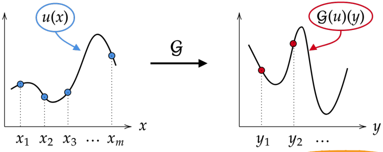
Deep Learning in Scientific Computing 2023: Lecture 10

Siddhartha Mishra

Seminar for Applied Mathematics (SAM), D-MATH (and),
ETH AI Center,
ETH Zürich, Switzerland.

Operator Learning

function   function



- Underlying Solution Operator: $\mathcal{G}(a) = u$ for PDE $\boxed{\mathcal{D}u = a}$
- Task: Find a Surrogate (based on DNNs) $\mathcal{G}^* \approx \mathcal{G}$ from data.
- Inputs+Outputs for $\mathcal{G}^*$ are Functions.
- Some notion of Continuous-Discrete Equivalence

Operator Learning Architectures

$$
\begin{tikzcd}
X \arrow[rr, "\mathcal{G}"] \arrow[d, bend right, squiggly, "\mathcal{E}"'] & & Y \arrow[d, bend right, "\mathcal{P}"'] \\
\mathbb{R}^m \arrow[u, bend right, "\mathcal{D}"'] \arrow[rr, squiggly, "\mathcal{A}"'] & & \mathbb{R}^p \arrow[u, bend right, squiggly, "\mathcal{R}"']
\end{tikzcd}
$$

DNNs

Architecture	Encoder	Approximator	Reconstructor
SNO[1]	Coeffs	DNNs	Chebychev basis
DeepOnet	Sensor Evals.	DNNs	DNNs
PCA-Net[2]	Input PCA	DNNs	Output PCA

[1] Fanaskov and Oseledets, 2022
[2] Bhattacharya et al, 2020

Why do ONets work ?: Lanthaler, SM, Karniadakis, 2022

DNNs approximated measurable functions to any desired accuracy.

$\hat{\mathcal{E}} = \| G - G^* \|_{\text{operator norm}}$

DeepOnet

$$
\begin{tikzcd}
X \arrow[r, "\mathcal{G}"] \arrow[d, shift left, "\mathcal{D}"] & Y \arrow[d, shift left, "\mathcal{R}"] \\
\mathbb{R}^m \arrow[u, shift left, "\mathcal{E}"] \arrow[r, squiggly, "\mathcal{A}"] & \mathbb{R}^p \arrow[u, shift left, "\mathcal{P}"]
\end{tikzcd}
$$

$G : L^2 \mapsto L^2$

$G^* \approx \mathcal{R} \circ \mathcal{A} \circ \mathcal{E}$

- Universal Approximation Thm: For $\mu \in Prob(L^2(D))$ and any measurable $\mathcal{G} : H^r \mapsto H^s$ and $\epsilon > 0$, $\exists \mathcal{N}$ (Onet) $\hat{\mathcal{E}} < \epsilon$
- Upper (and lower) bounds via $\mathcal{E}_{\mathcal{R}} \leq \hat{\mathcal{E}} \leq C(\mathcal{E}_{\mathcal{E}} + \mathcal{E}_{\mathcal{A}} + \mathcal{E}_{\mathcal{R}})$.
- $\mathcal{E}_{\mathcal{E},\mathcal{R}}$ decay as spectrum of Covariance operator of μ and $\mathcal{G}\#_{\mu}$.
- For $G = \mathcal{P} \circ \mathcal{G} \circ \mathcal{D} \in C^k(\mathbb{R}^m, \mathbb{R}^p) \Rightarrow \text{size}\ (\mathcal{N}) \sim \mathcal{O}\left(\epsilon^{-\frac{m(\epsilon)}{k}}\right)$.
- Curse of Dimensionality (CoD) for Onets !!!

$\|_{G^*}$

Qn: Is there an similar universal approximation result for operators with DeepOnets

Given $v \in X$, $\mu \sim \mathrm{Prob}(X)$

Covariance Operator: $C = \mathbb{E}_{v \sim \mu}[v \otimes v]$

$$= \int_X v \otimes v \, d\mu(v) \simeq \frac{1}{m} \sum_{k=1}^{m} v_k \otimes v_k$$

Eigenvalues of Cov λ_ℓ, $\ell = 1, 2, \ldots$

Eigenfunctions of Cov ψ_ℓ.

C_μ

<u>Encoding</u>: $\mu \sim \mathrm{Prob}(X)$

<u>Reconstruction</u>: $G_\# \mu \in \mathrm{Prob}(Y)$ $\quad$ G is the operator $G : X \mapsto Y$

$$\int_Y F(w) \, d G_\# \mu = \int_X F(Gv) \, d\mu(v)$$

Operator — G

DeepOnet — G^*

$$\text{Size}(G^*) \sim O\left(\epsilon^{-\frac{m(\epsilon)}{k}}\right)$$

$\epsilon \to$ Error amplitude

$k \to$ measure of regularity $k \geq 0$

$m(\epsilon)$ — # number of sensor points / fourier modes

$\epsilon \simeq 0$

To ensure $\mathcal{E}_\epsilon \to 0$

we have: $m(\epsilon) \to \infty$

$m(\epsilon) \gg 1$,

$$\text{Size}(G^*) \sim \overline{O\left(\frac{1}{\epsilon}^{\underline{m(\epsilon)}}\right)}^{\to +\infty}$$

$\xrightarrow{\frac{1}{\epsilon}} +\infty$

On CoD for Operator Networks

PCA-net / SNO

algebraic dependence

Size (DeepOnet)

- DeepOnet breaks the Curse of Dimensionality !!
- For operators $\mathcal{G}$ corresponding to many PDEs:

0.3%

elliptic

Allen-Cahn

NS

PDE	Operator	Complexity
$-u'' = \sin(u) + f$	$\mathcal{G} : f \to u(T)$	$M \sim \mathcal{O}(\epsilon^{-\eta})$ $\eta \approx 0$
$-\mathrm{div}(a\nabla u) = f$	$\mathcal{G} : a \to u$	$M \sim \mathcal{O}(\epsilon^{-\eta})$ $\eta \approx 0$
$u_t = \Delta u + f(u)$	$\mathcal{G} : u_0 \to u(T)$	$M \sim \mathcal{O}(\epsilon^{-2(d+1)})$
$u_t + u \cdot \nabla u + \nabla p = \nu \Delta u$	$\mathcal{G} : u_0 \to u(T)$	$M \sim \mathcal{O}(\epsilon^{-(d+1)})$
$u_t + \mathrm{div}(f(u)) = 0$	$\mathcal{G} : u_0 \to u(T)$	$M \sim \mathcal{O}(\epsilon^{-\alpha(d+1)})$

$\log(1/\epsilon)$

- Case by Case basis: Operator Holomorphy, Emulation of Numerical Schemes etc...
- Unified framework in DeRyck, SM, 2022.

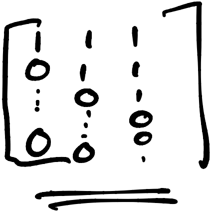
$\left(\frac{1}{10^2}\right)^3 \sim 10^6$

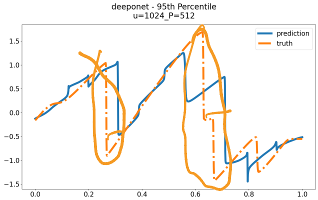
physical dimension $d = 1, 2, 3$

DeepONets: Issues / SNO / PCA-Net

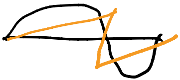
- Affine Reconstructors $\Rightarrow \sqrt{\sum_{\ell>p}\lambda_\ell} \leq \mathcal{E}_{\mathcal{R}} \leq \mathcal{E}$
- Large error for Slow decay of eigenvalues of Covariance operator of $\mathcal{G}\#_\mu$.
- Holds for Transport dominated problems
- Error of 30% for Burgers' equation with GRF initial data !!

$u_t + \left(\frac{u^2}{2}\right)_x = 0$





DeepOnet



Alternative: Neural Operators

$V: \mathbb{R}^{d_{in}} \mapsto \mathbb{R}^{d_{out}}$

- Formalized in Kovachki et al, 2021.
- Recall: DNNs are $\mathcal{L}_\theta = \sigma_K \odot \sigma_{K-1} \odot \dots\dots \sigma_1$ — layer weight
- Single hidden layer: $\sigma_k(y) = \sigma(A_k y + B_k)$
- Neural Operators generalize DNNs to ∞-dimensions:
- NO: $\boxed{\mathcal{N}_\theta = \mathcal{N}_L \odot \mathcal{N}_{L-1} \odot \dots\dots \mathcal{N}_1}$
- Single hidden layer;

$\sigma(f)$ is a function

$\sigma(f)(x) = \sigma(f(x))$

$$(\mathcal{N}_\ell v)(x) = \sigma\left(A_\ell v(x) + B_\ell(x) + \int_D K_\ell(x,y)v(y)dy\right)$$

pointwise. $\;=0\;$ DNN parametrized

- Kernel Integral Operators
- Learning Parameters in A_ℓ, B_ℓ, K_ℓ
- Different Kernels $\Rightarrow$ Low-Rank NOs, Graph NOs, Multipole NOs, ...

DNN

Input - vector y_l 　　 l-th layer

Output - vector y_{l+1}

$y_l \in \mathbb{R}^{d_l}$

$y_{l+1} \in \mathbb{R}^{d_{l+1}}$

$$y_{l+1} = \boxed{\sigma(A_l y_l + B_l)}$$

$\sigma(u_1, \dots, u_n) = (\sigma(u_1), \dots, \sigma(u_n))$

$A_l \in \mathbb{R}^{d_{l+1} \times d_l}$ 　 $B_l \in \mathbb{R}^{d_{l+1}}$

$$N = N_L \circ \dots \circ N_2 \circ N_1$$

N_l is a operator $N_l \, v_l \longmapsto v_{l+1}$

$v_l, v_{l+1} \in C(D, \mathbb{R}^{d_l}) / C(D, \mathbb{R}^{d_{l+1}})$

continuous.

Bias vector $B_l \xrightarrow{\text{replace}}$ Bias function

$A_l y_l \to$ Matrix-vector multiply:

- general form of a linear funct.

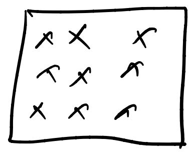
Infinite dimensional form of Matrix-vector multiply:

$$L : v \longmapsto Lv$$

$L \to$ linear Operator

$$\boxed{Lv(x) = \int_D k(x,y)\, v(y)\, dy}$$

Kernel-Integral Operator:



D

use MC-Quadrature

$$Lv(x_i) \approx \frac{1}{m} \sum_{j=1}^{m} k(x_i, y_j)\, v(y_j)$$

let $K_{ij} = \frac{1}{m} k(x_i, y_j)$.

$$Lv_i = (Kv)_i$$

Neural Operator:

$$\mathcal{N} = \mathcal{N}_L \circ \dots \circ \mathcal{N}_2 \circ \mathcal{N}_1$$

$$\mathcal{N}_\ell v = \sigma\left[\int k_\ell(x,y)\, v(y)\, dy + B_\ell\right]$$

Discretize v by $\{v(x_i)\}_{i=1}^{N}$ pt values

$$\mathcal{N}_\ell v(x_i) = \sigma\left[\int_D k_\ell(x_i, y) v(y) dy + B_\ell(x_i)\right]$$

Quadrature

$1 \le i, j \le N$

$$\approx \sigma\left[\sum_{j=1}^{N} k_\ell(x_i, y_j)\, v(y_j)\, \omega_j + B_\ell(x_i)\right]$$

Computational Complexity ~ $O(N^2)$

$O(N \log N)$

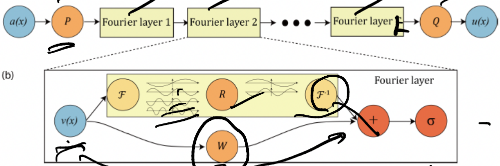
Fourier Neural Operators (FNO)

- FNO proposed in Li et al, 2020.

$d >> d_0$

lifting to a latent space

Projection from latent space.



Projection:

$Q: L^2(D, \mathbb{R}^{d_c}) \to L^2(D, \mathbb{R}^{d_{out}})$

$d >> d_{out}$

$a \in L^2(D, \mathbb{R}^{d_0})$

$Pa \in L^2(D, \mathbb{R}^{d_1})$

$- P \in \mathbb{R}^{d_1 \times d_0}$ matrix

Residual connection.

- Translation invariant Kernel $K(x, y) = K(x - y)$
- Use Fourier and Inverse Fourier Transform to define the KIO:

$$\int_D K_\ell(x, y) v(y) dy = \mathcal{F}^{-1}(\mathcal{F}(K)\mathcal{F}(v))(x)$$

- Parametrize Kernel in Fourier space.
- Fast implementation through FFT

FNO: $\mathcal{N}_l v(x) = \sigma\left(B_l(x) + \int_D k_l(x,y)\, v(y)\, dy \right)$

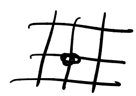
1st trick: $k_l(x,y) = k_l(x-y)$

$$\int_D k_l(x-y)\, v(y)\, dy = k_l * v \quad \text{(convolution)}$$

Main Idea: Perform Convolution in Fourier space

Fourier transform: $\mathcal{F} : L^2(D, \mathbb{C}^n) \longmapsto \ell^2(\mathbb{Z}^d, \mathbb{C}^n)$

$D \subseteq \mathbb{R}^d$ – fix $1 \leq j \leq n$ $\quad \mathcal{F}v$



$$(\mathcal{F} v_j)(k) = \int_D v_j(x)\, \psi_k(x)\, dx$$

mode

$$\psi_k(x) = e^{-2\pi i \langle k, x \rangle}$$

Inverse Fourier Transform: $\mathcal{F}^{-1}$

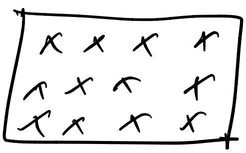
$$\mathcal{F}^{-1} : \ell^2(\mathbb{Z}^d, \mathbb{C}^n) \longmapsto L^2(D, \mathbb{C}^n)$$

$$\mathcal{F}^{-1}\omega_j(x) = \sum_{j \in \mathbb{R}^d} \omega_j \psi_j(x)$$

$$\mathcal{F}(k * v) = \mathcal{F}(k)\,\mathcal{F}(v)$$

$$\underline{\mathcal{F}^{-1}\mathcal{F}}(k * v) = \mathcal{F}^{-1}\big(\mathcal{F}(k)\,\mathcal{F}(v)\big)$$

$$\Rightarrow \quad k * v = \mathcal{F}^{-1}\big(\underset{R_0}{\underset{\parallel}{\mathcal{F}(k)}}\,\mathcal{F}(v)\big)$$



$O(N^2)$

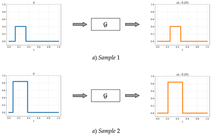
FFT $\sim$ $O(N\log(N))$

Theory for FNOs

- Results of Kovachki, Lanthaler,SM, 2021
- Universal Approximation Thm: For $\mu \in Prob(L^2(D))$ and any measurable $\mathcal{G} : H^r \mapsto H^s$ and $\epsilon > 0$, $\exists \mathcal{N}$ (FNO): $\hat{\mathcal{E}} < \epsilon$
- FNOs break the Curse of Dimensionality for a variety of PDEs.
- FNO Size grows polynomially wrt Error !

DeepONet vs. FNO

- Theory of (Lanthaler, Molinaro, Hadorn, SM, 2022):
- For Linear Advection Equation with Discontinuities:

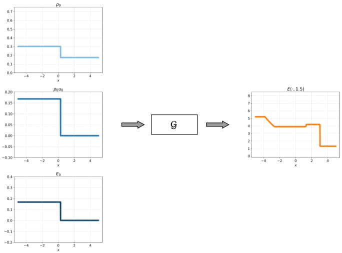


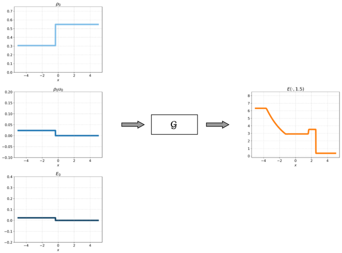
- Thm: To obtain ϵ error:
  - Size(DeepOnet) $\sim \mathcal{O}(\epsilon^{-2})$
  - Size(FNO) $\sim \mathcal{O}(\log(\epsilon^{-1}))$!!
- Results:

Architecture	ResNet	FCNN	DONet	FNO
Error	14.8%	23.3%	7.9%	0.7%

- Analogous theorem for Burgers' equation.

Operator Learning for Euler Equations

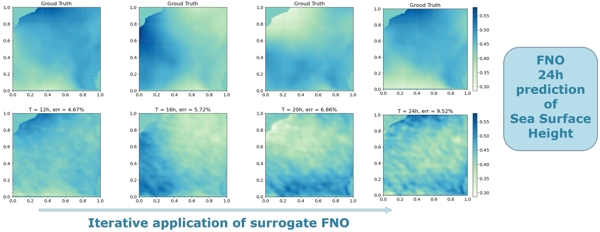




Sample 1 Sample 2

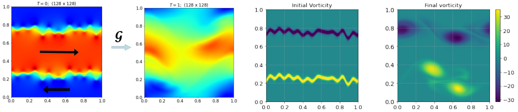
Architecture	Error
ResNet	4.5%
ConvNet	8.9%
DeepONet	4.2%
FNO	1.6%

A more realistic problem: NEMO data set

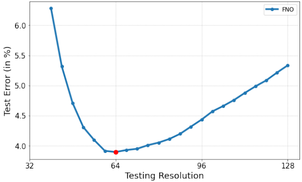


- Large errors with FNO.
- What is going on ?

Another Red Flag



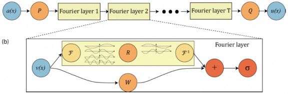
- FNO Results:



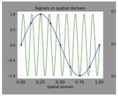
- Recall Desiderata for Operator Learning:
  - Input + Output are functions.
  - Some notion of Resolution Invariance

A Possible Culprit: Aliasing Errors

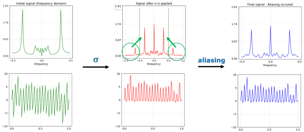
- We evaluate FNO inputs on a grid:



- Can lead to Aliasing:



- Particularly due to the Activation:



- Rigorous analysis in Bartolucci et al, 2023.