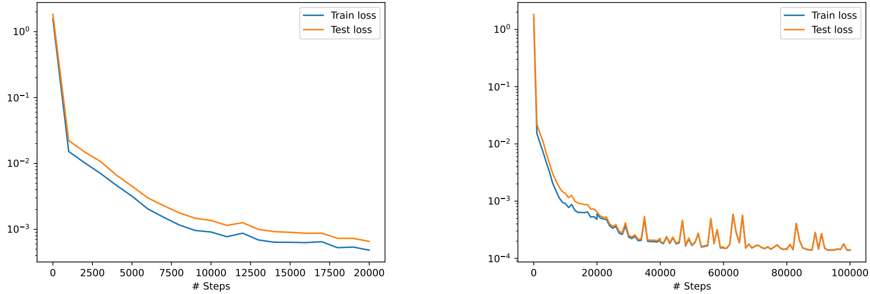


Figure: Loss history for correlation lengths of 0.25 for position and 0.125 for velocity. Number of sensor points = 51. Sample functions: (Left) 100 and (right) 10000.

Numerical experiments: the wave equation

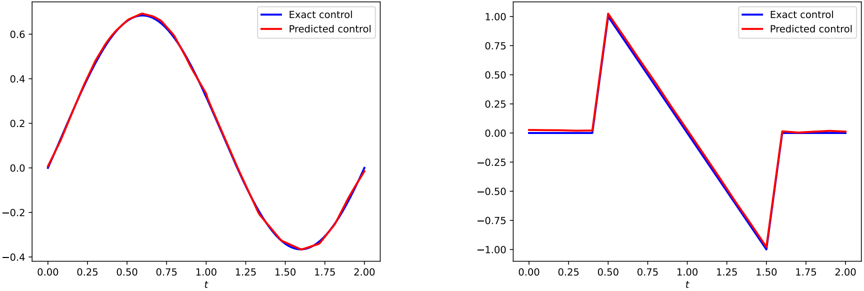


Figure: Exact versus predicted solutions. $n_{functions} = 10^4$ **(Left) Smooth initial data:** $y^0 = y^1 = \sin(\pi x)$, $(\ell_{pos}, \ell_{vel}) = (0.25, 0.125)$, $n_{sensors}$=100. Relative error $\approx 1\%$.
(Right) Non-smooth initial data: $y^0(x) = \begin{cases} 4x, & 0 \leq x \leq 0.5 \\ 0, & 0.5 < x \leq 1 \end{cases}$,
$(\ell_{pos}, \ell_{vel}) = (0.0625, 0.03125)$, $n_{sensors}$=10. Relative error $\approx 4\%$.

Numerical experiments: the heat equation

$$\begin{cases} y_t - y_{xx} = 0, & 0 < x < 1, 0 < t < T \\ y(x,0) = y^0(x), & 0 \leq x \leq 1 \\ y(0,t) = 0, & 0 \leq t \leq T \\ y(1,t) = u(t), & 0 \leq t \leq T \end{cases} \tag{7}$$

$y(x, T) = 0$, $0 \leq x \leq 1$.

Numerical experiments: the heat equation

$$\begin{cases} y_t - y_{xx} = 0, & 0 < x < 1, 0 < t < T \\ y(x,0) = y^0(x), & 0 \leq x \leq 1 \\ y(0,t) = 0, & 0 \leq t \leq T \\ y(1,t) = u(t), & 0 \leq t \leq T \end{cases} \tag{7}$$

$y(x, T) = 0$, $0 \leq x \leq 1$.

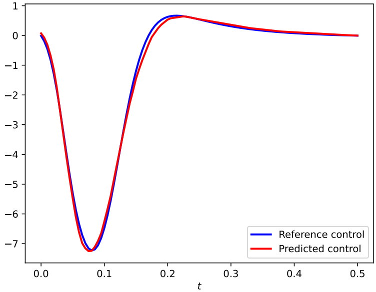


Figure: Heat equation. PINN (reference control) versus DeepONet (predicted) controls for the initial condition $y^0(x) = \sin(\pi x)$. $n_{functions} = 275$, $\ell_0 = 0.25$, $n_{sensors} = 101$, $p = 100$.